User Manual for OpenHarmonics

Prepared by

Prof. Peter W. Lehn, Ph.D., P.Eng. and Philippe A. Gray, B.A.Sc.

April 4, 2020

Contents

Chapter 1

Overview

1.1 OpenHarmonics

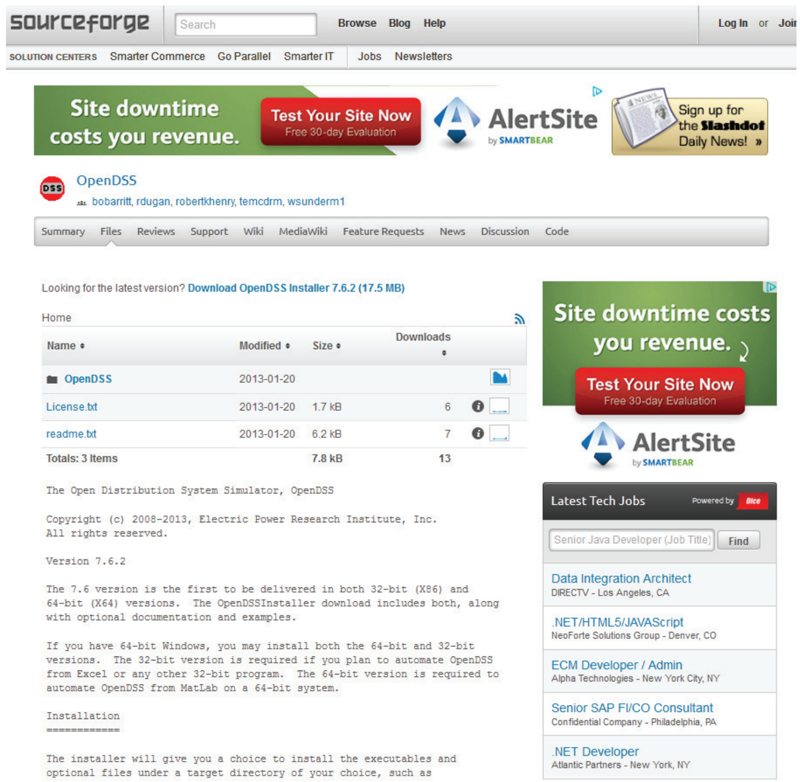
OpenHarmonics is a harmonic analysis software tool designed to perform high-accuracy harmonic analysis studies of distribution systems. OpenHarmonics has the ability to analyze the voltage and current harmonic spectra throughout an arbitrarily large system for an arbitrary number of harmonics. OpenHarmonics is built on-top of the OpenDSS open source distribution system simulator with the primary purpose of extending the harmonic analysis functionality of OpenDSS through the use of exact-accuracy power electronic converter models. As such, the advantages of OpenHarmonics are realized when the distribution system to be analyzed has loads/generators interfaced to the grid through power electronic converters. The derived converter models are somewhat more computationally intensive than the converter models currently being used in the commercial harmonic analysis software programs. However, the derived converter models are orders of magnitude more accurate–offering accurate prediction of non-characteristic harmonics (e.g. positive sequence 3rd harmonic) that are entirely neglected in commercial harmonic analysis software. OpenHarmonics was developed in MATLAB and the user interacts with OpenHarmonics through the MATLAB workspace.

1.2 Installation

In terms of software programs, both MATLAB (only the base software is needed) and OpenHarmonics are required. Note at this time, OpenDSS is only available for the Windows OS. The following two sections outline step-by-step the installation process for OpenDSS and the configuration process for OpenHarmonics.

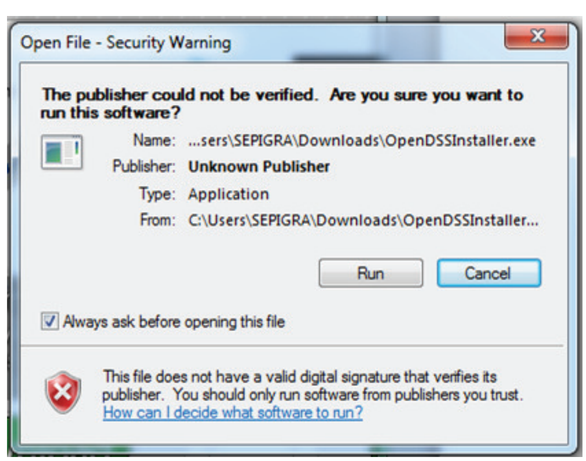
1.2.1 Installing OpenDSS

1. Go to: `http://sourceforge.net/projects/electricdss/files/`

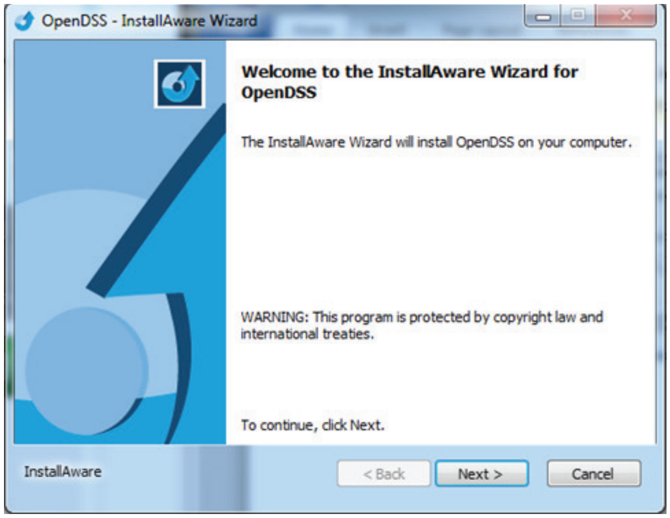


Click on the "Download OpenDSSInstaller 7.6.1 (16.9 MB)" link at the top of the page or download the latest version of OpenDSS available. Your download should start in a few seconds. Save the files on your local disk. Once the download has been completed, run the downloaded file: "OpenDSSInstaller.exe".

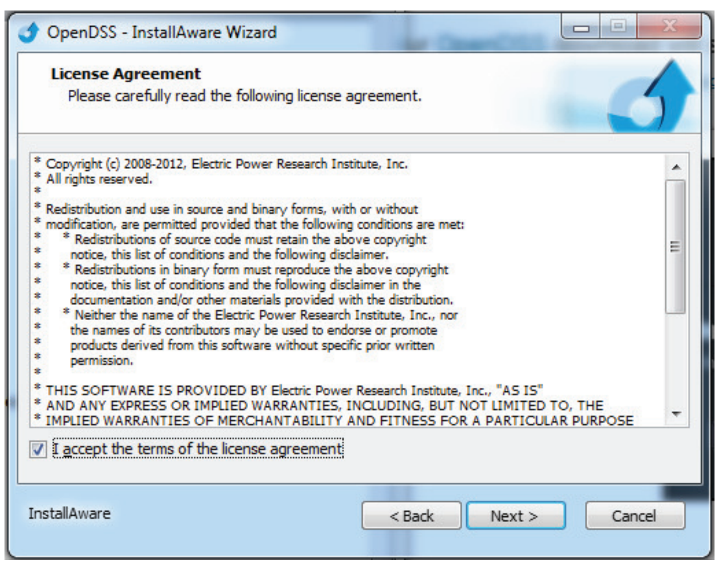
2. Click "Run"



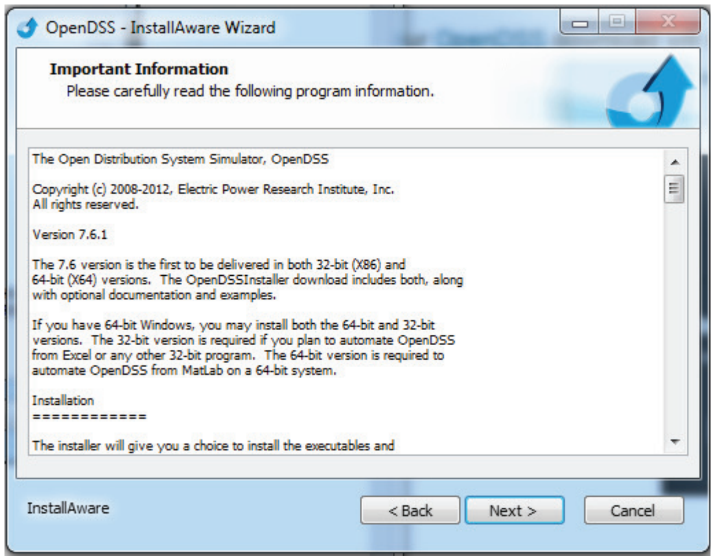
3. Click “Next”



4. Check the “I accept the terms of the license agreement” box and then click “Next”.



5. Click "Next"



6. In the "Custom Setup" window, change the program features to be identical to Fig. 1.1 or Fig. 1.2, for a 64–bit or 32–bit Windows OS, respectively. You can make changes by clicking on the icons next to the texts and selecting appropriate options. When finished this step, click "Next".

