Connection Imbalance in Low Voltage Distribution Networks

Lee James Thomas

School of Engineering

Cardiff University

A thesis submitted for the degree of

Doctor of Philosophy

3rd July, 2015

Contents

Abstract

On British electricity distribution networks, the phase to which single phase loads and generators are connected is, in most cases, unknown. There is concern that large imbalances in connection will limit the capacity of the network to support distributed generation as well as the electrification of heating and transport. The roll-out of Smart Metering in Britain, expected to be completed by the end of 2020, provides Distribution Network Operators with a means to predict the phase of single phase connections and more accurately assess the impact of increased distributed generation. This thesis examines these possibilities.

There are three main sections:

1. Development of a steady state LV feeder modelling program allowing for flexible definition of connection imbalance and suitable for use with a supercomputer.
2. Development of a stochastic method to assess the combined influence (on voltages, currents and losses) of connection imbalance and photovoltaic generation.
3. Creation of an algorithm for the prediction of phase connections using Smart Meter Data, based on the GB smart metering proposals.

The LV feeder model uses an unbalanced load flow based on network reduction and re-expansion with nodal analysis. It was validated using PSCAD. The feeder model uses a TNS earthing arrangement; this was shown to be equivalent to TN-C-S in normal operation, allowing for simpler modelling. A metric for connection imbalance was introduced – the highest proportion of houses connected to any phase. The model is capable of varying connection imbalance by changing the phase to which each house is connected.

The connection imbalance was varied by randomly allocating houses to different phases. Demand profiles were created stochastically and PV generation was added to a varied proportion of houses (0 to 100% in 10% steps). More than 19 million unbalanced load flow calculations were performed using a supercomputer. It was found that, for a typical urban feeder serving residential properties, connection imbalance is not a significant problem for DNOs until it becomes severe (>60% of houses on one phase).

The phase identification algorithm combines two methods found in the literature; voltage measurement clustering and solution of the subset sum problem. It uses

smart meter voltage profiles and active power profiles with current measured at the supply substation. It correctly predicted the phase connection for 97% of smart meters, using simulated data representing a set 100 different connection configurations, across 6 different days (different sets of demand profiles) with a measurement averaging timeframe of 30 minutes.

Declaration

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed..... (Candidate) Date.....

This thesis is being submitted in partial fulfilment of the requirements for the degree of PhD.

Signed..... (Candidate) Date.....

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed..... (Candidate) Date.....

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter- library loan, and for the title and summary to be made available to outside organisations.

Signed..... (Candidate) Date.....

Dedication

For Sophie

For Debbie

For Mervyn and Janet

Acknowledgements

I'd like to thank my supervisors, Professor Jianzhong Wu and Professor Nick Jenkins, for their guidance and patience.

For all the interesting and informative conversations, I'd like to thank to my many friends, colleagues and former-colleagues at CIREGS (Centre for Integration of Renewable Energy Generation and Supply) and across Cardiff University.

This work was performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University. Many thanks to the ARCCA staff for their help in understanding and accessing the facilities, particularly Thomas Green.

Thanks to all the developers of the free and open software tools (Including Python, MatPlotLib, NumPy, SciPy, Pandas, Notepad++, WinSCP, MinGW, IPython and SpyderIDE) I used for this research.

Thanks to the StackOverflow, SuperUser and Mathematics communities at stackexchange.com for the swift and useful answers to my questions.

Thanks most of all to my loving family for their steadfast patience and support.

List of Figures

List of Tables

Abbreviations

AC	Alternating Current
ADMD	After Diversity Maximum Demand
ADN	Active Distribution Network
AVC	Automatic Voltage Controller
BETTA	British Electricity Trading Transmission Arrangements
BSC	Balancing Settlement Code
BSCCo	Balancing and Settlement Code Company
CAD	Consumer Access Devices – in home devices with communication link to smart meters, under the GB specifications.
CDCM	Common Distribution Charging Methodology
CUSC	Connection and Use of System Code
CPLEX	IBM's proprietary optimisation software package.
CPU	Central Processing Unit
CSP	Communication Services Provider - company responsible for wide area communication within the GB smart metering system.
DCC	Data Communications Company
DDR3	Double Data Rate Type 3 - a type of SDRAM (Synchronous Dynamic RAM)
DECC	Department of Energy and Climate Change
DG	Distributed Generation/Generator
DNO	Distribution Network Operator
DR	Demand Response
DSO	Distribution System Operator
DSP	Data Services Provider – company responsible for data storage and access within the GB smart metering system.
DSTATCOM	Distributed Static Compensator

