

Title: **Report for Deliverable 1.2 "Tool for Translating Network Data from ENWL to OpenDSS"**

Synopsis: This document describes the methodology used to process, correct and translate GIS-based data of ENWL low voltage distribution networks to be used by the distribution network simulation package OpenDSS.

Document ID: UoM-ENWL_LVNS_Deliverable1.2v01

Date: 19th June 2012

Prepared For: Dan Randles
Technology Development Manager
Electricity North West Limited, UK

Prepared By: Alejandro Navarro Espinoza
The University of Manchester
Sackville Street, Manchester M13 9PL, UK

Revised By: Dr Luis(Nando) Ochoa
The University of Manchester
Sackville Street, Manchester M13 9PL, UK

<i>Contacts:</i>	Dr Luis(Nando) Ochoa	Alejandro Navarro Espinoza
	+44 (0)161 306 4819	+44 (0)161 306 4807
	luis.ochoa@manchester.ac.uk	alejandro.navarroespinosa-2@postgrad.manchester.ac.uk

Executive Summary

This report corresponds to Deliverable 1.2 "Tool for Translating Network Data from ENWL to OpenDSS" part of the Low Carbon Network Fund Tier 1 project "LV Network Solutions" run by Electricity North West Limited (ENWL).

The aim of the LV Network Solutions project is to provide ENWL with greater understanding of the characteristics, behaviour, and future needs of their low voltage networks. This will be based on the analysis of data gathered by appropriate monitoring schemes to be deployed on hundreds of LV feeders and substations, and the assessment of the corresponding computer-based network models in current and future scenarios.

Given the volume of network data to be managed and the need of carrying out steady-state analysis (i.e., snapshot and time-series power flows), network models have to be created in a pseudo-automatic fashion. This report describes the methodology used to translate ENWL GIS-based network data into a format ready to be used by the distribution network simulation package OpenDSS (developed by the Electric Power Research Institute, USA).

GIS-based information of the Pavilion Way area received on February 2012 has been used to test the methodology. Although the data includes the topology and technical parameters of the corresponding LV distribution networks the main issue found was the connectivity of the network elements. For example, many line segments were not actually connected to the main LV feeder. Details about the connectivity issues and the corresponding automatic solution are presented in this report.

Once the connectivity aspect is solved, the database is translated to OpenDSS format also in an automatic way; this process is explained here as well.

To demonstrate the simulation capabilities of OpenDSS once the GIS files are adequately translated, some preliminary results from time-series simulations are presented in the last part of this report.

Contents

Contents	3
1 Introduction	4
1.1 Deliverable 1.2	4
1.2 General Description	4
2 Data Analysis	5
2.1 Pavilion Way Area.....	5
2.2 Polyline representation of the main feeder	6
2.3 Connectivity Issues	7
2.3.1 Breath First Search Implementation	8
2.3.2 Breath First Search Application	9
3 Reconnection Process	12
3.1 Problem Identification	12
3.1.1 Vertex to Vertex	12
3.1.2 Vertex to Line	12
3.2 Reconnection Methodology	12
3.2.1 Connection of Vertices.....	13
3.2.2 Determination of connected components	13
3.2.3 Reconnection of connected components	14
3.3 Results of the Reconnection Methodology	19
4 From the connectivity process to OpenDSS	21
4.1 OpenDSS Characteristics	21
4.2 OpenDSS Representation	22
4.3 OpenDSS Simulations	27
5 Conclusion	30

1 Introduction

As part of the transition towards a low carbon economy, Electricity North West Limited (ENWL), the Distribution Network Operator of the North West of England, is involved in different projects funded by the Low Carbon Network Fund. The University of Manchester is part of the Tier 1 project "LV Network Solutions".

The objective of this project is to provide ENWL with greater understanding of the characteristics, behaviour, and future needs of their LV networks. This will be based on the analysis of data gathered by appropriate monitoring schemes to be deployed on hundreds of LV feeders and substations, and the assessment of the corresponding computer-based network models in current and future scenarios.

1.1 Deliverable 1.2

Given the volume of network data to be managed and the need of carrying out steady-state analysis (i.e., snapshot and time-series power flows) network models have to be created in a pseudo-automatic fashion. Accordingly, Deliverable 1.2 was defined as "Tool for Translating Network Data from ENWL to OpenDSS". This report describes the methodology used to translate ENWL GIS-based network data into a format ready to be used by the distribution network simulation package OpenDSS (developed by the Electric Power Research Institute, USA).

1.2 General Description

The modelling tasks will be performed using OpenDSS. As any other simulation tool, this software package requires detailed technical data. However, the volume of data that will be required from ENWL (~200 distribution transformers and ~1000 feeders) is significant to be translated manually. Consequently, an automatic process is required in order to read the ENWL GIS-based data, to check/improve the connectivity of the network, and to finally translate it into the OpenDSS format in order to be able to run power flow simulations.

From the initial data analysed (Pavilion Way area) it was evident that the ENWL database for low voltage distribution networks has connectivity problems. In fact, many line segments are not connected to the main LV feeder, making it impossible to be directly used for power flow calculations. This connectivity issue and the corresponding solution will be explained and presented in this report. The solution process has already been successfully applied to 46 low voltage feeders.

Additionally, once all the segments of a given feeder are totally connected it is possible to translate the corresponding information into OpenDSS files. The automatic process to achieve this is also explained in this report.

Finally, to demonstrate the simulation capabilities of OpenDSS once the GIS files are adequately translated, some preliminary results from time-series simulations are presented.

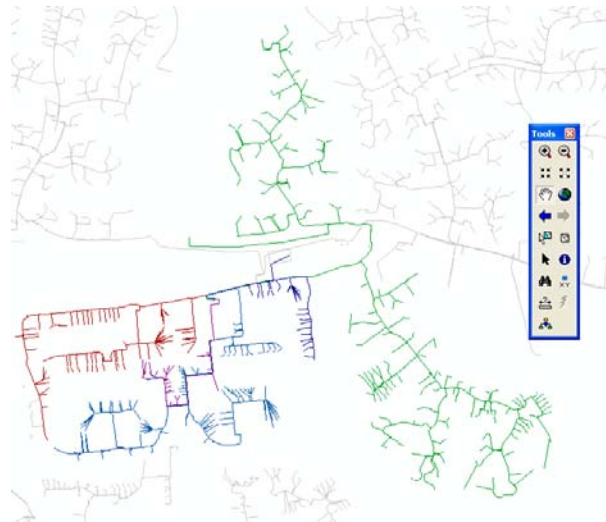


Figure 3: Pavilion Way feeders

Two important issues associated with the data base structure will be analysed in the next sections: the polyline representation of the information, and the connectivity problems among some segments and the main LV feeder.

2.2 Polyline representation of the main feeder

The GIS files deal with a large amount of information by using the concept of polyline to store the data. This representation does not allow us to have direct access to all information about each segment in the polyline.

Basically, the polyline is a continuous line comprised by one or more line segments. The polyline is treated as a single object within the GIS file and, therefore, there is no direct visualisation of the coordinates associated with one polyline. For example, the polyline in Figure 4 is represented just by one row in the attributes table (Table 1) and this row has just the coordinates for the source point and end point of the polyline, which is not enough to build the electrical model.

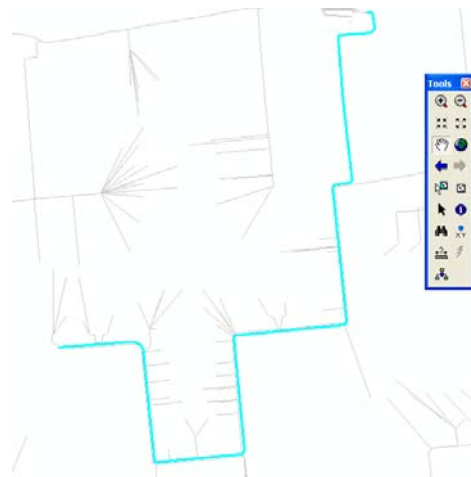


Figure 4: Polyline example from one Pavilion Way feeder

FID	Shape	WALL_TER_0	SUBSTATI_1	NAME_PLA_0	NAME	TRANSFOR_1	VARIABLE_0	TRANS
5357	Polyline							
5358	Polyline							
5359	Polyline							
5360	Polyline							
5361	Polyline							
5362	Polyline							
5363	Polyline							
5364	Polyline							
5365	Polyline							
5366	Polyline							
5367	Polyline							
5368	Polyline							

Table 1: Polyline information associated with the polyline example

In order to have access to all the segments of the polyline it is necessary to implement a procedure in ARCGIS 9.2. This procedure uses the following functions:

1. Split polyline to segments: The polyline is divided into segments. Each of them contains the whole information of the polyline, which means if the polyline has a "feature A", each segments has the same "feature A".
2. Split segments to vertices: This function indicates the vertices of each segment previously determined.
3. Add XY coordinates to each vertex: This function includes the coordinates of each vertex for each segment of the polyline.

Figure 5 exemplifies the application of the previous procedure over one low voltage feeder. The points indicate the position of the vertices of each segment. As we can see, the amount of data to be managed increases significantly (the polyline is divided into segments). In fact, the attributes table increases from 7533 segments to 57253 segments. This, however, is believed not to be an issue for OpenDSS.



Figure 5: Vertices of each segment of the polyline

2.3 Connectivity Issues

In order to carry out a power flow study for a LV network, every line segment in the circuit has to be connected. To check the connectivity in the GIS file, the connected component of each feeder needs to be determined. A connected component of a graph is a sub-graph in which any vertices are connected to each other by paths (by definition one connected component is not connected to another connected component).

With the purpose of determining the connected components, the Breath First Search Algorithm has been implemented in this report.

2.3.1 Breath First Search Implementation

The Breath First Search is a graph search algorithm that begins at the root node and explores all the neighbouring nodes. Then, for each of the latter nodes, it explores their (unexplored) neighbouring nodes, and so on, until it finds every node connected. This is the basic structure of the algorithm:

1. Enqueue the root node.
2. Dequeue a node and examine it.
3. If the element sought is found in this node, quit the search and return a result. Otherwise, enqueue any successors (the direct child nodes) that have not yet been discovered.
4. If the queue is empty, every node on the graph has been examined. Otherwise, repeat from Step 2.

An example of the application of this algorithm is shown in Figure 6.