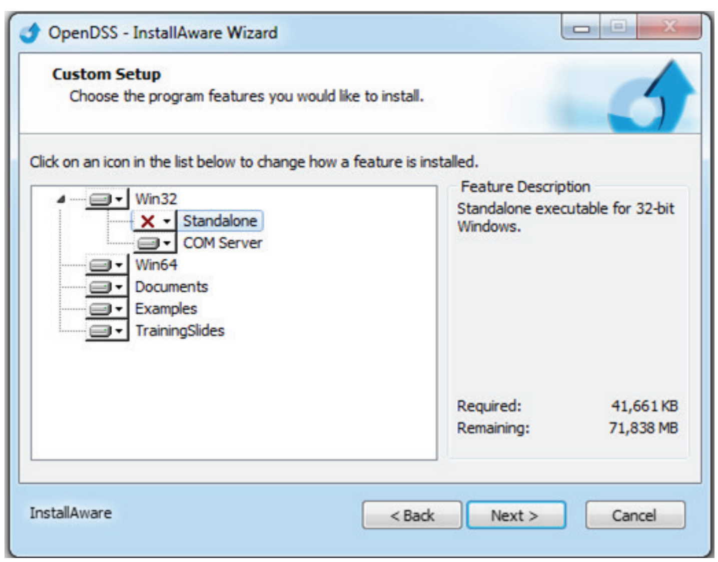


Figure 1.1: Configuration for 64-bit Windows OS

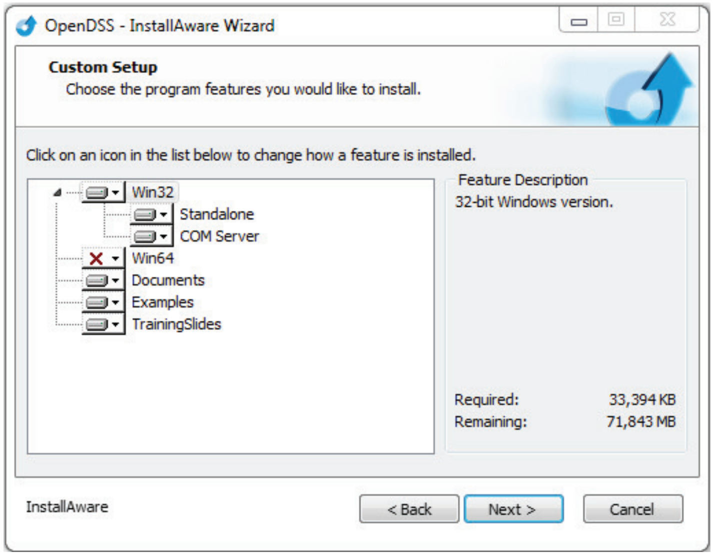
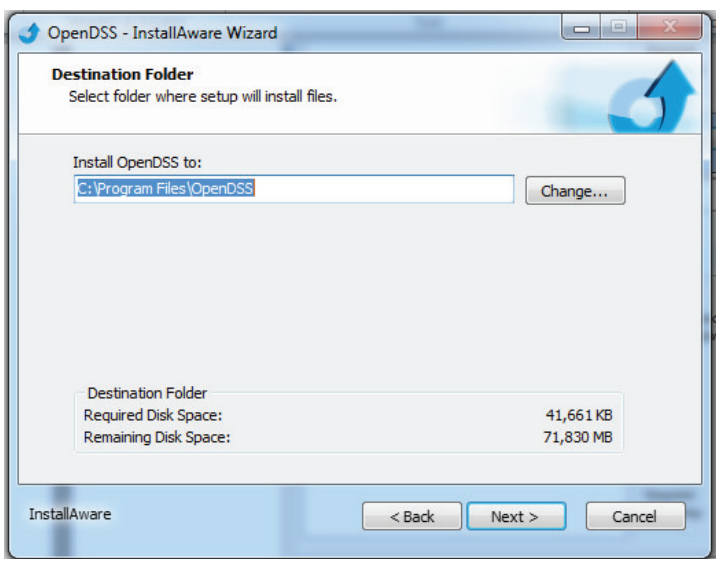
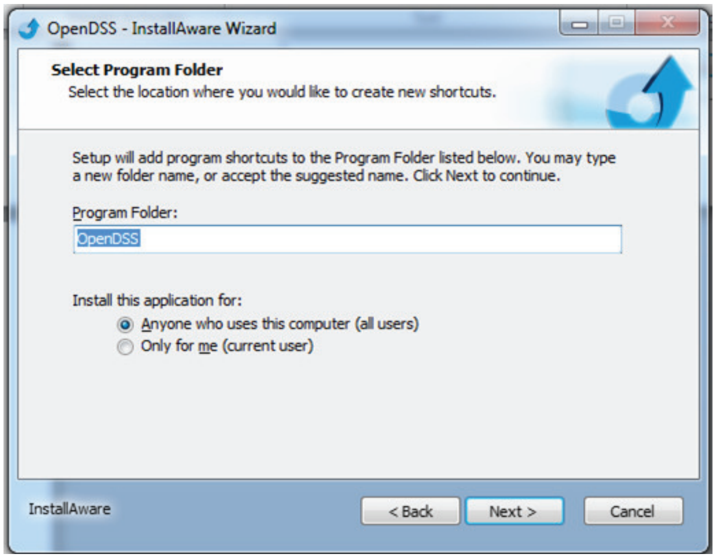


Figure 1.2: Configuration for 32-bit Windows OS

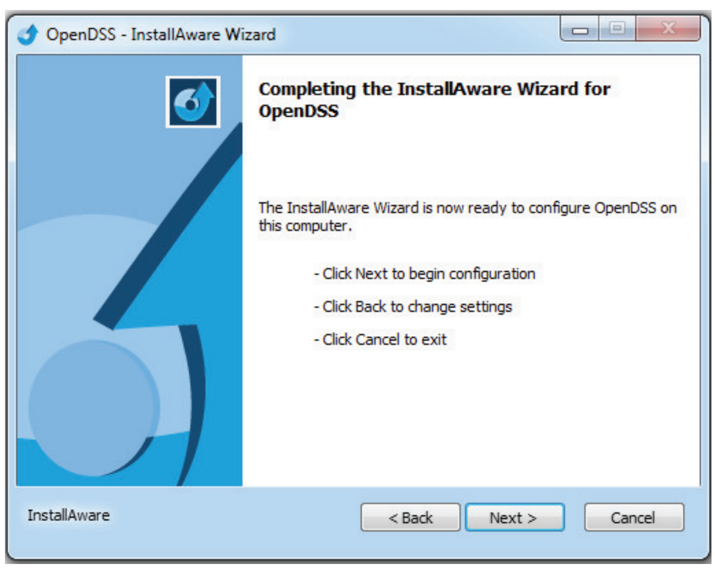
7. Install to "\Program Files\OpenDSS". Click "Next".



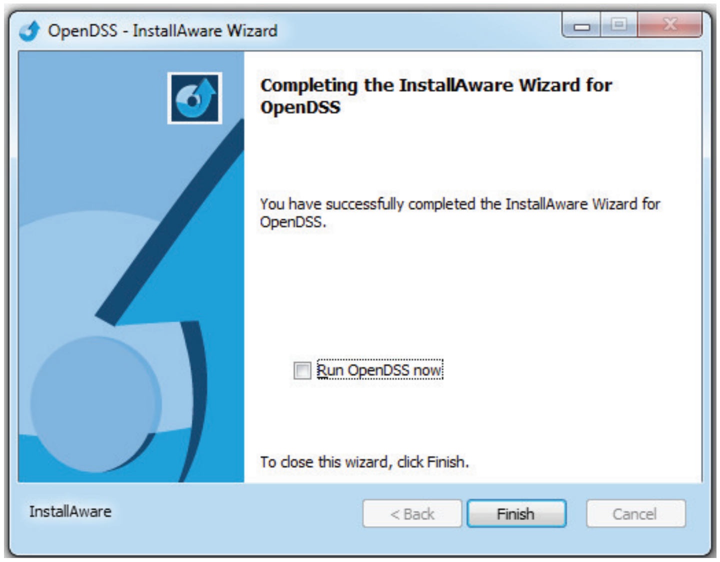
8. Select "Program Folder" and define who can access the program at the bottom of the window. Click "Next".



9. Click "Next".



10. Click “Finish”



OpenDSS is now fully installed on your computer.

1.2.2 Configuring OpenHarmonics

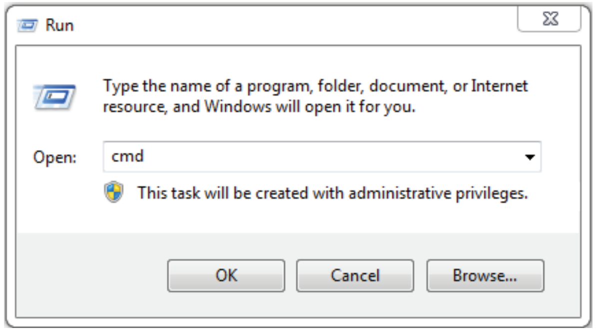
There are three steps to configure OpenHarmonics on the local computer: 1) extracting the contents of OpenHarmonics.zip to an appropriate directory, 2) registering the COM server, and 3) modifying the MATLAB paths.

Extracting Contents of OpenHarmonics.zip

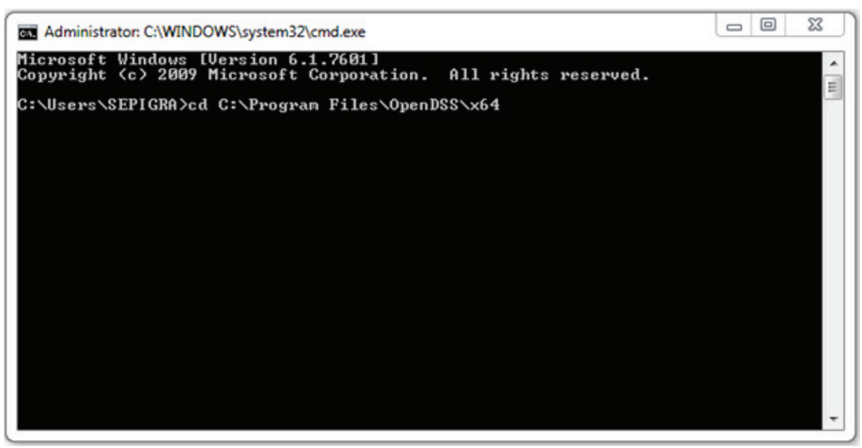
Assuming that you have the file “OpenHarmonics.zip”, extract the contents of “OpenHarmonics.zip” into the folder where you will perform the harmonic analysis studies. For example, I have the OpenHarmonics folder located on my desktop: “\Desktop\OpenHarmonics”.

Registering the COM server

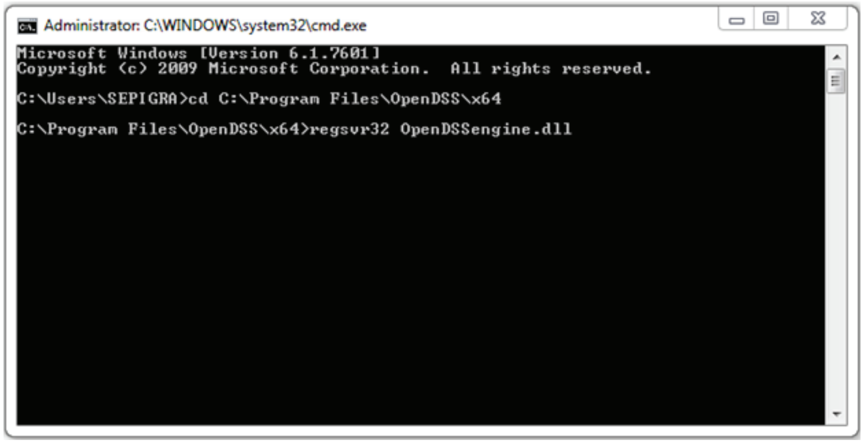
1. Open the Start menu and go to Start → Accessories → Run. Type in “cmd” in the “Open:” text field. Click “OK”.