DUoS	Distribution Use of System. Refers to charges made by DNOs for the use of the distribution network.
EHV	Extra High Voltage
EMT	Electromagnetic Transients
EMTP	Electromagnetic Transients Program
EPRI	Electric Power Research Institute
EV	Electric Vehicle
FSB	Front Side Bus (connects processor to RAM)
GB	Great Britain
GSM	Global System for Mobile
HAN	Home Area Network
HiProp	The highest proportion of loads connected to a phase, on an LV distribution feeder.
HPC	High Performance Computing
HV	High Voltage
ID	Identification number
IHD	In Home Display
LCNF	Low Carbon Network Fund
LV	Low Voltage
MPI	Message Passing Interface
MPPT	Maximum Power Point Tracker
MV	Medium Voltage
NETSO	National Electricity Transmission System Operator
PV	Photovoltaic
Ofgem	Office of Gas and Electricity Markets
OLTC	On Load Tap Changer
PME	Protective Multiple Earth

PNB	Protective Neutral Bonding
RAM	Random Access Memory
RHEL6	Red Hat Enterprise Linux 6
RMS	Root Mean Square
RPM	Revolutions Per Minute
SATA	Serial Advance Technology Attachment (interface between disk and CPU)
SCADA	Supervisory Control and Data Aquisition
SEC	Smart Energy Code
SSD	Solid Sate Drive
SO	System Operator (see also NETSO)
SOP	Soft Open Point
SWA	Steel Wire Armoured
TN-C-S	Terra Neutral – Combined – Separate earthing arrangement
TN-S	Terra Neutral - Separate earthing arrangement
TT	Terra Terra earthing arrangement
TO/TNO	Transmission Operator/Transmission Network Operator
TUoS/TNUos	Transmission Use of System. Refers to charges made by TOs for the use of the transmission network.
UK	United Kingdom
VUF	Voltage Unbalance Factor
WAN	Wide Area Network – the communication network linking smart meters and the DCC under the GB smart metering specification.
XLPE	Cross Linked Poly Ethylene
ZIP	Impedance (Z), Current (I), Power (P) – refers to load models that include some portion of fixed impedance, fixed current and fixed power.

1 Introduction

1.1 Background

The UK government aims to bring about the creation of an electricity and gas smart metering system, including 53 million new smart meters for all domestic and small non-domestic properties, by the end of 2020 [1]. The data collected by, and the demand response functions of, the proposed smart metering system will give electricity network operators opportunity to improve the planning and operation of their networks.

The GB electricity network is expected to face significant changes, including an increase in peak demand, a growth in the electrification of transport (electric vehicles) and heating (heat pumps), continued growth in renewable sources (including distributed generation) and a reduction in system inertia [2]. This will require changes in approach to asset utilisation, system stability, and management of constraints (e.g. voltage and thermal). To address these changes, a more actively monitored and controlled system will be required, including an increased role for energy storage and demand response [3].

The consequent change in practice will be significant in distribution networks, where, traditionally, relatively little monitoring and automation takes place [4]. Some automated monitoring and control has already been deployed, much of it as part of centrally funded technology deployment trials [5]. Smart meters are, therefore, only one of a number of data monitoring and control systems that will be available to influence network operation and planning in future [3]. The term “smart grid” is widely used to refer to electricity networks and control structures that can cope with complexity brought about by increased monitoring and control.

The requirement to rigorously test new control ideas before deployment, and the growth in complexity of electricity network control, has lead to the use of larger computer models [6]. This has also been supported by the growth in computing resource generally available to researchers and engineers [7]. One aspect of this trend is the need to model demand in more detail, driven, in part, by a forecasted increase in demand side participation. Therefore, stochastic demand models, combined with load flow network models, are becoming more common [7].

In most cases, the exact configuration (the phase to which each household is connected) of the Low Voltage (LV) network is unknown [8]. This makes it difficult for network operators to accurately know the capacity of networks to support small scale

Photovoltaics (PV), the growth of which has been significant, see Figure 1.1 [9]. Therefore, there is a need to develop techniques for modelling of unbalanced LV distribution networks with realistic demand models – a need addressed in Chapters 3 and 4 of this thesis. Before this can be done in practice, however, the phase connection of all houses must first be known. A method for determining phase connections, using smart meters, as set out in Chapter 5, would therefore be valuable.