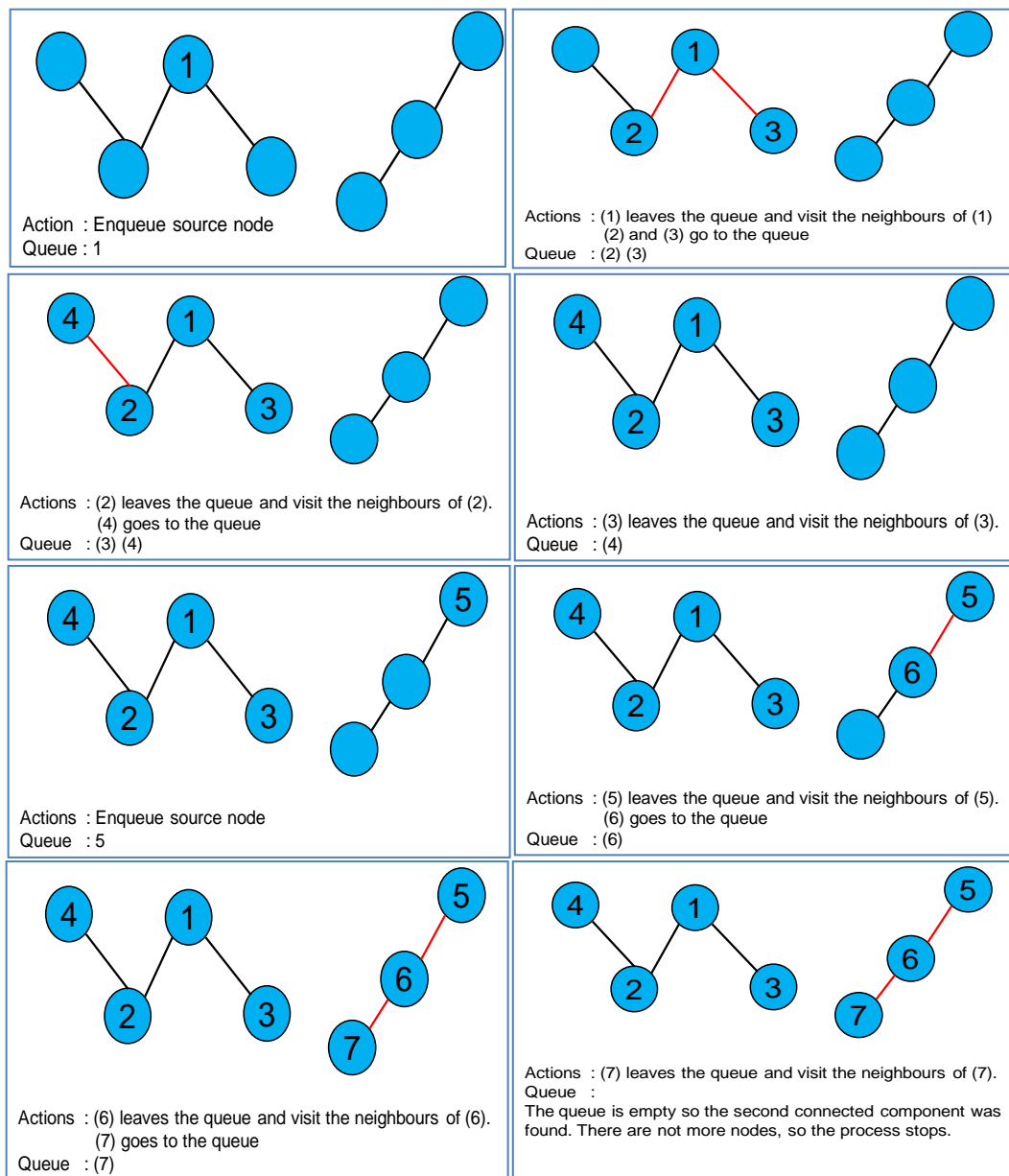


Figure 6: Example of Breath First Search Implementation

This algorithm was implemented in MATLAB and applied to the GIS-based information (now as a file readable by MATLAB). The main result indicates that there are many segments that are not connected to the main low voltage feeder. Figure 7 shows the connected components in one feeder of the Pavilion Way network (each colour represents one connected component). There are fifteen connected components; therefore, there are fourteen groups of segments without connection with the main LV feeder (blue line in the figure below).

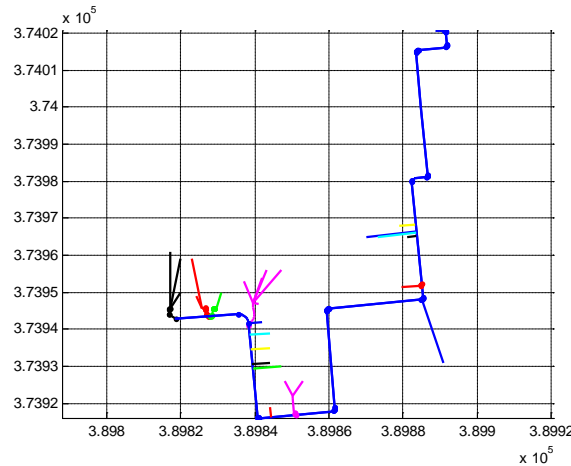


Figure 7: Connected components in one feeder of the Pavilion Way network

To validate the previous results, all the connection problems were checked in the raw GIS files (i.e., same files sent by ENWL without any modifications). In those files it was possible to appreciate connection problems in the scale of centimetres. For example, two points seem to be connected but in reality they are separated by 1.5 cm, or one point and the main feeder seem to be connected but in reality they are separated by 1 mm. These two cases are shown in Figure 8.

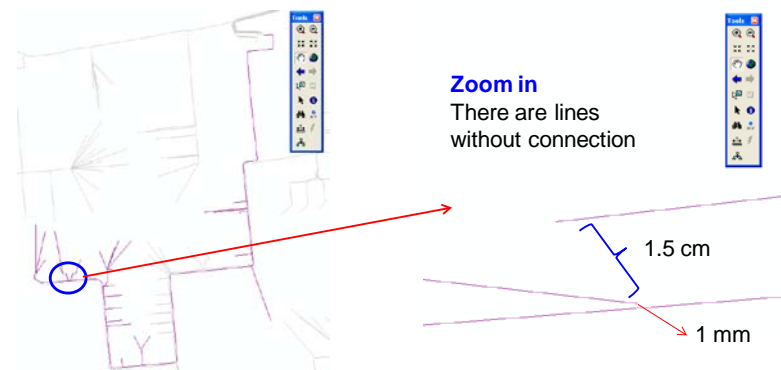


Figure 8: Example of connectivity problems

2.3.2 Breath First Search Application

To measure the problem magnitude, the algorithm developed in the previous section is applied to the whole Pavilion Way area. This means that every connected component of each feeder is determined. In particular, Figure 9 shows the connected components of the five feeders of the Pavilion Way network. It is possible to see that the number of connectivity issues is significant.

To have a better idea about how big the problem is, the connectivity index is defined. This index is the percentage of the segments (using the length) which is connected to the main feeder (the main feeder is defined as the longer connected component in one feeder). Taking into account just the networks inside of the snapshot (without border networks), the connectivity index for the LV networks received is equal to 0.58, which means that only the 58% of the total kilometres are connected to the main feeder.

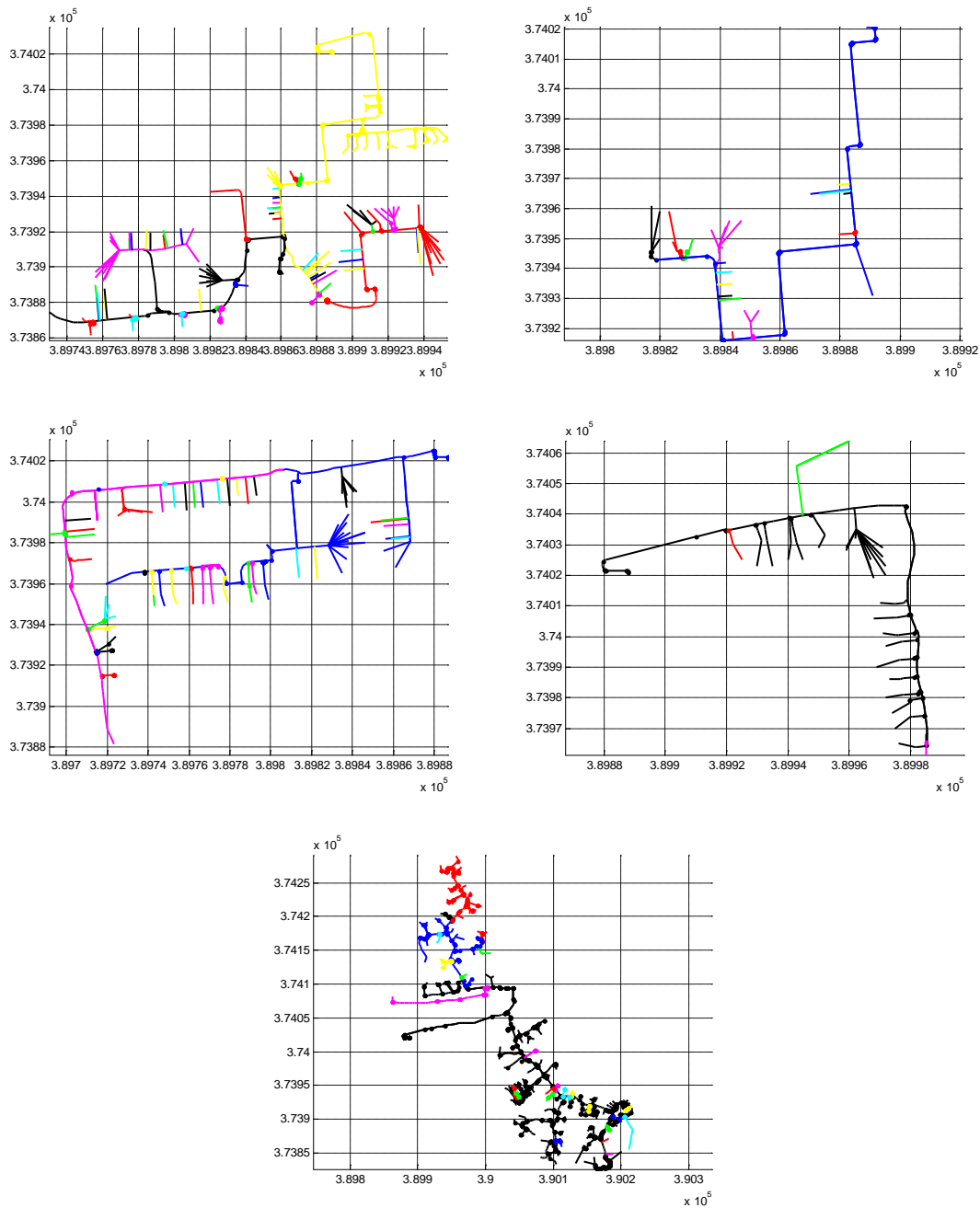


Figure 9: Connected components for the Pavilion Way network feeders

It is important to remark that the number of feeders analysed is 46 and the connectivity index is the aggregation of all of them. Thus, if each connectivity index is calculated for each feeder, it will be possible to see some feeders with a connectivity index lower than 0.58 and others with connectivity indexes higher than 0.58. The histogram of the connectivity index per feeder is shown in Figure 10 (the mean value is equal to 0.72).