2. The command prompt window should now be displayed. In this step, the working directory of the command prompt is changed to the location of the "OpenDSSengine.dll" file. "OpenDSSengine.dll" is located in the installation directory. For example, if OpenDSS is installed in "C:\Program Files", "OpenDSSengine.dll" would be located in "C:\Program Files\OpenDSS\x64". Therefore for this case, the command prompt line should be modified to: "cd C:\Program Files\OpenDSS\x64", as is shown in the figure below. Once this is done, click Enter on the keyboard.



3. Type in: "regsvr32 OpenHarmonics.dll". Click Enter on the keyboard.

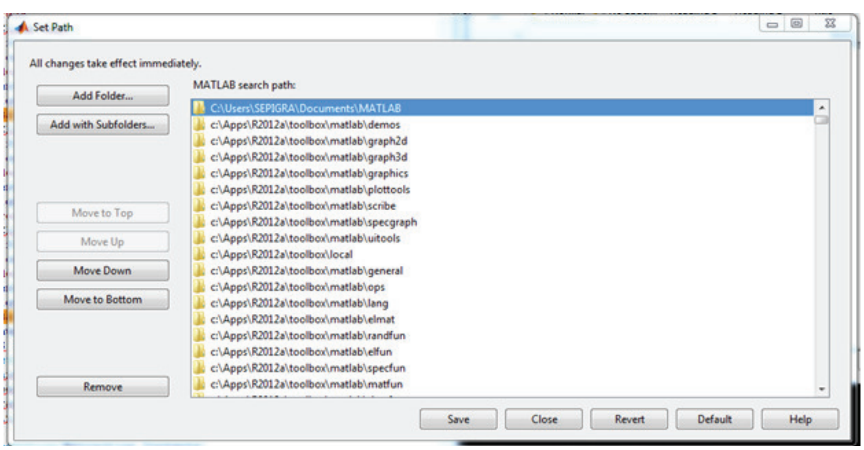


MATLAB should now able to communicate with OpenDSS. Note, if an error message is encountered at this stage, open a new terminal with administrative privileges and try this step again.

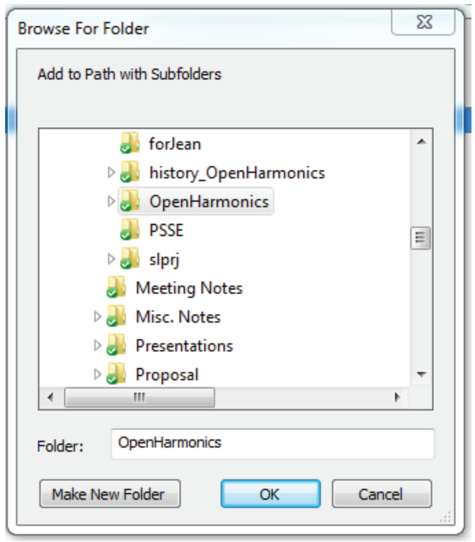
Modifying Path in MATLAB

To ensure that the OpenHarmonics subfiles can be located, the OpenHarmonics subfolders must be included in the MATLAB workspace path.

1. Run MATLAB.exe
2. Inside of the MATLAB workspace go to: File → Set Path...
3. Click on "Add with Subfolders..."



4. Locate and select the OpenHarmonics folder. Then click "OK".



5. Click "Save"

6. Click "Close"

OpenHarmonics will now be able to locate all of the subfiles required to run the software tool.

At this point, both OpenDSS and OpenHarmonics should be fully installed and configured.

1.3 Features

The key feature in OpenHarmonics is the ability to perform high-accuracy harmonic analysis studies of distribution systems when converter interfaced loads/generators are present in the system being analysed. To achieve this, 1) OpenDSS is leveraged to perform the harmonic analysis and 2) innovative converter modeling research, conducted at the University of Toronto, is used to implement the converter models.

1.3.1 OpenDSS

OpenDSS is a steady-state frequency-domain harmonic analysis software used to perform harmonic analysis studies of distribution systems. Systems are built in OpenDSS using a text-based interface—a GUI does not exist at this time. Some of the key features of OpenDSS—that are applicable to OpenHarmonics—are:

- Solves balanced and unbalanced systems
- Arbitrary number of harmonics

- Applicable for single as well as three phase systems
- Solves harmonics using the current injection approach
- Solves fundamental frequency power flow through use of Newton Raphson technique

OpenDSS features a fairly exhaustive library of customizable models for common circuit elements that can readily be used. These modeled circuit elements include: capacitors, transformers, lines, reactors, transformers, generators, loads, voltage sources, and current sources. For more information on OpenDSS refer to the OpenDSS manual (located in the local directory where OpenDSS was installed) and/or the following wiki webpage: `http://electricdss.wiki.sourceforge.net/`.

1.3.2 Converter Modelling Research

OpenHarmonics includes 5 converter models that are high-accuracy, have intuitive inputs, and are up to 10x faster than the closest models in literature. This high-accuracy is achieved because very few assumptions are made in the derivation of the models (the assumptions that were made are discussed in the Technical Description section of this manual). Each of the 5 converter models feature the following properties:

- Includes the interaction of all harmonic sources on the network
- Does not suffer from harmonic truncation
- Has a high degree of customizability
- Maintains accuracy in unbalanced systems

The 5 converter models that are modelled are:

- 6-pulse voltage source converter
- 6-pulse diode rectifier
- 6-pulse thyristor rectifier
- 12-pulse diode rectifier
- 12-pulse thyristor rectifier

A significant percentage of generators/loads interfaced to the distribution network through power electronic converters are done through one of these converter types. Three-phase VSC converters are typically used to interface solar PV, wind, and FACTS devices to the network. And the 6 and 12 pulse converters are typically used in drive applications for both utilities and industry.

Chapter 2

Technical Description

To understand how OpenHarmonics works, it is first necessary to understand how OpenDSS works. As such, the relevant features of OpenDSS are first discussed. Then, OpenHarmonics will be discussed; in particular how OpenHarmonics is designed to extend the functionality of OpenDSS through use of the new converter models.

2.1 OpenDSS

There are a three main steps in the OpenDSS harmonic analysis procedure: 1) defining harmonic spectra of circuit elements, 2) computing fundamental power flow, and 3) sequential computation of harmonics.

Defining the Harmonic Spectrum of a Circuit Element

The harmonic spectra of relevant circuit elements are defined. The elements that have modifiable harmonic spectra are: current sources, voltage sources, loads, and generators (Note: impedances are not included here because OpenDSS automatically updates the impedance for each harmonic frequency). The magnitude and phase for each harmonic of interest must be specified prior to the simulation. The particular format of defining the harmonics is with respect to the fundamental frequency component of the particular circuit element. For example, a 3rd harmonic of a particular load could be defined as having a magnitude equal to 15% of the magnitude of the load's fundamental frequency current and a phase angle that leads the fundamental frequency current by 25°.

Fundamental Power Flow

Once the harmonic spectrum for each circuit element in the system has been defined, the system is solved at 60 Hz (the fundamental frequency can be changed to 50 Hz but for sake of discussion we assume 60 Hz here) using a Newton-Raphson based power flow technique. Once the convergence criteria has been met, the results are used to generate the actual values of the harmonic spectra for each circuit element in the system. For example, using the previous example, if the fundamental frequency solution returned a fundamental component of the load equal to $10\angle 45^\circ$ A, the 3rd harmonic would be assigned $1.5\angle 70^\circ$ A.

Sequential Solving of Harmonics

Now that the harmonic spectrum for each load, generator, current source, and voltage source have been defined in terms of actual values, the harmonic currents and voltages throughout the system can be solved. The harmonics are solved by sequentially solving the system at harmonic multiples 120Hz, 180Hz, ..., $n * 60$ Hz; where n is the number of harmonics defined by the user. Note, that the harmonic spectra is defined before the simulation begins and does not change during the simulation.

2.2 OpenHarmonics

In order to account for the interaction between harmonic sources on a system, the harmonic spectra of non-linear generators/loads must change during run time. As such, an iterative technique is required to update the harmonic spectra of each converter object, each time OpenDSS completes a simulation run. To use the 5 converter models in OpenDSS—and to perform these iterations—an active communication link is established between OpenHarmonics and OpenDSS so that OpenHarmonics can have control over the OpenDSS analysis. Each simulation, using the OpenHarmonics software tool, can be seen as consisting of three steps: 1) definition of system, 2) iterative harmonic analysis, and 3) generation of results, as shown in Fig. 2.1.

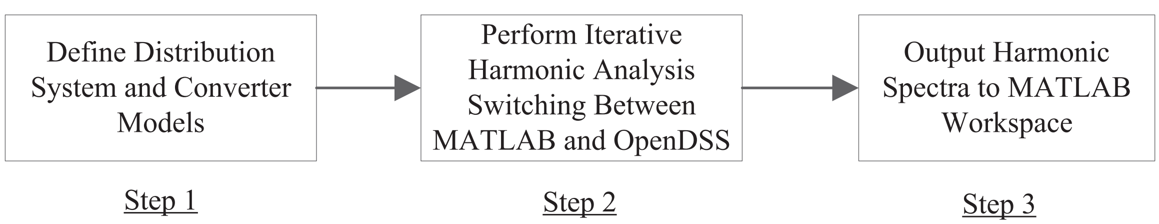


Figure 2.1: OpenHarmonics Functional Blocks

Step 1) Definition of System

The distribution system is fully defined, except for the converter models, in OpenDSS; the converter models are defined in MATLAB. Additionally, the number of harmonics to analyze are specified in MATLAB. To allow MATLAB to communicate with OpenDSS, the COM interface is utilized. The COM interface is a built-in feature common to both MATLAB and OpenDSS. It is initiated at start-up to allow two-way communication between OpenHarmonics and the OpenDSS circuit files which define the distribution system to be simulated. Using this COM interface, MATLAB has total control over all the features in OpenDSS.

Step 2) Iterative Harmonic Analysis

OpenHarmonics simulates a system by iteratively switching back and forth between MATLAB and OpenDSS until the change in PCC voltages, for each of the converters, does not appreciably differ between successive iterations. A high-level flow chart, outlining how OpenHarmonics switches between MATLAB and OpenDSS, to solve the converter harmonics, is shown in Fig. 2.2. The first step, in each iteration, is to solve for the system voltage harmonics with OpenDSS. To do this, three steps are performed:

1. The fundamental power flow solution of the system is first solved for.
2. Using this solution, the harmonic currents of each element in the network are calculated.
3. Using the calculated harmonic currents, OpenDSS sequentially solves the system at each of the harmonic frequencies of interest. For example, if the user requires 15 harmonics to be solved, the system would solve the system first for 120 Hz, than 180 Hz, ... 900 Hz.