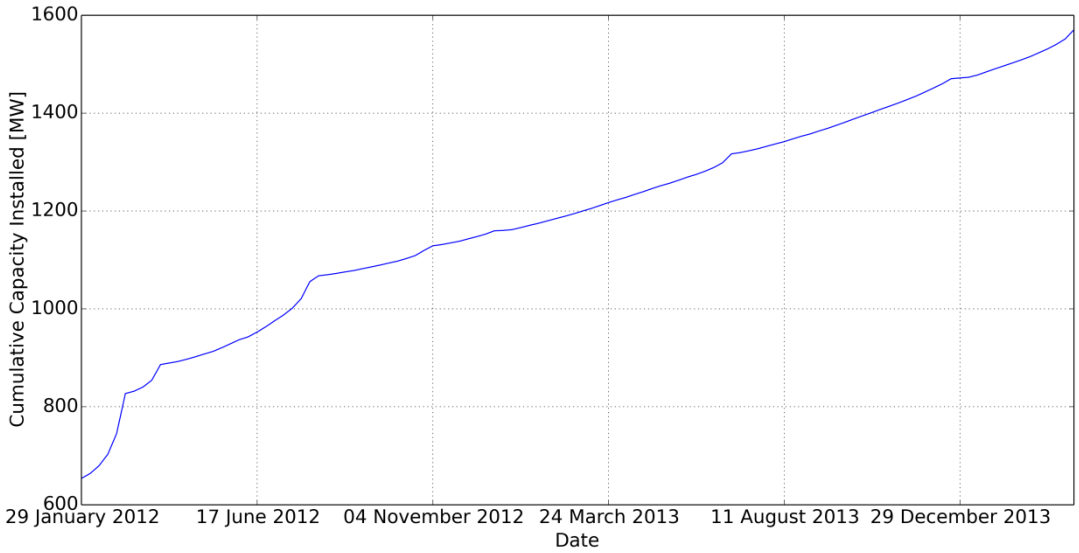


Figure 1.1 - Cumulative installed capacity of 0-10kW PV systems in Britain, data from [9]

1.1 Research Objectives

The aims of the research presented in this thesis were:

- To create and validate a Low Voltage feeder model allowing for stochastic variation of the phase to which houses are connected, suitable for use with a supercomputer.
- To establish a way to quantify single phase connection imbalance.
- To examine the relationship between imbalance in single phase connections and the amount of PV generation able to be connected to an LV feeder.
- To create an algorithm for identifying the phase connection of smart meters, based on the GB smart metering specification, and test it using data gained from the LV feeder model.

1.2 Contributions of the Thesis

1.2.1 Chapter 2 – Literature Review

A literature review was undertaken, covering the following areas:

- Overview of the GB electricity network,
- Distribution networks and distributed generation
- The proposed GB smart metering system
- Smart metering use cases for network operators
- Phase imbalance in distribution networks
- Representation of imbalance
- Modelling of unbalanced distribution networks
- Modelling of domestic demand
- Phase imbalance and distributed generation
- Phase identification

1.2.2 Chapter 3 - Modelling of an Unbalanced Low Voltage Feeder

The main contributions of Chapter 3 are as follows:

- Development of an unbalanced Low Voltage feeder modelling tool, suitable for large numbers of repeat runs on Cardiff University's Raven supercomputer.
- Development of a method for varying connection imbalance within a Low Voltage feeder model.
- Showed that the "highest proportion of houses connected to a phase" is a meaningful measure of connection imbalance, on a Low Voltage distribution feeder.
- Validated the LV distribution feeder model using PSCAD.
- Showed the equivalence of TN-S and TN-C-S (PME) earthing arrangements for the purpose of steady state modelling (allowing for a simpler model).

1.2.3 Chapter 4 - The Influence of Connection Imbalance and Solar Photovoltaics on an LV Feeder

The main contributions of Chapter 4 are as follows:

- Development of a method for the stochastic variation of connection imbalance.
- Performed over 19 million load flow calculations, using stochastic demand profiles and clear-sky solar PV profiles, to test the combined influence of connection imbalance with increasing levels of PV penetration on the voltages, currents and losses of the feeder.