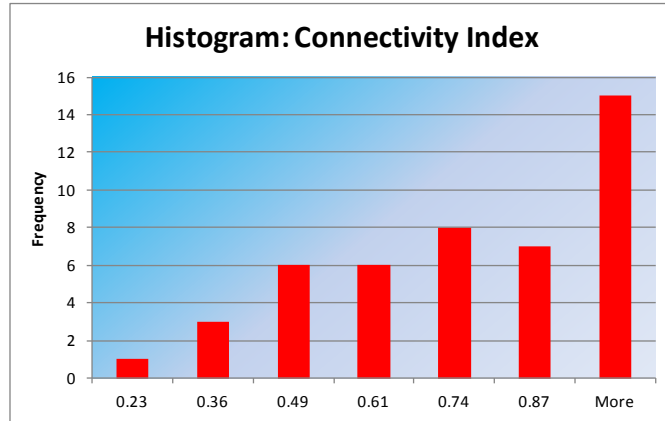


Figure 10: Histogram of connectivity index per feeder

Finally, the networks totally connected are identified in Figure 11.

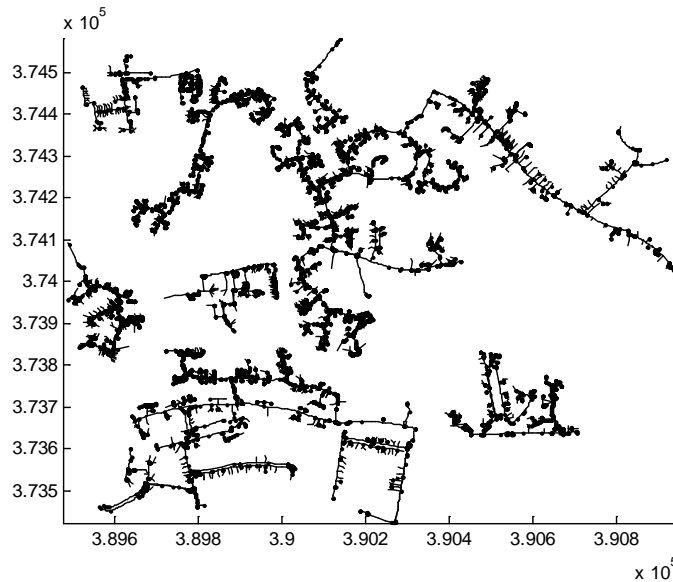


Figure 11: Networks totally connected (connectivity index = 0.58)

The connectivity problem described above is a serious issue if power flow studies are required. Since many segments (and the loads associated to them) could be neglected, the results of the simulations would be distant from reality. In fact, if the original data is used to run a power flow, only the 58% of the network will be simulated. An algorithm to solve this problem is proposed in the next section.

3 Reconnection Process

3.1 Problem Identification

To achieve successfully the reconnection process it is necessary to identify the type of problems in the database. Two main problems were identified from the analysis of the connected components. These are:

3.1.1 Vertex to Vertex

These are closer vertices which should be connected (and seem connected) but they are separated by a couple of centimetres. This is the case of vertex A and vertex B in Figure 12 (the separation is less than two centimetres).

3.1.2 Vertex to Line

These are vertices close to the main feeder which should be connected (and seem connected) but they are separated by a couple of centimetres. In this case, there is an additional problem because the main feeder does not have a vertex to join with the closer one. Therefore, a new vertex needs to be created in this case. This is the case of the red line and the blue line in Figure 12, where the connection point in the blue line, vertex C, must be included in the network structure to incorporate the connection between vertex B and vertex C.

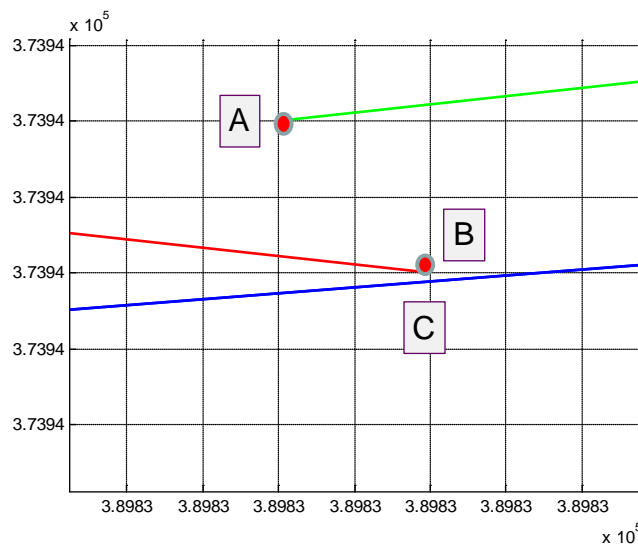


Figure 12: Problems Identification

3.2 Reconnection Methodology

To face the problems previously highlighted, a reconnection algorithm is developed and applied to the Pavilion Way area. The reconnection methodology is divided into the following steps (applied sequentially):

1. Connection of vertices;
2. Determination of connected components; and,
3. Reconnection of connected components.

The first one faces the vertex to vertex problem, the second one determines the connected components, and the last one connects the different connected components to the main connected component in order to have a totally connected network. These stages are developed in the next sections.

3.2.1 Connection of Vertices

The aim of this part of the algorithm is to connect all the vertices that are close enough to belong to the same connected component. This procedure follows the next steps:

1. Identification of closest vertices: the distance among all the vertices of the feeder with all the vertices of the feeder is calculated. All the vertices closer than 5 centimetres are selected (all the cases with zero distance are neglected because they represent the distance between the vertex and itself).
2. Connection of those nodes: Every couple of closest vertices selected in the previous stage are connected. Because their distance is very small instead of using a new line connection between the vertices, the position of the closest vertices is changed. In fact, the new position of the vertices is the middle point between the vertices. For example, vertex A and vertex B in Figure 13 are moving to the new position A'.
3. Elimination of the old vertices from the network structure: The closest nodes are eliminated because they are replaced by the new node. In the example, the vertices A and B are replaced by the new vertex A'.
4. Incorporation of the new vertex to the network structure: Besides the aggregation of the new vertex (with its new position), the structure of the network needs to be modified. That must be done because, before the connection, the closest nodes belong to certain line segments. Therefore, those segments after the connection are not any more connected to the closest nodes. Instead, they now need to be connected to the new vertex. For instance, in Figure 13, the green line ends in vertex A and the red line ends in vertex B. After the connections, both lines end in vertex A'. Hence, all of these changes are incorporated to the network structure.

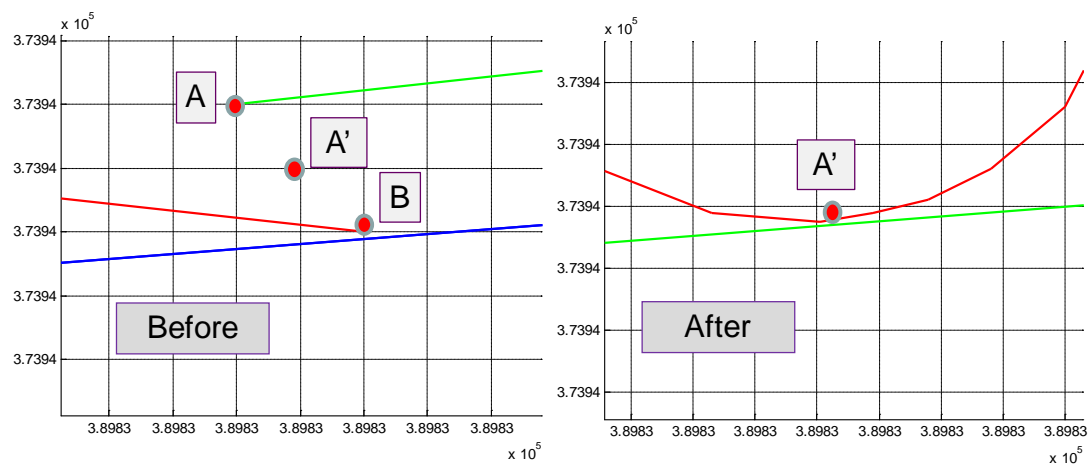


Figure 13: Connection of the closest nodes

3.2.2 Determination of connected components

The previous algorithm was applied to the Pavilion Way area. After that, the Breadth First Search algorithm is run over the modified database (ENWL database with the closest vertices connected) to determine the connected components. Due to the connection among the closest vertices, the number of connected components decreases in comparison with the original case (without the reconnection of vertices). As an example, Figure 14 shows one Pavilion Way feeder before and after the connection of vertices. It is possible to appreciate that the number of connected components decreased significantly, i.e., the connectivity issue has been reduced.

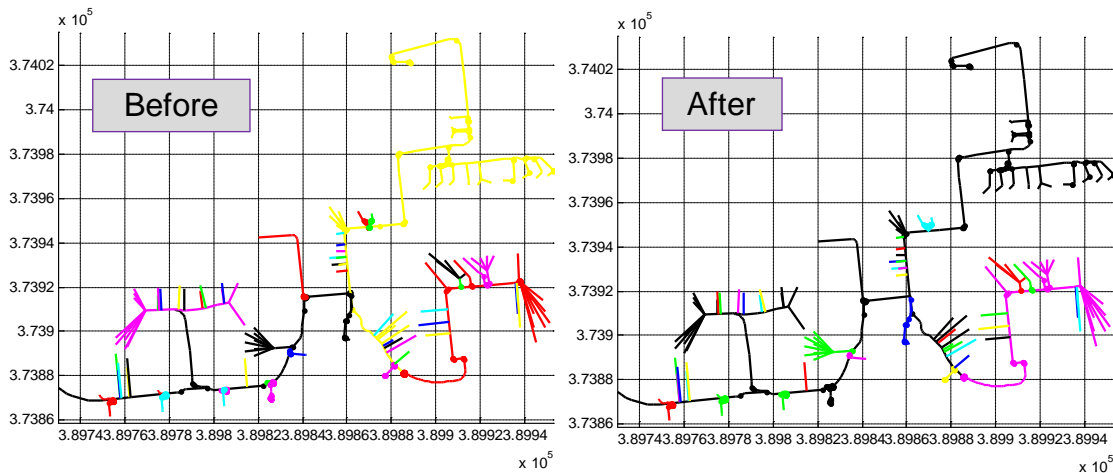


Figure 14: Application of the algorithm to one feeder of the Pavilion Way network

Once all the closest vertices are connected and the connected components are determined, it is fundamental to reconnect all the connected components among each other in order to achieve LV feeders totally connected.

3.2.3 Reconnection of connected components

This is the last stage of the proposed methodology. Its aim is to reconnect the connected components in each LV feeder. The flow chart of this stage is presented in Figure 15.