Once the system has been solved at all harmonics of interest, the resulting voltage harmonic spectra are read into MATLAB where they are used as inputs to the converter models. The converters are then solved one-by-one for their respective line current harmonic spectra. These line current harmonic spectra are then fed back into OpenDSS to update the harmonic spectrum of each current source object representing a converter. This entire process constitutes one iteration. And this process repeats until the solved PCC voltage harmonics from one iteration to the next do not differ by more than 5%.

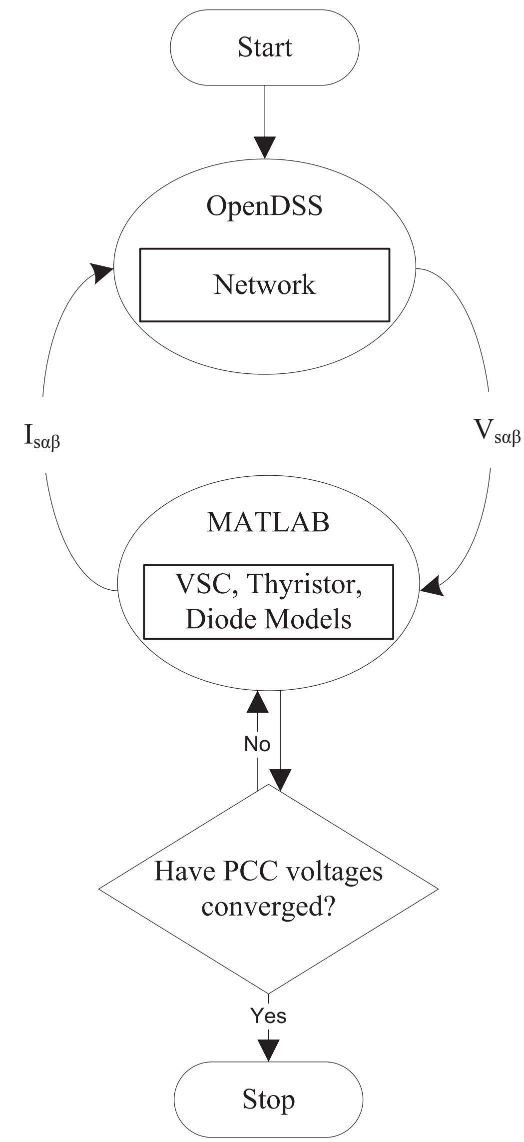


Figure 2.2: OpenHarmonics—Iterative Harmonic Analysis Functional Block

Step 3) Generation of Results

Once the system has been solved, the harmonic spectra of the bus voltages, of interest to the user, are saved in the MATLAB workspace.

2.3 Derived Converter Models

OpenHarmonics features five high-accuracy converter models, which are listed as follows:

- 6-pulse voltage source converter (VSC)
- 6-pulse diode rectifier
- 6-pulse thyristor rectifier
- 12-pulse diode rectifier
- 12-pulse thyristor rectifier

Each model is solved using the same general approach and therefore this general approach is first described. After this, the specifics of each of the five converter models are briefly outlined.

2.3.1 Derivation Approach

Each converter model is a frequency-domain model but solved using time-domain derived equations. This eliminates harmonic truncation errors normally associated with frequency-domain models, while retaining a speed more comparable to a model solved in the frequency-domain. The specific derivation technique used is called the frequency coupling matrix method (FCM), alternatively termed the harmonic admittance matrix method. The FCM is a matrix that couples the ac and dc side harmonics in a system. The FCM is illustrated mathematically in (2.1).

$$\begin{bmatrix} \underline{I_{ac}} \\ \underline{V_{dc}} \end{bmatrix} = \mathrm{FCM} \begin{bmatrix} \underline{V_{ac}} \\ \underline{I_{dc}} \end{bmatrix} \tag{2.1}$$

where, $\underline{I_{ac}}$ is a vector containing the harmonic components of the ac-side currents, $\underline{V_{dc}}$ is a vector containing the harmonic components of the dc-side voltage, $\underline{V_{ac}}$ is a vector containing the harmonic components of the ac-side voltages, $\underline{I_{dc}}$ is a vector containing the harmonic components of the dc-side current.

Once the FCM, corresponding to the input vector $\begin{bmatrix} \underline{V_{ac}} \\ \underline{I_{dc}} \end{bmatrix}$, has been calculated, the ac-current harmonics of the converter can be directly solved. Since, only the ac-current harmonics are needed when solving harmonic power flow of a system, the bottom half of the FCM is ignored in OpenHarmonics.

The output of each converter model is a vector containing the ac-side current harmonics specified in space-vector format or $\underline{I_{s\alpha\beta}}$.

2.3.2 Three-Phase VSC Model

The VSC topology assumed in the derivation of the model is shown in Fig. 2.3.

The required inputs to the VSC model are the following: V_{dc}, P, Q, $\underline{V_{s\alpha\beta}}$, f_{sw}, and L; V_{dc} is the average voltage on the dc-link of the converter; P is the average power absorbed by the dc-side current source; Q is the reactive power drawn by the

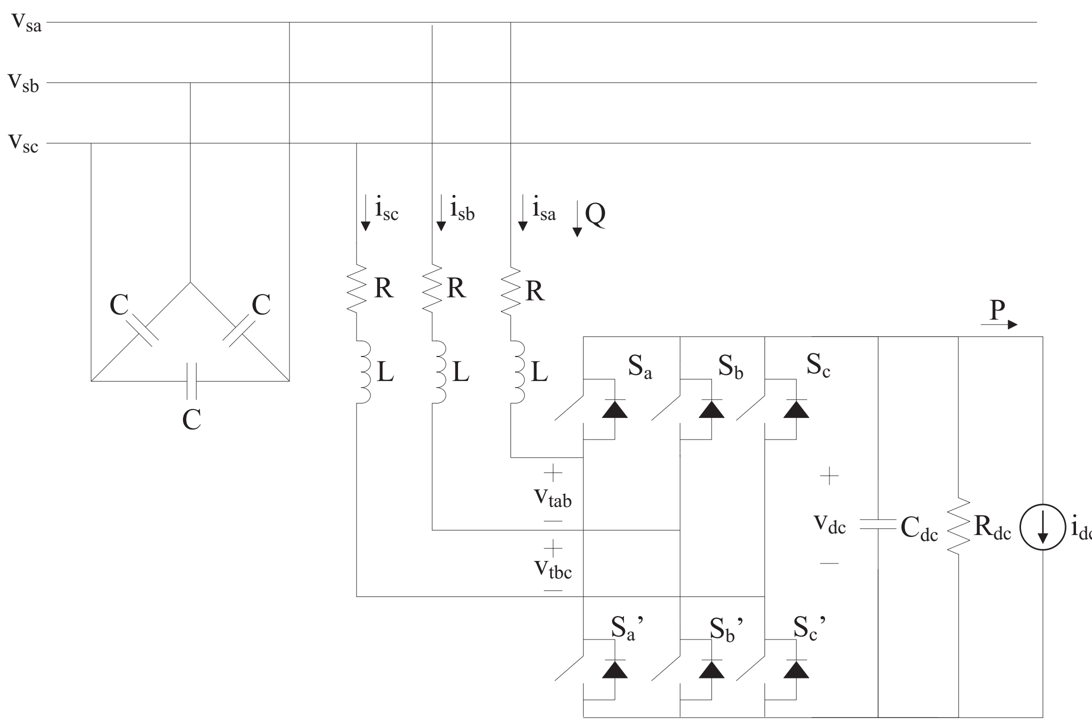


Figure 2.3: Assumed 3 Phase VSC Converter Topology

converter; $\underline{V_{s\alpha\beta}}$ is a vector containing the PCC voltage harmonic spectra; f_{sw} is the switching frequency in Hz; and L is the equivalent line inductance. C is an optional input, that specifies the capacitance of the input capacitor bank. If a value for C is not provided, the capacitor bank is not added to the model. $\underline{V_{s\alpha\beta}}$ is generated by OpenDSS once per iteration—after the call to the harmonic flow analysis procedure. Note, the impedance values, not mentioned as inputs, are assumed by OpenHarmonics. For more information on these assumed values refer to Section 3.2.4.

2.3.3 6-Pulse Thyristor Converter Model

The 6-pulse thyristor rectifier topology assumed in the derivation of the model is shown in Fig. 2.4. In the derivation, it is assumed that the converter operates i) in continuous operation mode and ii) with 2 and/or 3 valve conduction. Furthermore, at this time the model is restricted to load mode.

The required inputs to the 6-pulse thyristor model are the following: V_{dc}, P, $\underline{V_{s\alpha\beta}}$, and L; V_{dc} is the average emf voltage on the dc-side of the converter; P is the average power absorbed by the dc-side emf voltage; $\underline{V_{s\alpha\beta}}$ is a vector containing the PCC voltage harmonic spectra; and L is the equivalent line inductance. C is an optional input, that specifies the capacitance of the input capacitor bank. If a value for C is not provided, the capacitor bank is not added to the model. $\underline{V_{s\alpha\beta}}$ is generated by OpenDSS