- Concluded that connection imbalance does not cause meter voltage excursions beyond 230V+10%/-6% (on a typical urban feeder with optimal transformer tap setting) until phase connection imbalance becomes extreme, >75% houses on one phase. This drops to ~60% if all households are given 1.3kWp south facing clear-sky PV.

1.2.4 Chapter 5 - Using Smart Meter Data for Phase Identification

The main contributions of Chapter 5 are as follows:

- Combination of existing techniques (voltage clustering and subset sum) to form new phase identification method suitable for use with GB smart meters.
- Creation of simulated smart metering data from the output of Chapter 4
- Demonstration of the phase identification method using simulated GB smart meter data. In the test case with half-hourly measurements and 0% PV (600 test cases), 97% of meters were correctly identified.
- Showed that PV penetration causes little (<2% correctly identified meters) degradation to the algorithm.

1.2.5 Research Questions and Contributions - Summary

The three main identified research in the thesis with the related contributions, are shown in Table 1.1.

Research Question:	**Related Contribution:**
How can connection imbalance in LV networks be realistically modelled and tested?	A new method for varying and modelling connection imbalance on Low Voltage Feeders was developed.
How much does connection imbalance constrain the connection of Distributed Generation on typical residential LV feeders?	A stochastic method to assess the combined influence (on voltages, currents and losses) of connection imbalance and photovoltaic generation was developed. It was shown that, for a typical urban feeder serving residential properties, that, even with high levels of PV, connection imbalance is not a significant problem for DNOs until it becomes severe (>60% of houses on one phase).
Can Distribution Network Operators use GB Smart Meters to ascertain unknown phase connections?	Combination of existing techniques to form a new phase identification algorithm. The algorithm was demonstrated using simulated smart meter data (based on the GB specifications) and measurements at the substation. It correctly predicted the phase of ~97% of meters based on 600 test cases.

Table 1.1 - Thesis research questions and contributions.

1.3 Raven supercomputer

The research presented in this thesis made use of Cardiff University's Supercomputer, Raven [10]. The facility consists of 272 compute nodes (these are also referred to as blades – separate computers dedicated to research computing tasks). The compute nodes are categorised as Serial (also referred to as High Throughput Computers, HPC) or MPI (Message Passing Interface – referring to the parallel nature of the code that runs on them). There are a further 13 nodes dedicated to user access management and job scheduling. The hardware specifications are shown in Table 1.2 [10]. The Operating System is Bullxlinux 6.0 (based on RHEL6).

Node/Blade Category	**Quantity**	**Blade Model**	**Details**
MPI (Parallel) - 1	112	Bull B510	**CPU**: 2x Xeon E5-2670 2.6GHz - 1600FSB **Processor cores**: 16 **Microarchitecture**: Intel Sandybridge **RAM**: 64GB DDR3 **Disk**: 128GB SATA2 Flash SSD
MPI (Parallel with shared memory) - 2	16	Bull B510	**CPU**: 2x Xeon E5-2670 2.6GHz - 1600FSB **Processor cores**: 16 **Microarchitecture**: Intel Sandybridge **RAM**: 64GB DDR3 **Disk**: 128GB SATA2 Flash SSD
MPI (Parallel) - 3	12	Bull B520	**CPU**: 2x Xeon E5-2680v3 2.5GHz **Processor cores**: 24 **Microarchitecture**: Intel Haswell **RAM**: 256GB DDR3 **Disk**: 128GB SATA2 Flash SSD
Serial/HPC	60	Bull B520	**CPU**: 2x Xeon E5-2680v3 2.5GHz **Processor cores**: 24 **Microarchitecture**: Intel Haswell **RAM**: 128GB DDR3 RAM **Disk**: 256GB SATA2 Flash SSD
Serial/HPC	72	Bull B500	**CPU**: 2xXeon X5660-2.80GHz **Processor cores**: 12 **Microarchitecture**: Intel Westmere **RAM**: 48GB DDR3 **Disk**: 128GB SATA SSD
Login and Management	13	Bull R423	**CPU**: 2x Xeon E5-2650 2.0GHz - 8.00GT/s - 20MB - HT - Turbo+ - 95W - 1600 FSB **RAM**: 32GB RAM DDR3-1600ECC **Disk**: 2x 1000GB@7.2k RPM SATA2

Table 1.2 - Raven hardware specification overview

The facility was used to run 19 million load flow calculations and analyse the results (Chapter 4). This was done to assess the effect of connection imbalance and photovoltaics on the LV distribution network. The facility was also used, in Chapter 5, to demonstrate a new phase identification algorithm. The serial/HPC nodes were

used. In total, 19126.07 hours of computing time were booked on the system. However, some of this time was for familiarisation and trial/erroneous runs.