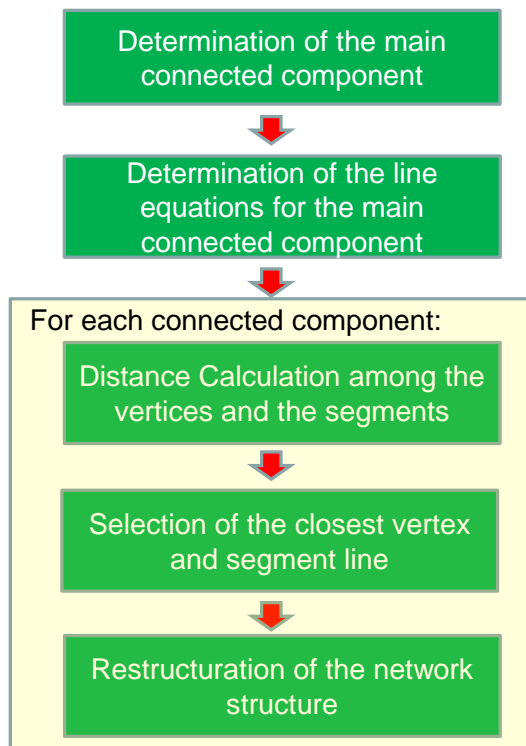


Figure 15: Flow chart of the reconnection process

To clarify the algorithm application, the explanation of each of the previous blocks is given in the next paragraphs by using a network example as a base case. This example has three connected components and is depicted in Figure 16.

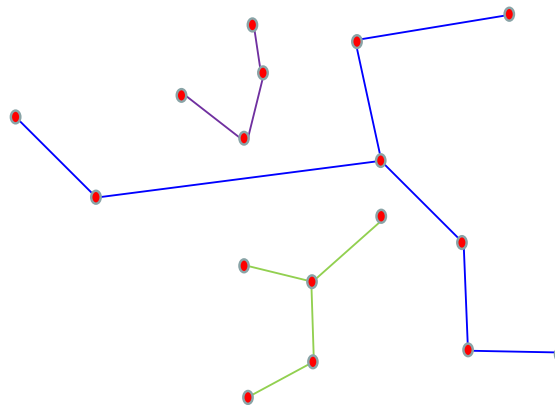


Figure 16: Network example with 3 connected components

The next steps are applied to each LV feeder, one by one, in an automatic way.

1. Determination of the main connected component: From the previous stage of the methodology all the connected components were calculated. The total length of each connected component is stored, thus, the main connected component (the longest one) is determined. Hence, the black connected component is the main connected component in the network example (Figure 17).

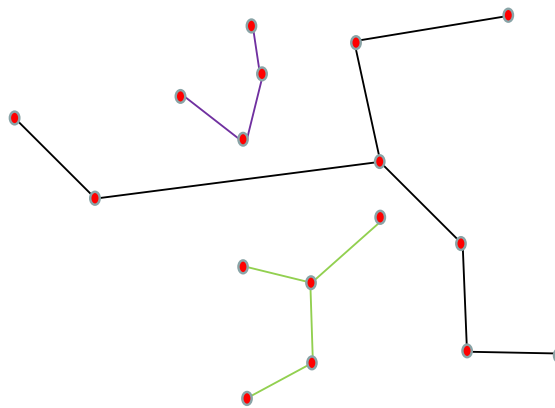


Figure 17: Main connected component (black line)

2. Determination of the line equations for each segment of the main connected component: To determinate the closest vertices from one connected component to the main connected component, it is required to apply basic analytical geometry. In fact, the problem to solve is the distance between one point (vertex of the connected component) and one line (segment line of the main connected component). Therefore, it is necessary to calculate the line equation for each segment of the main feeder (black line in Figure 17). A tolerance value is defined with the objective of avoiding too many calculations. Thus, the line equation is calculated just in the cases where the segments are longer than 1 meter. For all these segments, the equations in the format $Y=mX+n$ are determined. To have an idea about the size of this problem, the Pavilion Way network has 1420 segments bigger than 1 meter. Figure 18 indicates the lines (red dotted lines) for each segment of the main feeder.

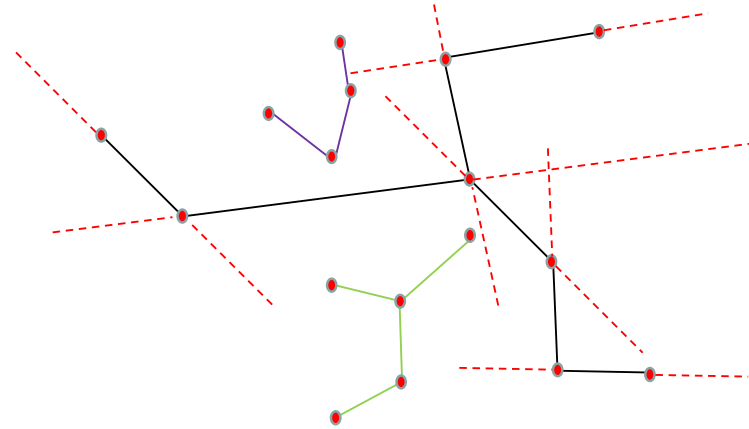


Figure 18: Equations determination for the main feeder

The next three steps are made for each connected component different to the main connected component of the LV feeder. For example, in Figure 16 there are three connected components but one of them is the main connected component. So, the following process is applied to the other two.

3. Determination of the distance between each vertex of the connected component and each segment line of the main connected component: Using the line equations from step 2, it is possible to calculate the distance between all the vertices and all the feasible segments. Indeed, if the line equation is in the format $Y=mX+n$ and the vertex location is (X_a, Y_a) , the distance will be:

$$distance = \frac{|m * X_a - Y_a + n|}{\sqrt{m^2 + 1}}$$

This distance is calculated among all the vertices of the connected component and all the line segments of the main feeder. This situation is shown in Figure 19 (blue dotted lines).

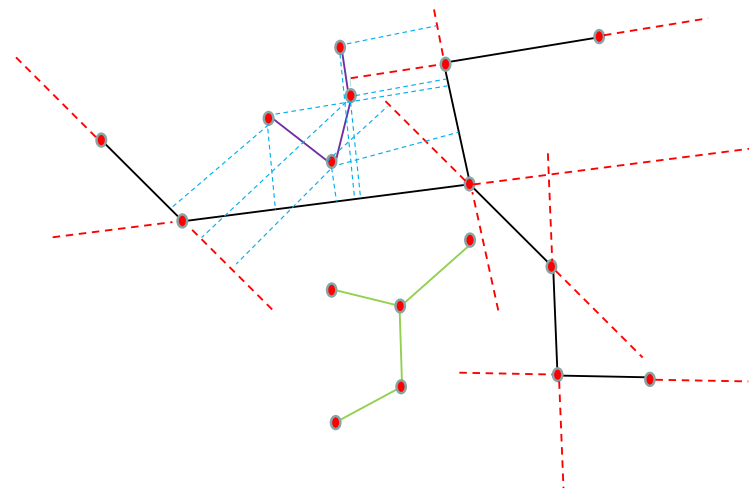


Figure 19: Distance determination

4. Selection of the closest vertex among the feasible vertices:
 - a. Determination of the new vertex on each segment line: The new vertex is the intersection point between the segment line of the feeder and the perpendicular line to the segment which passes through the closest point (connected component vertex).

This new vertex is calculated for all the feasible connections. As an example, Figure 20 shows the new vertex in each segment line (vertices from A to F).

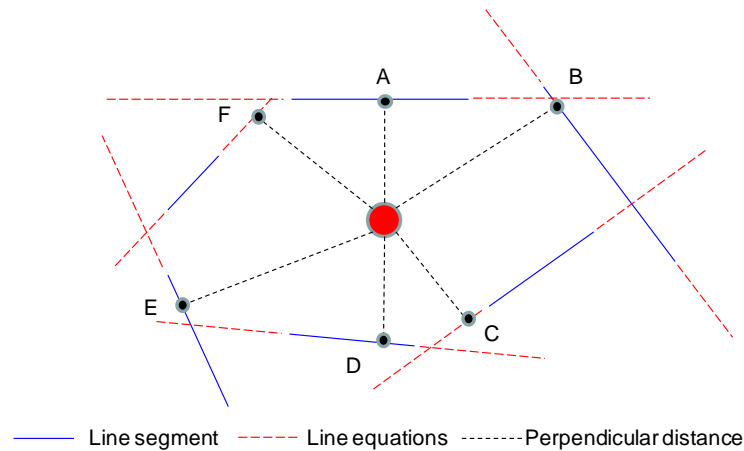


Figure 20: Example of feasible new vertices

- b. Feasible new vertices and connection: The distance is calculated between one point and one line equation, in this way it is possible that the intersection between the perpendicular line and the line equation is not inside of the segment line (the line equation has infinite length and the segment line has a limited one). For that reason, the feasible new vertices will be those vertices located along the segment line. Using Figure 20, it is easy to observe that some perpendicular distances (black lines) do not connect the node (red point) through the real limits of the segment (blue lines). Consequently, vertices F and C are not feasible.
- c. Determination of the connection: The final connection will be the shortest connection among the feasible solutions. In the case of the example, the shortest feasible solution is the solid red line shown in Figure 21.

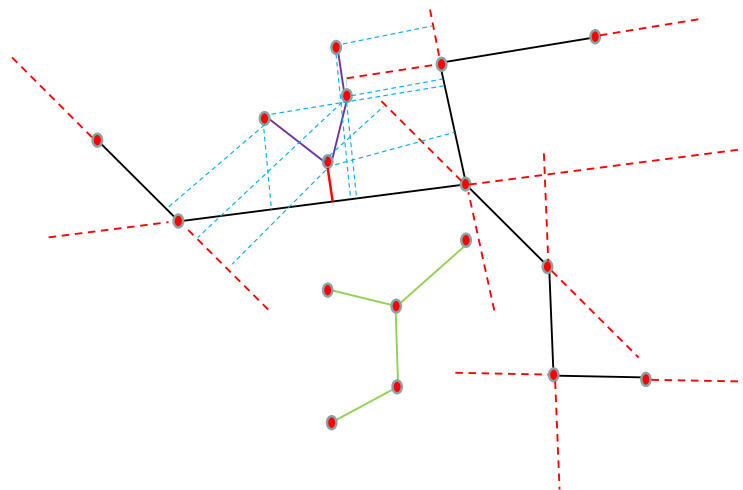


Figure 21: Example of feasible new vertices

Since the distance is too small (lower than 1 cm), instead of using a new connection line between the two vertices, the position of the closest vertices is changed to the position of the new vertex on the segment line. In this way, the closest vertices are merged in one vertex on the segment line (Figure 22).