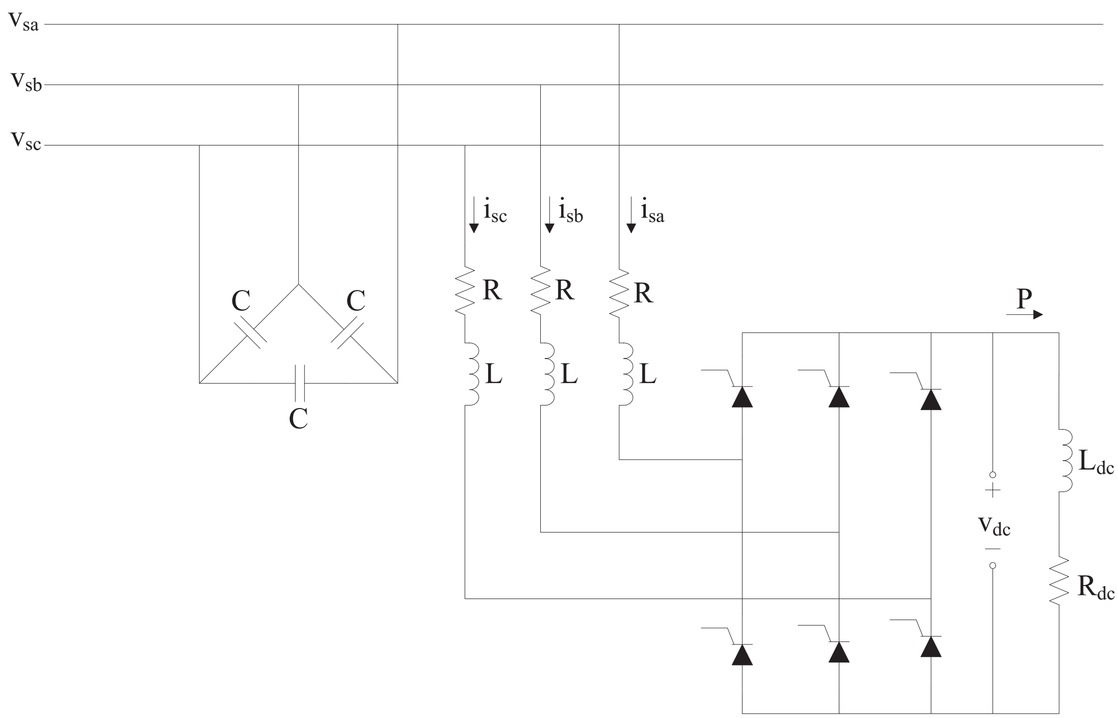


Figure 2.4: Assumed 6-Pulse Thyristor Converter Topology

once per iteration—after the call to the harmonic flow analysis procedure. Note, the impedance values, not mentioned as inputs, are assumed by OpenHarmonics. For more information on these assumed values refer to Section 3.2.4.

2.3.4 6-Pulse Diode Converter Model

The 6-pulse diode converter topology assumed in the derivation of the model is shown in Fig. 2.5. In the derivation, it is assumed that the converter operates i) in continuous mode and ii) with 2 and/or 3 valve conduction.

The required inputs to the 6-pulse diode model are the following: P, $\overline{V_{s\alpha\beta}}$ and L; P is the dc-power drawn by the load; $\overline{V_{s\alpha\beta}}$ is a vector containing the PCC voltage harmonic spectra; and L is the equivalent line impedance. C is an optional input, that specifies the capacitance of the input capacitor bank. If a value for C is not provided, the capacitor bank is not added to the model. $\overline{V_{s\alpha\beta}}$ is generated by OpenDSS once per iteration—after the call to the harmonic flow analysis procedure. Note, the impedance values, not mentioned as inputs, are assumed by OpenHarmonics. For more information on these assumed values refer to Section 3.2.4. Also, the R_{dc} variable is iteratively solved for such that the calculated P of the converter matches the user specified P.

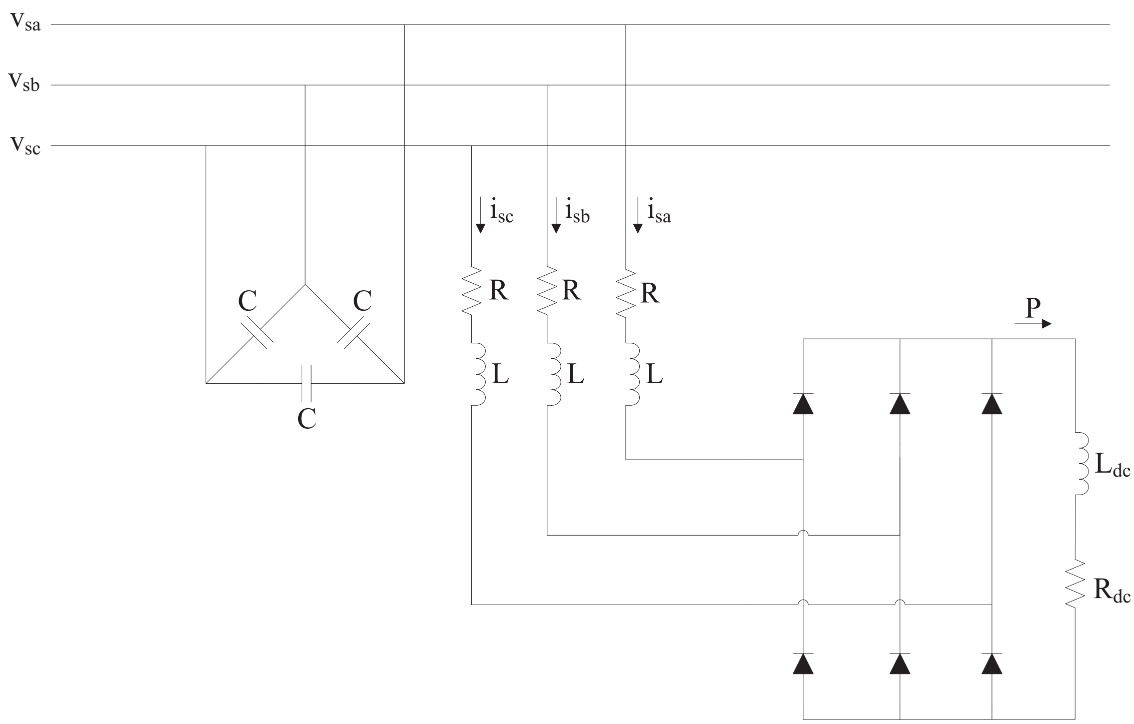


Figure 2.5: Assumed 6-pulse Diode Converter Topology

2.3.5 12-Pulse Thyristor Converter Model

The 12-pulse thyristor converter topology assumed in the derivation of the model is shown in Fig. 2.6. The inputs and outputs for this model are the same as the 6-pulse thyristor converter model. In the derivation, it is assumed that the converter operates i) in continuous operation mode and ii) with 4 and/or 5 valve conduction. Furthermore, at this time the model is restricted to load mode.

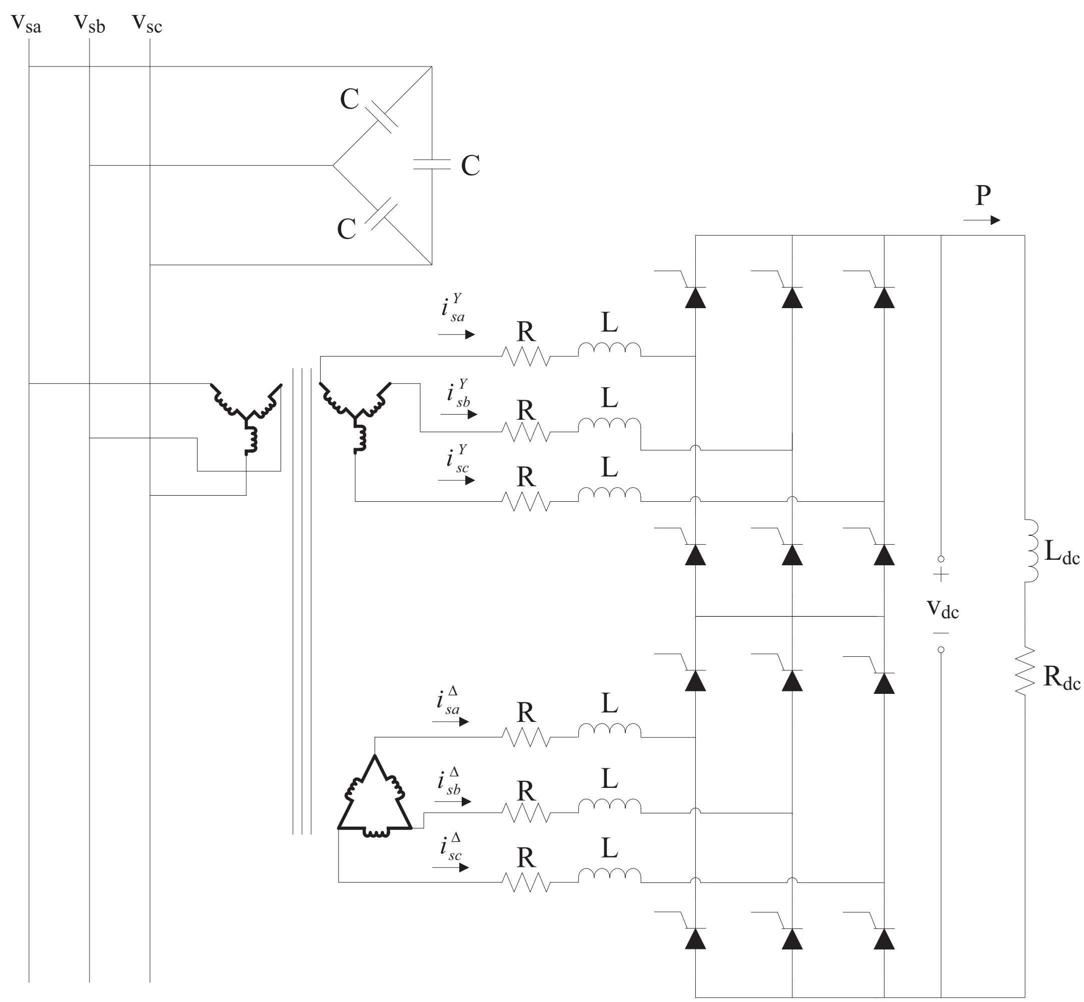


Figure 2.6: Assumed 12-pulse Thyristor Converter Topology

2.3.6 12-Pulse Diode Converter Model

The 12-pulse diode converter topology assumed in the derivation of the model is shown in Fig. 2.7. The inputs and outputs for this model are the same as the 6-pulse diode converter model. In the derivation, it is assumed that the converter operates with 4 and/or 5 valve conduction.