1.4 Publications by the author

1.4.1 International Research Conferences and Meetings

L. Thomas, A. Burchill, K. Samarakoon, Y. He, J. Wu, J. Ekanayake, and N. Jenkins, “Control of electricity networks using smart meter data,” in *CIGRE Session 2012*, 2012.

L. J. Thomas, J. Wu, J. B. Ekanayake, and N. Jenkins, “Enabling distributed frequency response using smart meters,” in *3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, 2012, pp. 1–5.

1.4.2 Research Position Papers

L. Thomas and N. Jenkins, “HubNet Position Paper Series - Smart Metering for the UK,” 2012. [Online]. Available: http://www.hubnet.org.uk/filebyid/191/SmartMetering.pdf

L. Thomas and N. Jenkins, “HubNet Position Paper Series - Smart Metering for the development and operation of the GB Power System,” 2014. [Online]. Available: http://hubnet.org.uk/filebyid/548/SmartMeteringForGB.pdf

1.4.3 Technical Reports

The IET Power Network Joint Vision (PNJV) Expert Group, “Electricity Networks Handling a Shock to the System - IET position statement and technical report,” 2013. [Online]. Available: http://mycommunity.theiet.org/energy/pnjv

M. Chaudry, A. Bagdanavicius, **L. Thomas**, R. Sansom, J. O. Calderon, N. Jenkins, and G. Strbac, “UKERC Energy Supply Theme Synthesis Report,” 2014. [Online]. Available: http://www.ukerc.ac.uk/publications/energy-supply-theme-synthesis-report.html.

2 Literature Review

2.1 Overview of the GB electricity network

The electricity network in Britain consists of the transmission network (above 132kV) and the distribution network (132kV and below, not including Scotland where 132kV is classed as part of the Transmission System [2]). The network is owned and operated by companies that have been granted monopoly licenses. These companies are regulated by the Office of Gas and Electricity Markets (Ofgem).

In its role as regulator, Ofgem oversees monopoly licenses for the ownership and operation of the transmission and distribution networks. Three companies own and maintain the transmission network; National Grid (England and Wales), Scottish Power Transmission Limited (southern Scotland) and Scottish Hydro-Electric Transmission Limited (SHETL, northern Scotland). The entire transmission system is operated by a single company – presently National Grid [11].

Distribution Network Operators (DNOs) are the licensed monopoly companies that own and operate the distribution network. The distribution network is divided into 14 licensed areas – these areas are presently divided amongst 6 companies; Electricity Northwest, Northern Power Grid, Scottish and Southern Energy, SP Energy Networks, UK Power Networks and Western Power Distribution.

As part of their license conditions, each licensed company must produce codes to define their principles and operating procedures. To this end, National Grid uses the Grid Code, the Balancing Settlement Code (BSC) and the Connection and Use of System Code (CUSC). The DNOs have jointly created the Distribution Code [12].

Generators connect to the network, subject to conditions made by the network operator, and sell their generated electricity on the market in advance. Suppliers are those companies that purchase the electricity on the market and re-sell it to consumers. Suppliers and generators can buy and sell electricity as they wish until one hour before the time of use – this moment is known as “gate closure”. In 2010, the GB generation capacity was approximately 80GW and the total demand ranged between approximately 19 and 58 GW [13].

The network owners and operators are paid for the use of their system by demand and generators (mainly via suppliers). These charges are known as Use of System (UoS) charges – specifically Distribution Use of System (DUoS) for distribution networks and Transmission Use of System (TUoS, also known as TNUoS –

Transmission Network Use of System) for transmission networks. The companies also charge for new connections to their network, based on the changes required to facilitate the connection. Under their license conditions, the companies must publish their methodology for the Use of System and connection charges. The DNOs use a common charging methodology (CDCM). The CDCM covers the DNO's system up to 132kV levels, a different method, the Common EHV Distribution Methodology (EDCM) is used for 132kV [14]. ll charging methodologies are overseen by the regulator, Ofgem.

As the demand cannot be exactly predicted by suppliers, there is always a mismatch between the power purchased from the market at gate closure and the power required to keep the supply and demand balanced. Therefore, one company (presently National Grid) is given responsibility for the balancing supply and demand – this company is known as the System Operator [15].