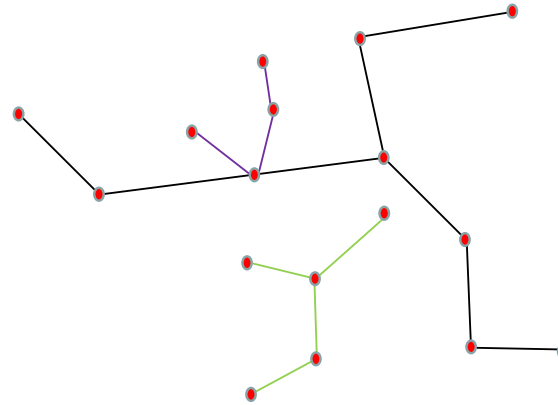


Figure 22: Example of feasible new vertices

5. Restructuration of the network structure: The incorporation of new vertices and new connections requires the elimination of old connections that are not available anymore from the database and the creation of the new connections. The main changes in the structure are:
 - a. The new vertex on the segment line needs to be incorporated to the network structure and the old one needs to be eliminated. The connection of the segments lines also need to be changed because the lines which start or end in the old vertex need to start or to end in the new vertex. For example in Figure 23, the segments which before included vertex D, now they need to include vertex C.

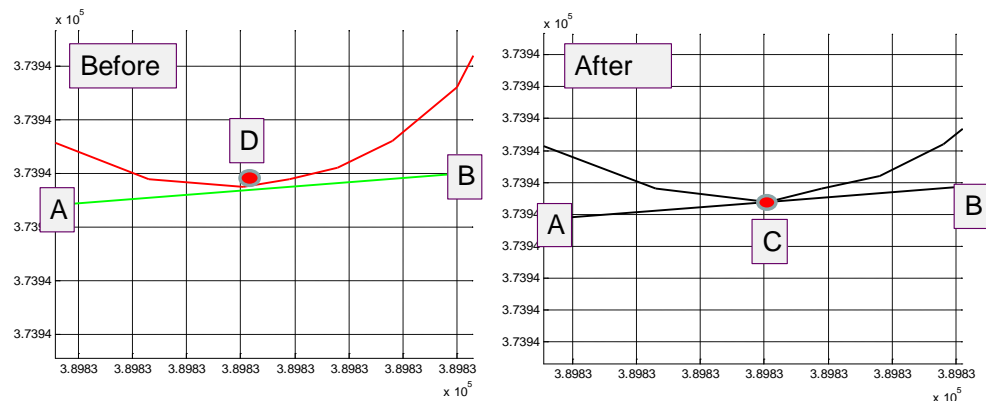


Figure 23: Connection between one vertex and the main feeder

- b. Split the segment line in order to incorporate the new position: This step is important because the incorporation of a new vertex in the segment line changes the network structure. For instance, Figure 23 shows that the segment line before the connection is defined by the vertices A and B. After the connection, the same line is defined by two segments, the first one from the vertex A to vertex C and the second one from vertex C to vertex B.

Finally, the steps 3 to 5 are applied to the rest of the connected components of the same feeder. In this way, all the segments are connected and therefore it will be possible to simulate the electrical behaviour of the network. The algorithm result for the network example is presented in Figure 24.

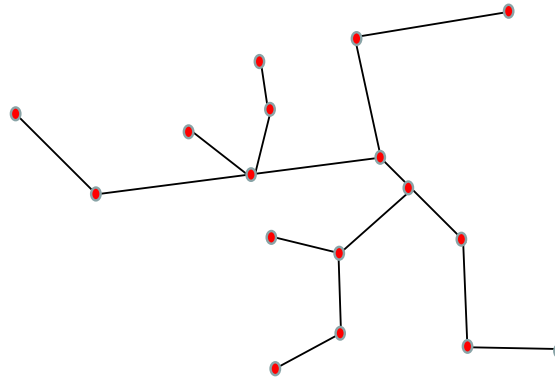


Figure 24: Connection between one vertex and the main feeder

3.3 Results of the Reconnection Methodology

The reconnection algorithm was successfully applied to all the Pavilion Way area. Some results are showed below.

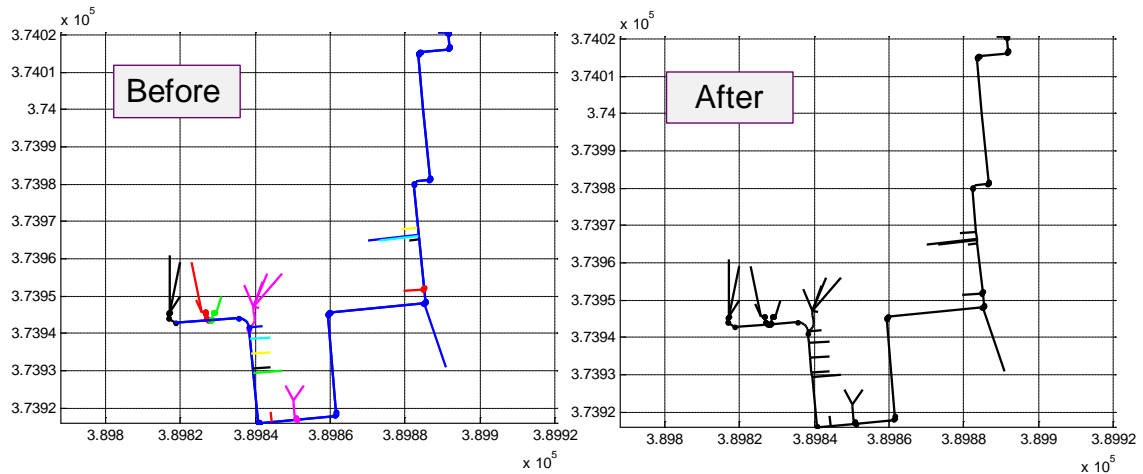


Figure 25: Application of the algorithm to one feeder of the Pavilion Way network

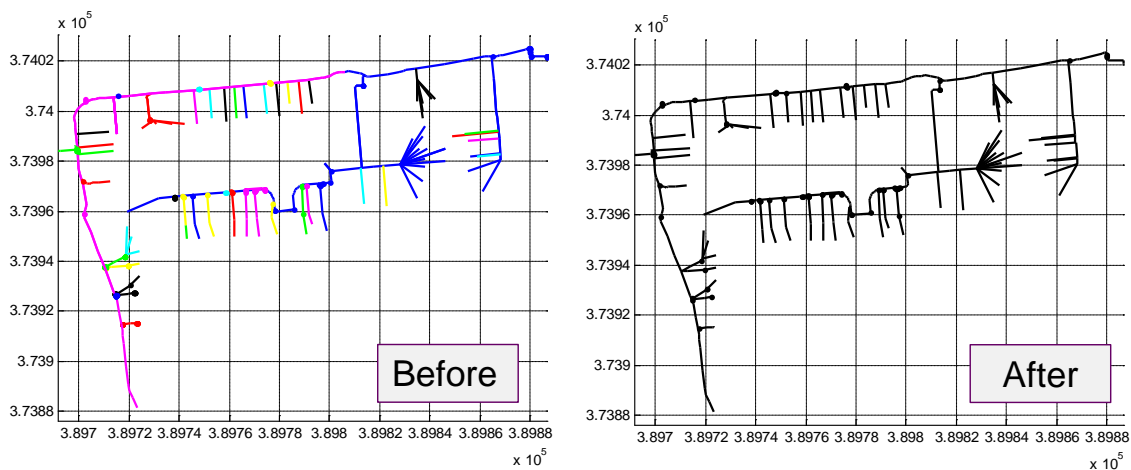


Figure 26: Application of the algorithm to another feeder of the Pavilion Way network

After the application of the algorithm, most of the networks are totally connected (i.e., Figure 25 and Figure 26). In fact, the connectivity index is now equal to 0.99, which means that 99% of the total kilometres are connected to the main feeder (the connectivity index before the application of the

algorithms was 0.58). This improvement is easy to observe in Figure 27. It allows us to visualise the network totally connected before and after the application of the reconnection methodology.

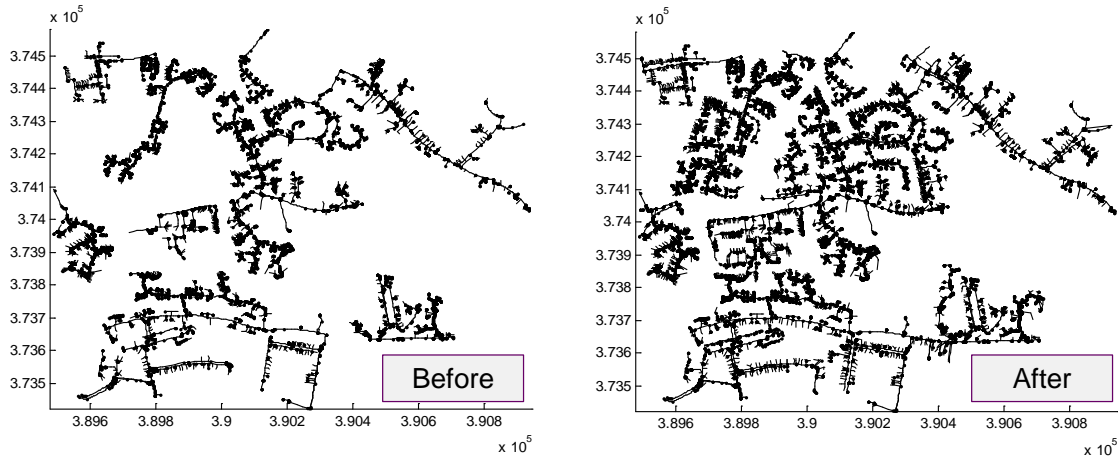


Figure 27: Network totally connected before and after the reconnection methodology

The histogram of the connectivity index calculated per feeder after the application of the algorithm is shown in Figure 28. 45 out of 46 feeders are totally connected as a result of the application of the reconnection procedure.

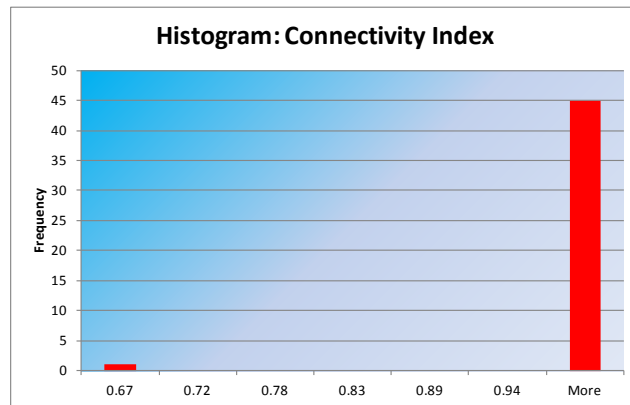


Figure 28: Connectivity Index per feeder after the reconnection methodology

There is just one feeder that is not 100% connected because the existence of a bigger distance (1 meter) between the main feeder and one connected component; this distance probably corresponds to one protection equipment that is not anymore in that position.