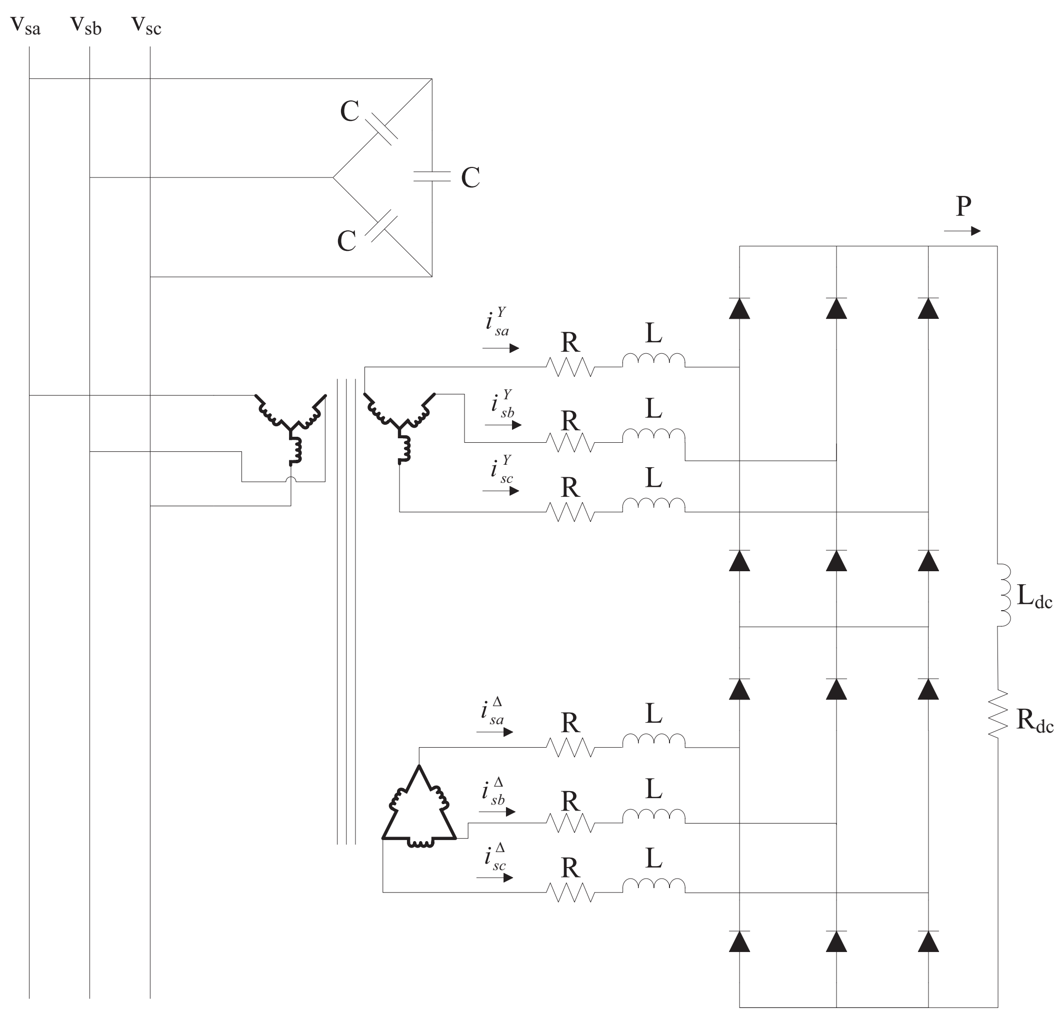


Figure 2.7: Assumed 12-Pulse Diode Converter Topology

Chapter 3

Harmonic Analysis with OpenHarmonics

To conduct harmonic analysis studies with OpenHarmonics, the user must first fully define the distribution system and simulation parameters. The distribution system—except for the converter interfaced generators/loads—is defined in OpenDSS. The simulation parameters and converter models are defined in the MATLAB OpenHarmonics.m file. The user interacts with OpenHarmonics through the MATLAB workspace.

3.1 Defining the Distribution System in OpenDSS

The distribution system is defined in the OpenDSS environment. The process of defining a system in OpenDSS is outside the scope of this user manual but as a reference some IEEE test systems have been written and tested by OpenDSS and can be found in the OpenHarmonics root folder at: \OpenHarmonics\IEEETestCases. Additionally, refer to the OpenDSS literature found in \Program Files\OpenDSS\Doc, as well as the following websites:

- `http://sourceforge.net/projects/electricdss/`
- `http://www.rogerdugan.com/OpenDSS/`
- `http://sourceforge.net/apps/mediawiki/electricdss/`.

3.2 Defining Simulation Parameters in OpenHarmonics.m

The user interacts with OpenHarmonics entirely through OpenHarmonics.m; the OpenHarmonics.m code is shown in Fig. 3.1. It is through OpenHarmonics.m that 1) simulation parameters are set; 2) converters are defined and added to the system; and 3) simulations are run.

The user makes these modifications through modifying certain blocks of code in OpenHarmonics.m. These blocks of code modify the following simulation parameters:

1. Maximum number of harmonics to analyze.
2. Directory path to the OpenDSS distribution system file
3. Converter model definition
4. The buses to analyze

The extent to which the user can modify these simulation parameters is the focus of the following sections.

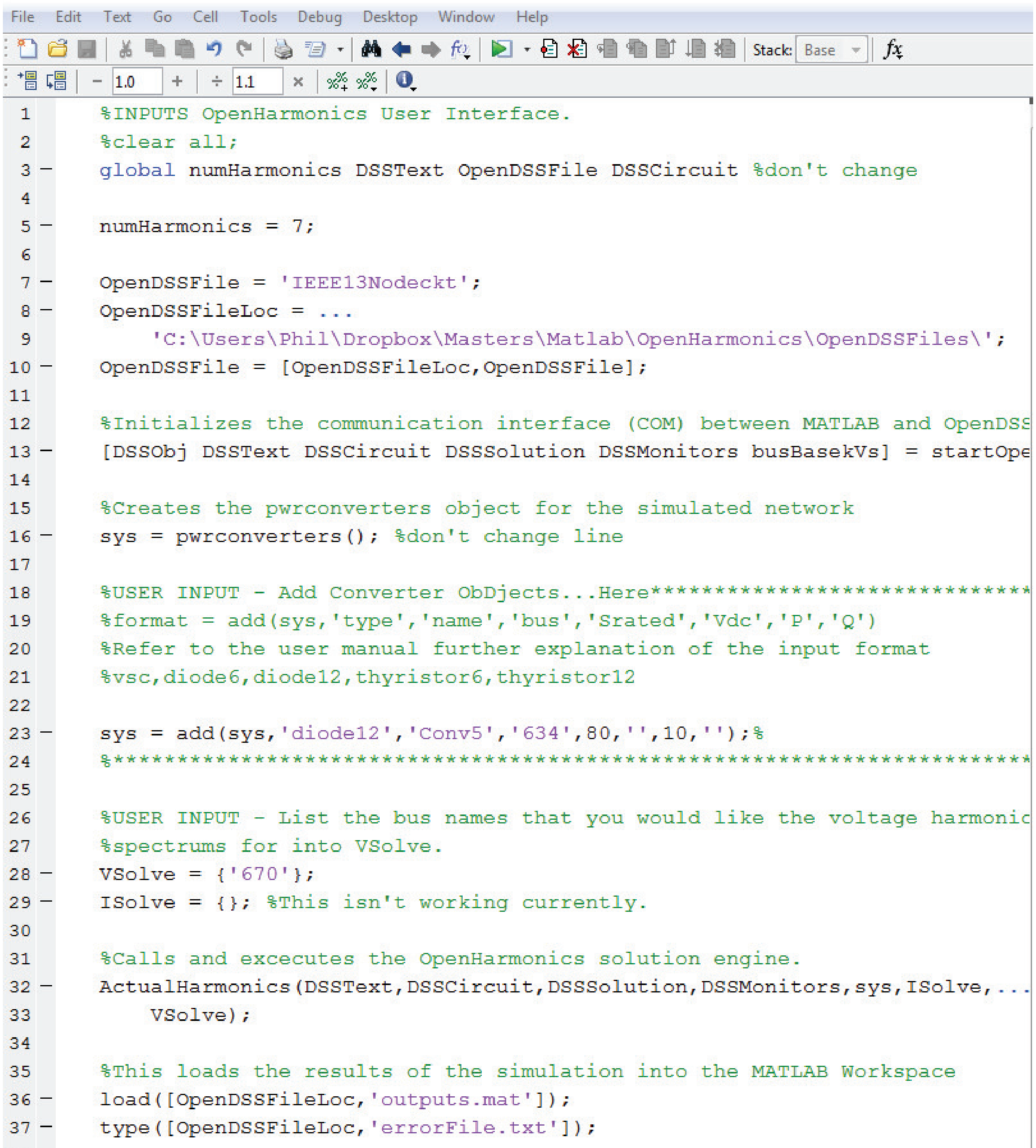


Figure 3.1: OpenHarmonics.m File

3.2.1 Setting the Number of Harmonics to Analyze

To change the maximum number of analyzed harmonics, modify the "numHarmonics = 7" line in OpenHarmonics.m which corresponds to line 5 of Fig. 3.1. The format of this line is the following:

$$\text{numHarmonics} = [\text{positive\_integer\_variable}]; \tag{3.1}$$

where, numHarmonics can be any positive interger value greater then and including 0.

As an example, by changing the line to "numHarmonics = 15", the system will be solved up to the 15th harmonic, which implies that OpenDSS will solve up to and including the 15th harmonic and the converter models will be derived and solved for up to the 15th harmonic. Note, the converter models do not make any harmonic truncation assumptions and therefore the results of the harmonic analysis are in most cases not affected by the value of numHarmonics. Also, note that the simulation time increases as the value of numHarmonics increases.

3.2.2 Specifying the OpenDSS File Path

The path to the OpenDSS file that defines the distribution system to be simulated must be referenced in OpenHarmonics. It is advised that this file is contained in the OpenHarmonics\OpenDSSFiles subfolder. Two lines must be modified in OpenHarmonics.m to reference the OpenDSS file:

The first line:

$$\text{OpenDSSFile} = \text{`[Insert\_OpenDSS\_File\_Name\_Here]'}; \tag{3.2}$$

which corresponds to line 10 of Fig. 3.1, references the name of the OpenDSS file. For example, if the OpenDSS file is called 'test.dss', the line would be changed to: "OpenDSSFile = 'test';".

The second line:

$$\text{OpenDSSFileLoc} = \text{`[Insert\_the\_Path\_to\_the\_OpenDSS\_File\_ Here]'}; \tag{3.3}$$

which corresponds to lines 8–9 in Fig. 3.1, references the path to the OpenDSS file, i.e. using the previous example, this would be the path to 'test.dss'. As an example, if this line was set to the path of 'test.dss' on the author's computer, it would be changed to: "OpenDSSFileLoc = 'C:\Users\SEPIGRA\Desktop\OpenHarmonics\OpenDSSFiles\';". Note, it is important to note that the name of the OpenDSS file must be the same as the name of the circuit referred to in that OpenDSS file. For example, if the OpenDSS file is called 'test' than in OpenDSS the circuit would be initialized as follows:

new circuit.test

3.2.3 Defining Converter Interfaced Loads/Generators

There is no hard-limit on the number of converters that can be added to the system. To add a converter to the system, the following line must be added to OpenHarmonics.m within the block of code contained between the lines 18–24 of Fig. 3.1:

$$\text{sys} = \text{add(sys,`[type]',`[name]',`[bus]'},S_{rated},V_{dc},P,Q,L,f_{sw},C) \tag{3.4}$$

This line is only valid for adding one converter to the system. To add multiple converters, each converter requires its own line. The following sections describe each of the parameters in (3.4).