After gate closure, the System Operator performs the necessary actions (often requiring response from demand or generation) to ensure the system remains stable. This process is known as the Balancing Mechanism [16]. Most of the costs incurred for doing this are recouped via an imbalance settlement process managed by another licensed company – the Balancing and Settlement Code Company (BSCCo, presently Elexon). The entire process, including the open market and the Balancing Mechanism, is known as the British Electricity Trading Transmission Arrangements (BETTA) [17]. An overview of the BETTA market structure is shown in Figure 2.1 [18].

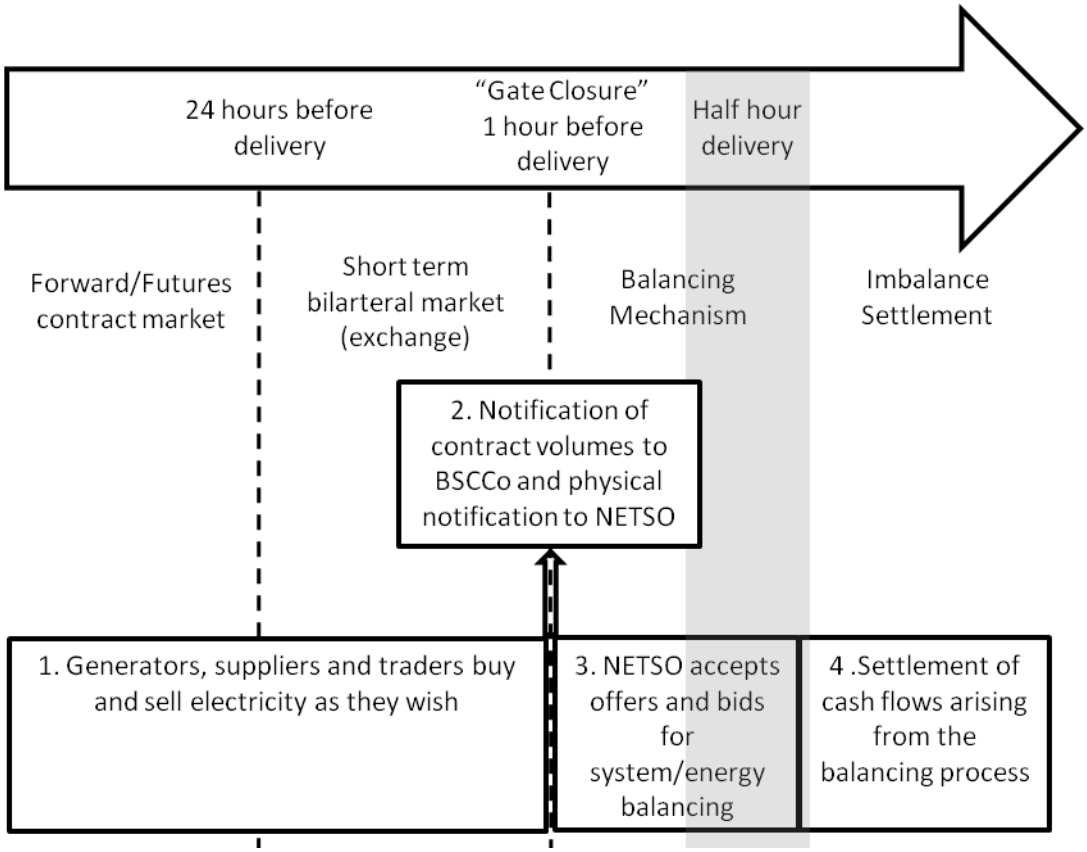


Figure 2.1 - Overview of the BETTA Market Structure

The number of monitoring and control devices throughout the electricity network is increasing. This trend is particularly noticeable at the distribution network and consumer part of the network, as the Low Voltage (LV) system is presently largely devoid of automated data collection. The information flows during operation (as distinct from planning) of the power system between the main actors in the GB electricity network are illustrated in Figure 2.2 (as prepared by the author of this thesis [3]). The anticipated future information flows, following the Smart Meter roll out and increased levels of distributed generation, are illustrated in Figure 2.3 (as prepared by the author of this thesis [3], [19]).