In conclusion, it possible to state that the reconnection methodology is able to reconnect all the connected components separated by a certain distance (threshold) from the main connected component.

4 From the connectivity process to OpenDSS

Once the network topology is totally connected, the next step to be able to run power flow simulations is to translate the processed information into the OpenDSS format.

4.1 OpenDSS Characteristics

OpenDSS is an open and free software package to solve power flows, harmonics analysis and fault current calculation in electrical distribution systems. This computational tool was developed by the EPRI to help the analysis associated with distributed generation. This software is able to solve unbalanced network and it can be driven from other software, such as VBA or MATLAB.

One of the main characteristics of OpenDSS is the ability to represent the time dimension (daily and yearly simulations with different time step) in networks with distributed generation, this is very important to measure the real impacts of intermittent sources (PV, micro-CHP, micron-wind turbines, etc.) and loads (EV, EHP etc.) on distribution networks. A visualisation of the software can be observed in Figure 29.

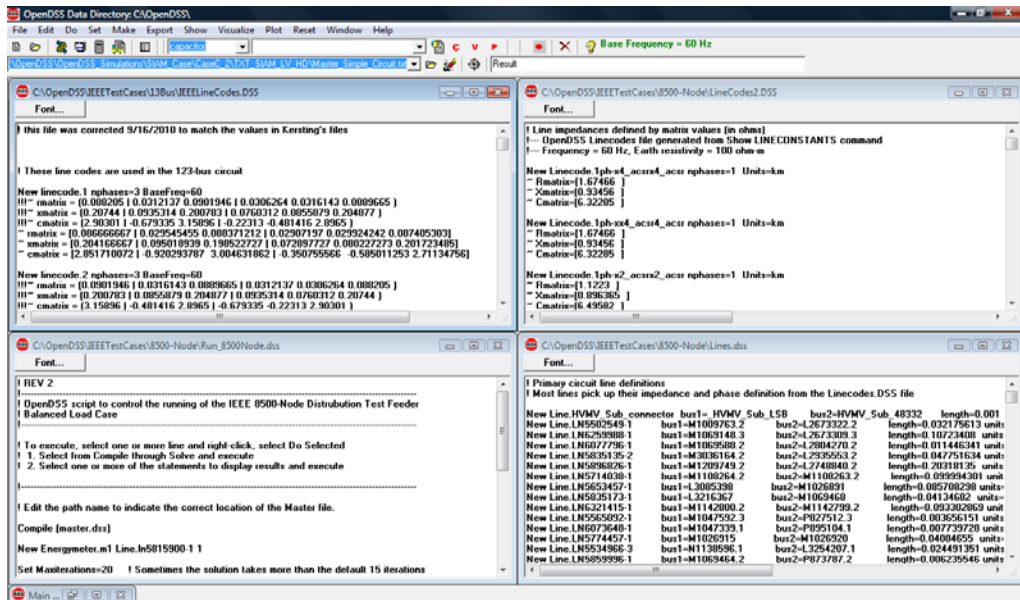


Figure 29: OpenDSS Visualisation

OpenDSS does not have a graphic interface. Instead, it uses its own programming language and each element must be incorporated by following certain instructions and codes. For example, Figure 31 summarizes the OpenDSS codes for the small network presented in Figure 30.

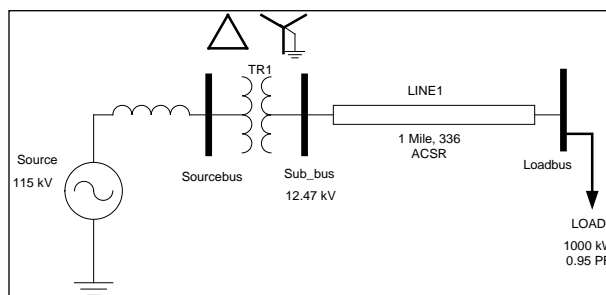


Figure 30: OpenDSS Visualisation

```

New Circuit.Simple      ! Creates voltage source (Vsource.Source)
Edit Vsource.Source BasekV=115 pu=1.05 ISC3=3000 ISC1=2500 !Define source V and Z
New Transformer.TR1 Buses=[SourceBus, Sub_Bus] Conns=[Delta Wye] kVs= [115 12.47]
~ kVAs=[20000 20000] XHL=10
New Linecode.336ACSR R1=0.058 X1=.1206 R0=.1784 X0=.4047 C1=3.4 C0=1.6 Units=kft
New Line.LINE1 Bus1=Sub_Bus Bus2=LoadBus Linecode=336ACSR Length=1 Units=Mi
New Load.LOAD1 Bus1=LoadBus kV=12.47 kW=1000 PF=.95

Solve

Show Voltages

Show Currents

Show Powers kVA elements
    
```

Figure 31: OpenDSS Visualisation

Consequently, to effectively use OpenDSS in this project it is necessary to translate the database files (after the application of the reconnection methodology) to OpenDSS format.

4.2 OpenDSS Representation

The typical structure for OpenDSS models is one master file that commands the reading and the executing of the rest of the files. These files are:

- LINECODE: this file has the technical information for each conductor.
- LOADSHAPES: this file has the variation of load profile for different type of customer.
- LINES: this file contains the information about the sending and receiving node, length and type of conductor for each line.
- LOADS: this file indicates the load shape associated with each customer and others technical specifications.
- TRANSFORMERS: this file has the technical information about the transformer feeder.
- MONITORS: this file allows us to monitor and store the data through distribution network over the time.

Therefore, the files presented above need to be built for each LV. Given the large number of networks to model, this process needs to be automatic. The flow chart of the corresponding automatic process is shown in Figure 32.

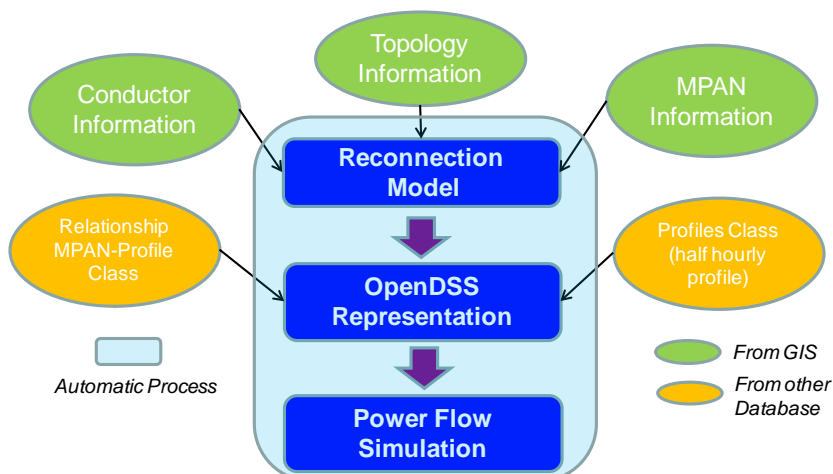


Figure 32: Automatic Translation to OpenDSS

From Figure 32 it is possible to appreciate that the information to build the OpenDSS files comes from different sources. The technical information comes from the ENWL GIS database. The topology information (connection between lines and vertices) comes from the database after the application of the reconnection methodology. The load information will be created by using the information from the measurements. However, to demonstrate the process, the load information from ENWL Half-Hourly Profiles, which are generated by Elexon will be used. These profiles are:

- Profile Class 1: Domestic Unrestricted Customers.
- Profile Class 2: Domestic Economy 7 Customers.
- Profile Class 3: Non-Domestic Unrestricted Customers.
- Profile Class 4: Non-Domestic Economy 7 Customers.
- Profile Class 5: Non-Domestic Maximum Demand (MD) Customers with a Peak Load Factor (LF) of less than 20%.
- Profile Class 6: Non-Domestic Maximum Demand Customers with a Peak Load Factor between 20% and 30%.
- Profile Class 7: Non-Domestic Maximum Demand Customers with a Peak Load Factor between 30% and 40%.
- Profile Class 8: Non-Domestic Maximum Demand Customers with a Peak Load Factor over 40%.

In particular, the half-hourly profiles received from ENWL are classified according to the Table 2¹. For each of these profile classes there is one daily half-hourly profile for the period 2010-2011.

Profile Class	LLFC	OFGEM DESCRIPTION
1	511	DOMESTIC UNRESTRICTED
2	531	DOMESTIC TWO RATE
2	581	DOM OFF PEAK (RELATED MPAN)
4	591	SML NONDOM O/P (REL MPAN)
3	631	SMALL NON DOM UNRESTRICTED
4	661	SMALL NON DOM TWO RATE
1 & 8	721	NHH UMS
5 to 8	751	LV MEDIUM NON DOMESTIC
5 to 8	752	LV SUB MEDIUM NON DOMESTIC
5 to 8	753	HV MEDIUM NON DOMESTIC
8	961	LV GENERATION NHH

Table 2: Profile Classes

As an example, the next four figures show the different ENWL profiles for April 1, 2010. The first one indicates the domestic profiles; the small and medium non domestic profiles are depicted in Figure 34 and Figure 35, respectively, and finally, the last one shows the special profiles (NHH UMS and LV generation).