[**type**]

This value must be set to: ‘vsc’, ‘thyristor6’, ‘thyristor12’, ‘diode6’, or ‘diode12’. This parameter refers to the type of the converter object. These types reference the vsc, thyristor and diode derived converter models. Note, the values are case-sensitive.

[**name**]

This value can be set to any string value. The parameter specifies a name identifier attached to the converter object. Each converter in the system must have a unique name.

[**bus**]

This value must reference a valid bus that exists in the system. This parameter specifies the bus in the system that the converter interfaces to. This value is case-sensitive.

S_{rated}

This value equals the rated power of the converter in kVA. This value can be approximated as it is only used to estimate the impedance values of the converter.

V_{dc}

For a converter of VSC type, this parameter specifies the average voltage across the dc-link capacitor in units of V. For a converter of thyristor type, this parameter refers to the average value on the dc-side of the converter in units of V. This parameter is not specified for the diode converters.

For the 6-pulse thyristor model, if the specified $V_{dc} > |V_s^{+1}|_{LL}^{pk}$, a zero-vector will be returned; where, $|V_s^{+1}|_{LL}^{pk}$ is the amplitude of the positive-sequence line frequency line-to-line PCC interconnect voltage for the particular converter. A zero-vector is returned because any average value about $|V_s^{+1}|_{LL}^{pk}$ is not possible. For the 12-pulse thyristor model, if the specified $V_{dc} > 1.928|V_s^{+1}|_{LL}^{pk}$ a zero-vector will be returned for similar reasons.

Table 3.1: Recommended Per-Phase Line-Inductance L for VSC as a Function of Converter Switching Frequency

f_{sw} (kHz)	L (pu)
1.5	0.2
3	0.12
8	0.06

P

For a converter of VSC type, this parameter refers to the average power, in kW, absorbed by the dc-side current source. For a converter of thyristor type, this parameter refers to the average power, in kW, absorbed by the dc-side emf source. For a converter of diode or thyristor type, this parameter refers to the average power, in kW, absorbed by the dc-side equivalent load resistance. For the VSC model P is restricted to the range -3 pu to +3 pu. For the thyristor and diode converter models P is restricted to the range 0 to +3 pu.

Q

This parameter is only used for converters of VSC type and refers to the average reactive power, in kvar, absorbed by VSC converter.

L

This parameter specifies the per-phase line inductance, in pu. This parameter is used for all 5 converter types. It is recommended, if an actual value is not known, to use the value of 0.2 for the diode and thyristor rectifier models. For the VSC model, the recommend values are shown in Table 3.1. Refer to Section 3.2.4 for how the per-unit value of L is converted internally to the actual value.

f_{sw}

This parameter is only used for converters of VSC type and refers to the switching frequency of the converter, in Hz.

C

This parameter specifies the capacitance, in pu, of the capacitor bank for the particular converter model. This is an optional input parameter for all 5 converter types and therefore if no value for C is specified, i.e. ‘’, than the capacitor bank is removed from that model. Refer to Section 3.2.4 for how the per-unit value of C is converted internally to the actual value.

Note: that not all of these parameters are used for each of the 5 converter types. In the cases where a particular parameter is not needed, simply leave that entry empty.

Example: Adding a VSC to the IEEE-34 Test System

As an example, a VSC converter is added to the IEEE-34 Bus Test System by adding the following line to OpenHarmonics.m:

$$\text{sys} = \text{add(sys, 'vsc', 'conv\_1', '814', 10, 800, 8, 2,0.2,900,'')} \tag{3.5}$$

The updated distribution system is graphically represented in Fig. 3.2.

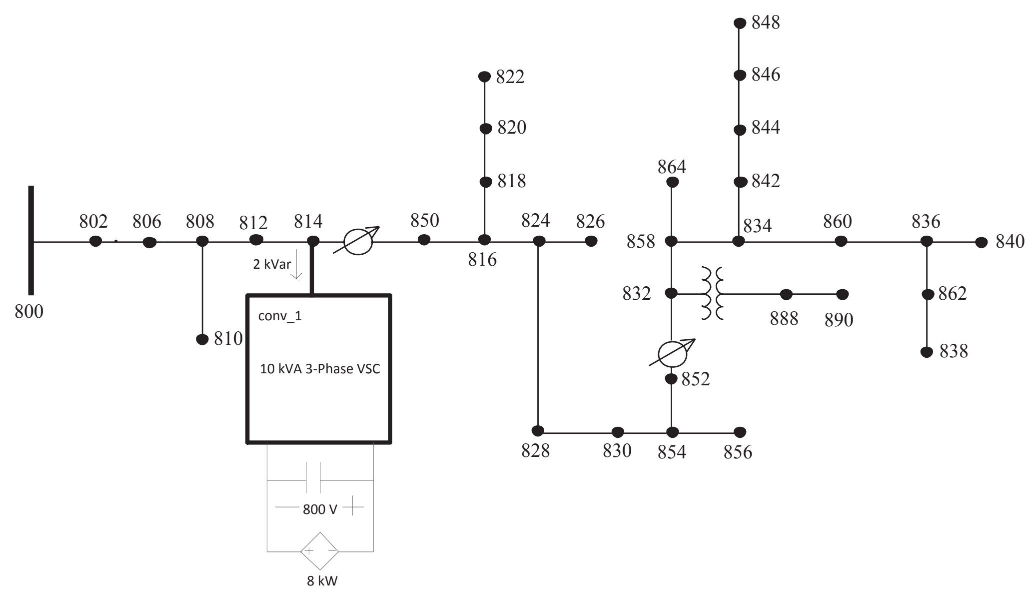


Figure 3.2: VSC Added to IEEE-34 Test System

3.2.4 Assumed Impedance Parameter Values

The converter models also have impedance parameters that need to be specified. For the VSCs these parameters are: C_{dc}, R_{dc}, and R. For the diode and thyristor based converters these parameters are: R, R_{dc}, L_{dc}. The impedance values are calculated and set by OpenHarmonics (Note, the converter properties are public properties and therefore these default values can be easily overwritten by the user). The assumed impedance values for the VSC, and the line-commutated converters—6-pulse and 12-pulse thyristor and diode rectifiers—are shown in Table 3.2 and 3.3 respectively.

Table 3.2: Assumed Per-Unit Impedances for VSC

Impedance Parameter	Value (pu)
R	0.02
R_{dc}	10^8

Table 3.3: Assumed Per-Unit Impedances for the Line-Commutated Converter Models

Impedance Parameter	Value (pu)
R	0.02
L_{dc}	0.5

The actual impedance values are calculated by combining the corresponding per-unit values with the base impedance Z_{base}. Z_{base} is calculated as follows

$$Z_{base} = \frac{3}{1000}\frac{V_{base}^2}{S}[\Omega] \tag{3.6}$$

where, V_{base} is defined by OpenDSS; and refers to the base-voltage at the PCC that the converter is interfaced to.

The process of using Z_{base} and the per-unit values to calculate the actual impedance values is illustrated for L, C and R as shown in (3.7), (3.7) and (3.9).

$$L = L_{pu}\frac{Z_{base}}{2\pi 60} \tag{3.7}$$

$$C = C_{pu}\frac{1}{2\pi 60 Z_{base}} \tag{3.8}$$

$$R = R_{pu}Z_{base} \tag{3.9}$$

Unlike the other impedance parameters, the actual dc-link capacitance C_{dc} for the VSC is directly calculated. There is a general rule of thumb for limiting the ripple on the dc-link:

$$\text{2-3 J per kVA(rated)} \tag{3.10}$$

This expression says that for every additional kVA of rated power of the converter the energy storage on the dc-cap should increase by 2-3 J. In OpenHarmonics, we assume a value of 4 J per kVA(rated). As such, the following equation results for energy E,

$$E = 4 * S \tag{3.11}$$

E for dc-capacitors is also equal to the following

$$E = \frac{1}{2} C_{dc} V_{dc}^2 \tag{3.12}$$

By combining (3.11) and (3.12), the following expression for C_{dc} is realized

$$C_{dc} = 8 \frac{S}{V_{dc}^2} \tag{3.13}$$

This expression is used to calculate the dc-link capacitance of the VSC.

It is important to note that it is possible for the user to override these assumed impedance values by utilizing the following function:

sys = changeValue(sys,'[name]','[parameter]','[value]')

where, [name] is the name of the model object; [parameter] is the name of the parameter for that model object that you would like to modify; and [value] is the new parameter value.

This function can be added to modify all model parameters and should be added in OpenHarmonics after the model has been added to the system, otherwise it will not work.

3.3 Example Case

In this example, a harmonic analysis study is conducted on the IEEE 13 bus test system for 21 harmonics with a 100 kVA VSC, and a 50 kVA 6-pulse diode converter interfaced load connected to bus 634. This system is shown in Fig. 3.3.

This particular IEEE 13-bus network is defined in a file named "IEEE13Nodeckt" and is located on my Dropbox folder at:

C:\Users\SEPIGRA\Dropbox\Masters\Matlab\OpenHarmonics\OpenDSSFiles.