¹ LLFC: Line Loss Factor Class

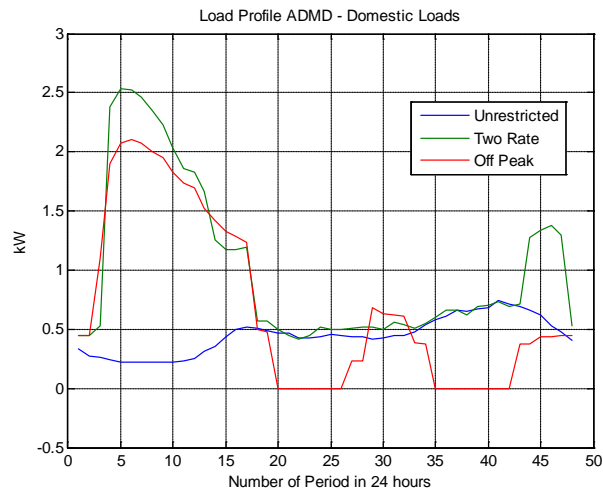


Figure 33: Domestic profiles

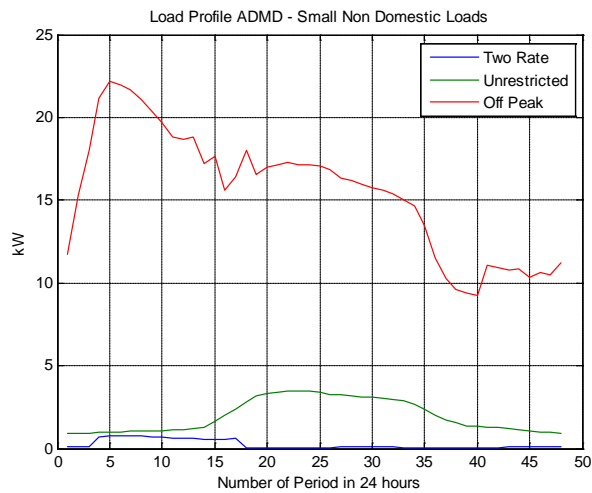


Figure 34: Small non-domestic profiles

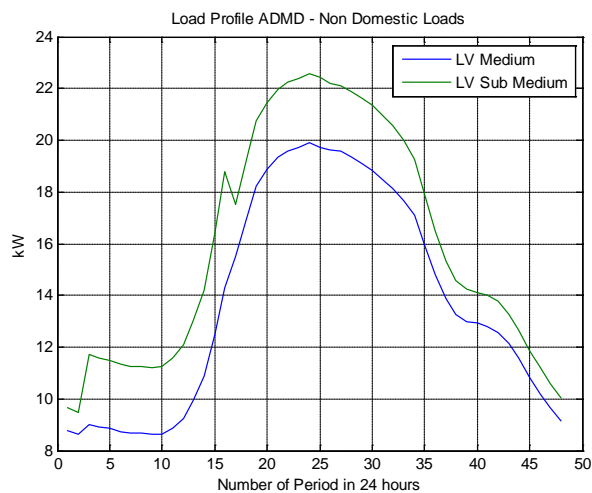


Figure 35: Medium non-domestic profiles

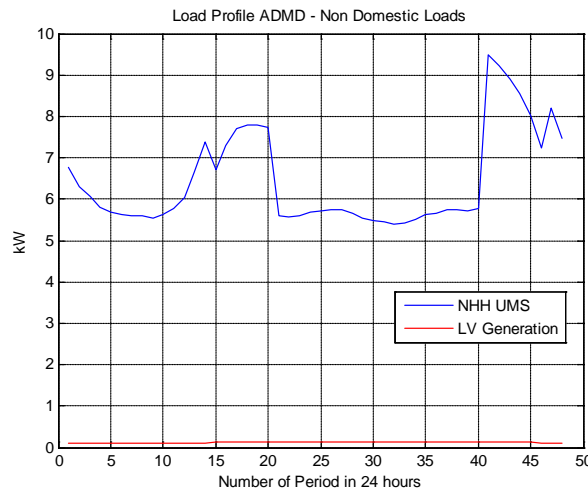


Figure 36: Special profiles

These profiles will be allocated to the different MPANs (Meter Point Administration Number) that are part of a given LV network.

With the topology information, the technical data, the profiles information and the relationship between the MPAN and the adopted load profiles, it is now possible to create an automatic process to translate all of the data into OpenDSS code. The files created in an automatic fashion for each feeder are:

- Lines.txt: After the application of the reconnection methodology, we know precisely which vertex is connected with which one. Also, from the data base, we know the type of cable and the distance between two vertices. With that information, the line file is created in MATLAB and the result is saved in notepad (OpenDSS can read files from this format).

```

Lines.txt - Notepad
File Edit Format View Help
New Line.LINE1 Bus1=1 Bus2=2 Linecode=185mm2 Length=1.1254 Units=m
New Line.LINE2 Bus1=2 Bus2=3 Linecode=185mm2 Length=5.4882 Units=m
New Line.LINE3 Bus1=3 Bus2=4 Linecode=185mm2 Length=0.13601 Units=m
New Line.LINE4 Bus1=4 Bus2=5 Linecode=185mm2 Length=0.1253 Units=m
New Line.LINE5 Bus1=5 Bus2=6 Linecode=185mm2 Length=0.59034 Units=m
New Line.LINE6 Bus1=6 Bus2=7 Linecode=185mm2 Length=0.1118 Units=m
New Line.LINE7 Bus1=7 Bus2=8 Linecode=185mm2 Length=0.21039 Units=m
New Line.LINE8 Bus1=8 Bus2=9 Linecode=185mm2 Length=0.11963 Units=m
New Line.LINE9 Bus1=9 Bus2=10 Linecode=185mm2 Length=0.13133 Units=m
New Line.LINE10 Bus1=10 Bus2=11 Linecode=185mm2 Length=0.24088 Units=m
New Line.LINE11 Bus1=11 Bus2=12 Linecode=185mm2 Length=0.22837 Units=m
New Line.LINE12 Bus1=11 Bus2=13 Linecode=185mm2 Length=0.23087 Units=m
New Line.LINE13 Bus1=11 Bus2=14 Linecode=185mm2 Length=1.8753 Units=m
New Line.LINE14 Bus1=12 Bus2=15 Linecode=185mm2 Length=0.21024 Units=m
New Line.LINE15 Bus1=13 Bus2=16 Linecode=185mm2 Length=0.89441 Units=m
New Line.LINE16 Bus1=14 Bus2=17 Linecode=185mm2 Length=14.5512 Units=m
New Line.LINE17 Bus1=14 Bus2=18 Linecode=185mm2 Length=5.6338 Units=m
New Line.LINE18 Bus1=15 Bus2=19 Linecode=185mm2 Length=0.27203 Units=m
New Line.LINE19 Bus1=16 Bus2=20 Linecode=185mm2 Length=14.814 Units=m
New Line.LINE20 Bus1=16 Bus2=21 Linecode=185mm2 Length=5.1545 Units=m
New Line.LINE21 Bus1=18 Bus2=22 Linecode=185mm2 Length=2.1869 Units=m
New Line.LINE22 Bus1=19 Bus2=23 Linecode=185mm2 Length=0.33541 Units=m
New Line.LINE23 Bus1=21 Bus2=24 Linecode=185mm2 Length=13.2567 Units=m
New Line.LINE24 Bus1=21 Bus2=25 Linecode=185mm2 Length=2.9151 Units=m
New Line.LINE25 Bus1=22 Bus2=26 Linecode=185mm2 Length=2.2895 Units=m
New Line.LINE26 Bus1=23 Bus2=27 Linecode=185mm2 Length=0.36401 Units=m
New Line.LINE27 Bus1=25 Bus2=28 Linecode=185mm2 Length=1.672 Units=m
New Line.LINE28 Bus1=26 Bus2=29 Linecode=185mm2 Length=0.76601 Units=m
New Line.LINE29 Bus1=26 Bus2=30 Linecode=185mm2 Length=2.0834 Units=m
New Line.LINE30 Bus1=27 Bus2=31 Linecode=185mm2 Length=0.65605 Units=m
    
```

Figure 37: Example of line file

- LOADS: This file indicates in which bus the loads are connected, the type of load (single phase or three phases), the nominal load voltage and the load power factor. Additionally, this file incorporates an important part to execute time-series analyses, which is the load shape (load profile) associated with each load. In this file, only the load shape names are incorporated, their characteristics are indicated in a different file. All these properties are from the original ENWL database and from the relationship between MPANs and load profiles. They are also automatically written in OpenDSS format.

```

File Edit Format View Help
New Load.LOAD1 Phases=3 Bus1=13 kV=0.412 kW=1 PF=0.93 Daily=Shape_1
New Load.LOAD2 Phases=3 Bus1=14 kV=0.412 kW=1 PF=0.93 Daily=Shape_2
New Load.LOAD3 Phases=3 Bus1=15 kV=0.412 kW=1 PF=0.93 Daily=Shape_3
New Load.LOAD4 Phases=3 Bus1=60 kV=0.412 kW=1 PF=0.93 Daily=Shape_4
New Load.LOAD5 Phases=3 Bus1=69 kV=0.412 kW=1 PF=0.93 Daily=Shape_5
New Load.LOAD6 Phases=3 Bus1=74 kV=0.412 kW=1 PF=0.93 Daily=Shape_6
New Load.LOAD7 Phases=3 Bus1=79 kV=0.412 kW=1 PF=0.93 Daily=Shape_7
New Load.LOAD8 Phases=3 Bus1=81 kV=0.412 kW=1 PF=0.93 Daily=Shape_8
New Load.LOAD9 Phases=3 Bus1=84 kV=0.412 kW=1 PF=0.93 Daily=Shape_9
New Load.LOAD10 Phases=3 Bus1=86 kV=0.412 kW=1 PF=0.93 Daily=Shape_10
New Load.LOAD11 Phases=3 Bus1=87 kV=0.412 kW=1 PF=0.93 Daily=Shape_11
New Load.LOAD12 Phases=3 Bus1=97 kV=0.412 kW=1 PF=0.93 Daily=Shape_12
New Load.LOAD13 Phases=3 Bus1=114 kV=0.412 kW=1 PF=0.93 Daily=Shape_13
New Load.LOAD14 Phases=3 Bus1=115 kV=0.412 kW=1 PF=0.93 Daily=Shape_14
New Load.LOAD15 Phases=3 Bus1=116 kV=0.412 kW=1 PF=0.93 Daily=Shape_15
New Load.LOAD16 Phases=3 Bus1=117 kV=0.412 kW=1 PF=0.93 Daily=Shape_16
New Load.LOAD17 Phases=3 Bus1=118 kV=0.412 kW=1 PF=0.93 Daily=Shape_17
New Load.LOAD18 Phases=3 Bus1=133 kV=0.412 kW=1 PF=0.93 Daily=Shape_18
New Load.LOAD19 Phases=3 Bus1=134 kV=0.412 kW=1 PF=0.93 Daily=Shape_19
New Load.LOAD20 Phases=3 Bus1=147 kV=0.412 kW=1 PF=0.93 Daily=Shape_20
New Load.LOAD21 Phases=3 Bus1=158 kV=0.412 kW=1 PF=0.93 Daily=Shape_21
    
```

Figure 38: Example of load file

- LoadShapes.txt: this file provides the electrical behaviour of one load along one determined period. In this report, that period corresponds to one day with a half-hourly resolution. This file includes the path where the profile (shape) is located. Each profile is located in one excel file.

```

File Edit Format View Help
New Loadshape.Shape_1 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_2 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_3 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_4 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_5 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_6 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_7 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_8 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_9 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Load
New Loadshape.Shape_10 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_11 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_12 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_13 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_14 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_15 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_16 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_17 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_18 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_19 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_20 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
New Loadshape.Shape_21 npts=48 interval=0.5 mult=(file=d:\2012\Models\ENWL\Connectivity_to_OpenDSS\Connectivity\Load_files_ENWL\Loa
    
```