To simulate this system, the following steps were taken:

1. Modify NumHarmonics. Change line to:

   NumHarmonics = 21;

2. Modify OpenDSSFile. Change line to:

   OpenDSSFile = 'IEEE13Nodeckt';

3. Modify OpenDSSFileLoc. Change line to:

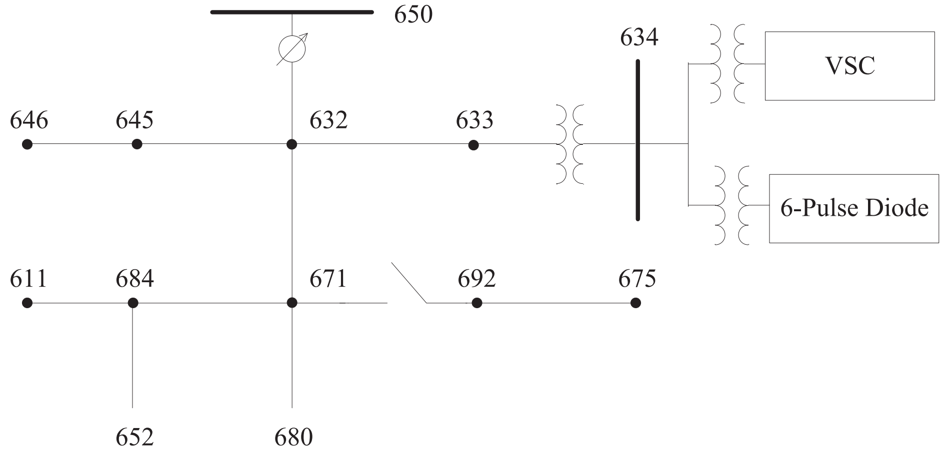


Figure 3.3: IEEE 13 Bus Test System

OpenDSSFileLoc = ‘C:\Users\SEPIGRA\Dropbox\Masters\Matlab\OpenHarmonics\OpenDSSFiles\’;

4. To add the VSC converter (with V_{dc} = 862, P = 88 kW, Q = -10 kvar) to the system, add the following line:

   sys = add(sys,‘vsc’,‘vsc1’,‘634’,100,862,88,-10,0.2,900,‘’)

5. To add the 6-pulse diode converter (with P = 30 kW) to the system, add the following line:

   sys = add(sys,‘diode6’,‘diode1’,‘634’,50,,30,‘’,0.2,‘’,‘’)

6. To generate the voltage harmonic spectra at bus 670, in additional to the voltage and current harmonic spectra at bus 634, modify VSolve as follows:

   VSolve = {’670’};

Once these modifications to OpenHarmonics.m have been made, click Run. After the simulation, to view the harmonic spectra outputs: click the “outputs” structure in the Workspace tab. “outputs” should contain 5 arrays: V_vsc1, I_vsc1, V_diode1, I_diode1, V_bus670.

V_vsc1 and V_diode1 are equivalent. They contain the positive and negative sequence voltage harmonics at bus 634. V_vsc1 is plotted in Fig. 3.4.

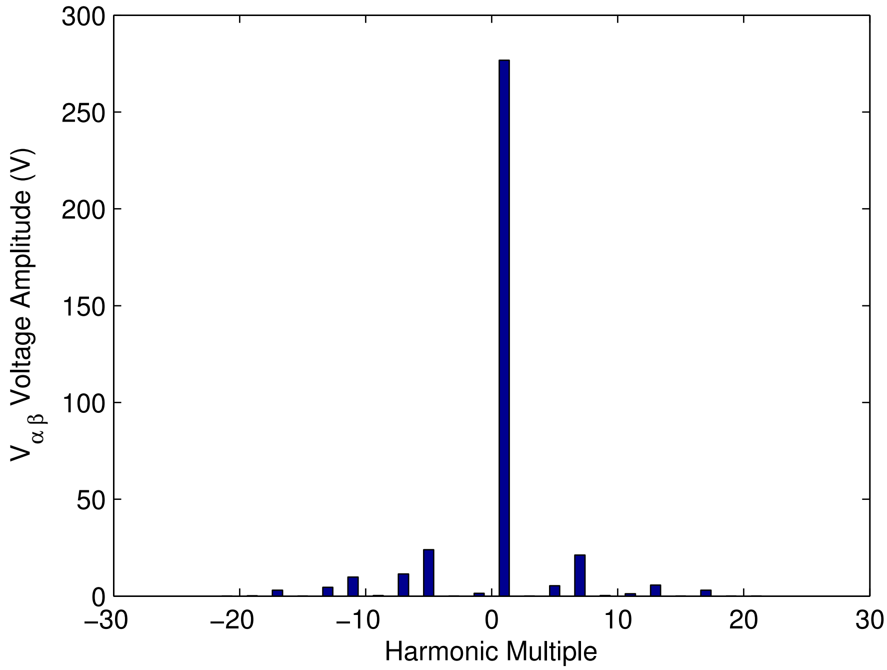


Figure 3.4: Bus 634 Voltage Harmonic Spectrum

I_vsc1 contains the positive and negative sequence current harmonics injected into bus 634 by the VSC. I_vsc1 is plotted in Fig. 3.5.

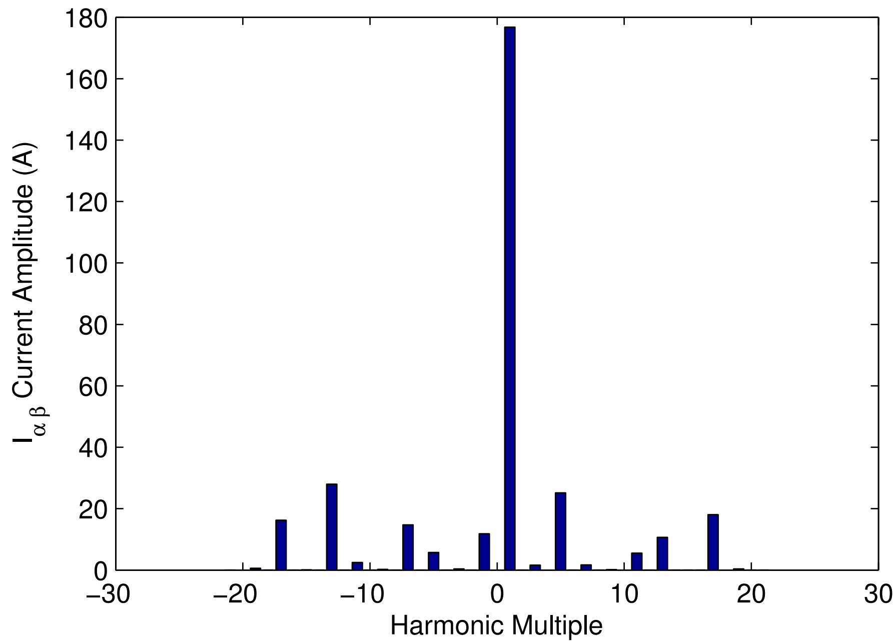


Figure 3.5: VSC Current Harmonic Spectrum

I_diode1 contains the positive and negative sequence current harmonics injected into bus 634 by the diode converter. I_diode1 is plotted in Fig. 3.6.

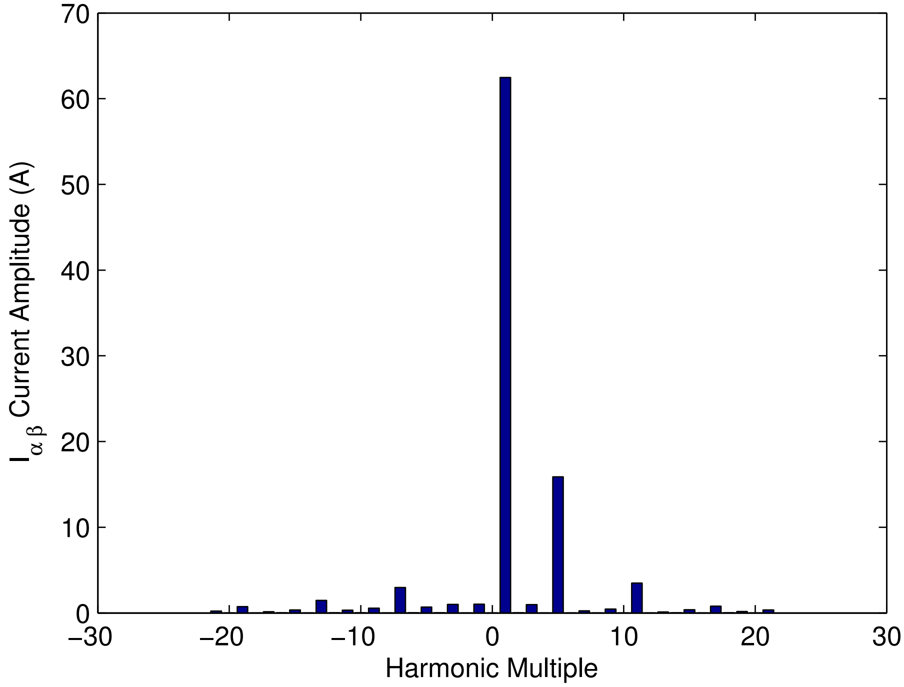


Figure 3.6: Diode Current Harmonic Spectrum

V_bus670 contains the positive and negative sequence voltage harmonics at bus 670. V_bus670 is plotted in Fig. 3.7.

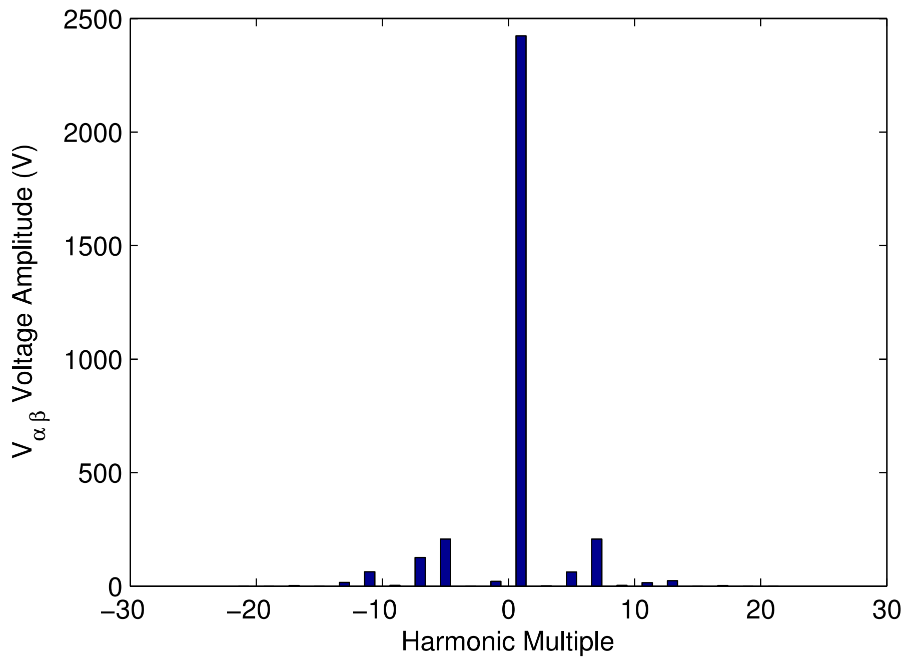


Figure 3.7: Bus 670 Voltage Harmonic Spectrum

Finally, the converter parameters used in the simulation are contained in the "sys" structure which is also found in the MATLAB workspace. To view the VSC converter parameters go to: sys → converterArr → (array element 1). For the diode converter parameters go to: sys → converterArr → (array element 2).