Figure 39: Example of load shape file

- LineCode.txt: This file incorporates the network parameters. The name of these conductors is the same name used in the Lines.txt file.

```

File Edit Format View Help
New LineCode.95mm2 R1=0.32 R1=0.32 X1=0.069 R0=0.96 X0=0.207 C1=0 C0=0 Units=km
New LineCode.185mm2 R1=0.164 R1=0.164 X1=0.0685 R0=0.492 X0=0.2055 C1=0 C0=0 Units=
New LineCode.300mm2 R1=0.1 R1=0.1 X1=0.0675 R0=0.3 X0=0.2025 C1=0 C0=0 Units=km
New LineCode.2x185mm2 R1=0.082 R1=0.082 X1=0.0343 R0=0.246 X0=0.1029 C1=0 C0=0 Unit
New LineCode.2x300mm2 R1=0.05 R1=0.05 X1=0.0338 R0=0.15 X0=0.1014 C1=0 C0=0 Units=ki
New LineCode.3x300mm2 R1=0.0333 R1=0.0333 X1=0.0225 R0=0.0999 X0=0.0675 C1=0 C0=0 U
New LineCode.25mm2_OH R1=0.32 R1=0.32 X1=0.069 R0=0.96 X0=0.207 C1=0 C0=0 Units=km
New LineCode.95mm2_OH R1=0.32 R1=0.32 X1=0.29 R0=0.96 X0=0.87 C1=0 C0=0 Units=km
New LineCode.185mm2_OH R1=0.164 R1=0.164 X1=0.28 R0=0.492 X0=0.84 C1=0 C0=0 Units=ki
New LineCode.300mm2_OH R1=0.1 R1=0.1 X1=0.276 R0=0.3 X0=0.828 C1=0 C0=0 Units=km
New LineCode.2x185mm2_OH R1=0.082 R1=0.082 X1=0.14 R0=0.246 X0=0.42 C1=0 C0=0 Units
New LineCode.2x300mm2_OH R1=0.05 R1=0.05 X1=0.138 R0=0.15 X0=0.414 C1=0 C0=0 Units=
New LineCode.3x300mm2_OH R1=0.0333333333333333 R1=0.0333333333333333 X1=0.092 R0=0.
    
```

Figure 40: Example of line code file

- TRANSFORMERS: this file has the technical transformer parameters.
- MONITORS: this file indicates the position where the monitors are located. The technical information from the network (i.e., voltages, currents, active power, etc.) can be obtained just in the locations with monitors. Therefore, the automatic process allocates monitors at least in each customer to check the voltage profiles and in the main lines to check thermal limits.

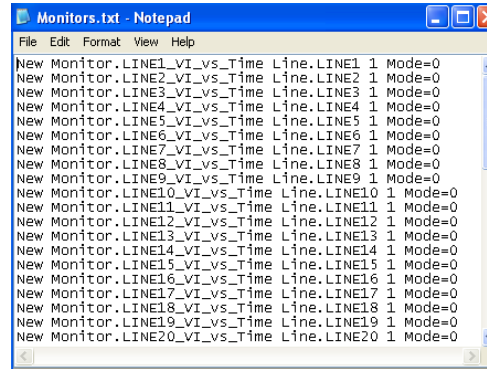


Figure 41: Example of line monitor file

4.3 OpenDSS Simulations

Once, the LV network is modelled in OpenDSS format it is possible to run power flow simulations. As an example, in this section we will develop a simulation from one particular feeder of the Pavilion Way network. This feeder was shown in a previous figure (Figure 7) and it is formed by 24 costumers. According to the MPAN information, fourteen of these customers have the profile class 2 (domestic two rate), nine of them have the profile class 1 (domestic un-restricted) and just one has the profile class 4 (small non domestic). The total length of the network is 418 meters and the number of vertices is 223 (therefore the number of network segments is 222).

As was stated before, the simulation is based on times-series analyses taking into account half-hourly profiles, so the number of periods in the simulation results is 48. The results are preliminary, but they allow us to explore the potential that times-series analyses have on low voltage distribution networks.

Figure 42 indicates the total power through the feeder (consumption plus losses), it is easy to observe the predominance of the two-rate customers in the energy consumption (the peak is produced during the night time). This figure also indicates that the maximum peak is 55 kVA and therefore the average peak demand in this feeder is 2.29 kVA.

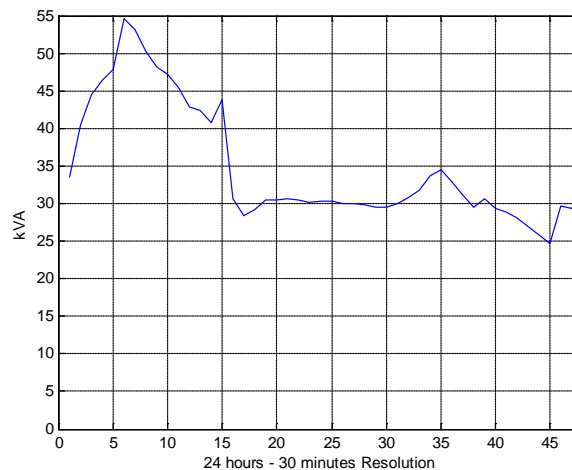


Figure 42: Power at the feeder head.

Considering a line-to-line voltage of 415V at the secondary of the distribution transformer, the voltage in the first conductor of the feeder is shown in Figure 43. The largest voltage drop is produced when the load requirements of the feeder are the highest. The voltage variation is not too high along the day. This is because it is the nearest conductor to the transformer and therefore the voltage regulation is better. In contrast, Figure 44 shows the voltage profile in the furthest conductor to the transformer, here we can appreciate that the voltage is lower and the magnitude of the voltage drop is much larger than closer to the transformer.

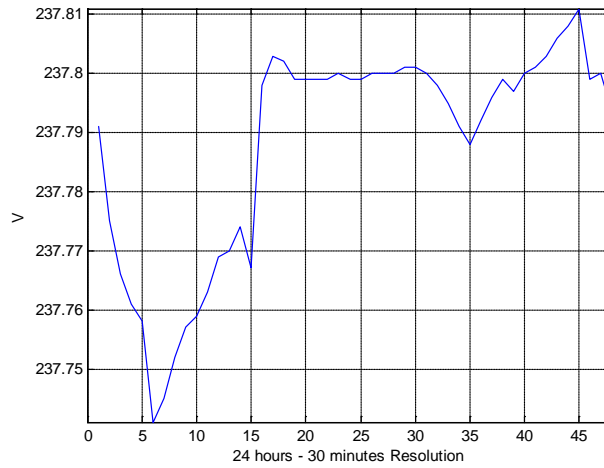


Figure 43: Voltage at the head of the feeder.

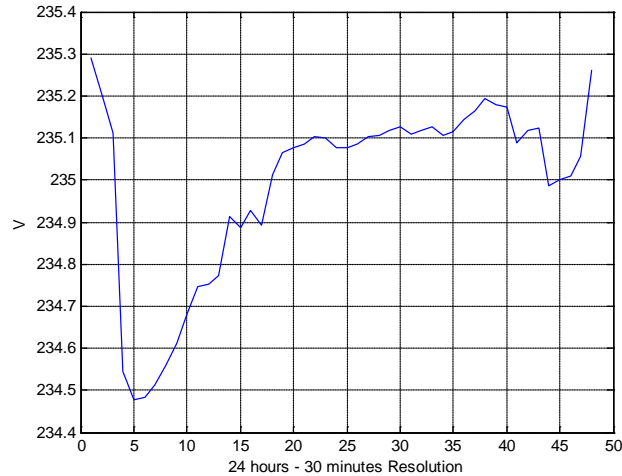


Figure 44: Voltage at the most remote customer.

The representation of the network in OpenDSS and the allocation of monitors allow us to examine the voltage profile of each of the customers in the feeder. Figure 45 shows the voltage profile in each of the 24 loads, where it is possible to appreciate that the voltage is going down from the first customer to the most remote one. The magnitude of the voltage drop increases as the distance to the transformer goes up. With the voltage profiles it is also possible to build the corresponding histogram (Figure 46) in order to realise which percentage of the time the feeder is within the limits. This information is useful and will be more useful when more low carbon technologies (e.g., photovoltaics, electric vehicles, electric heat pumps, etc.) are adopted by customers.

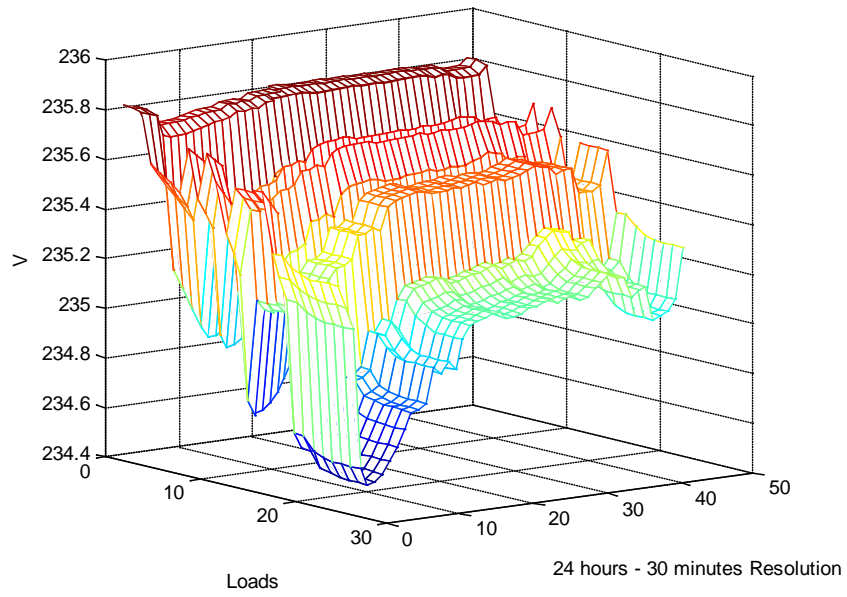


Figure 45: Voltage at the last customer.

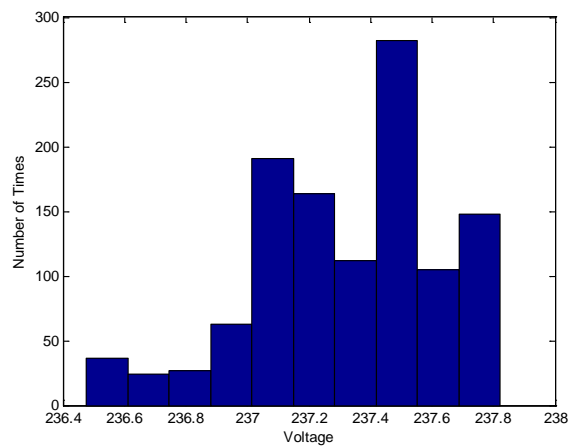


Figure 46: Voltage histogram of the feeder

5 Conclusion

In conclusion, Deliverable 1.2 "Tool for Translating Network Data from ENWL to OpenDSS" has been successfully completed.

The Pavilion Way network can now be used to perform time-series analyses with OpenDSS. All the sites where the monitoring devices will be deployed will be gradually processed and translated in the next few months.

With the arrival of data from the measurement devices, it will also be possible to start the validation of the corresponding network models.