Figure 2.3 shows an increasing level of demand response, enabled by smart meters and the increase in electrified transportation (electric vehicles) and heating (heat pumps). There is some, speculative, talk in the industry of the role of DNOs becoming more akin to the NETSO (National Electricity Transmission System Operator) – with localised stability (frequency control) and balancing (demand and generation influence) being added to the DNO's responsibilities [20]. The term Distribution System Operators (DSO) is used to characterise this change. Whether this should be done and, if so, how it would be done, are open research questions.

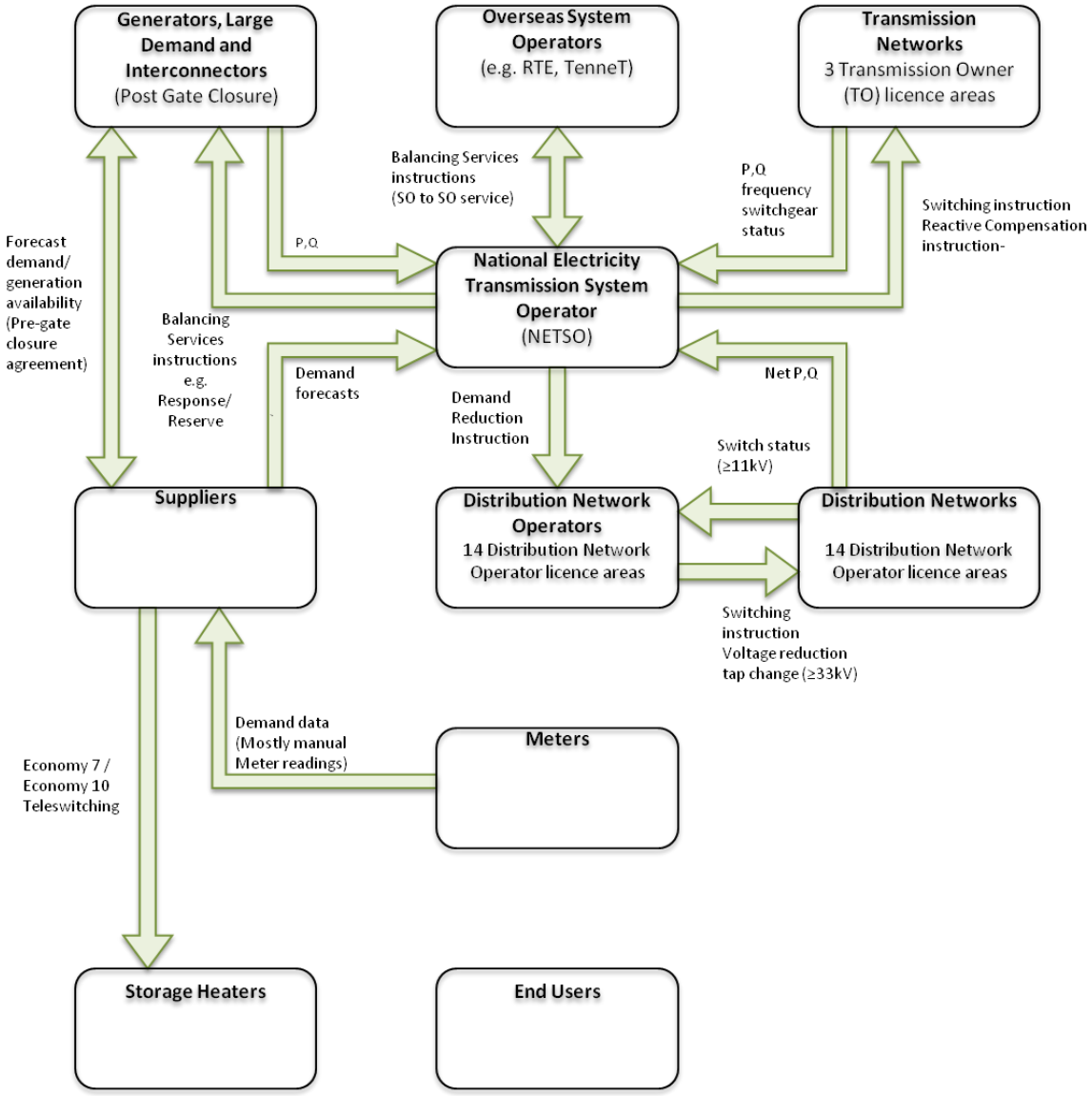


Figure 2.2 - Information Flows for GB Electricity Network Operation

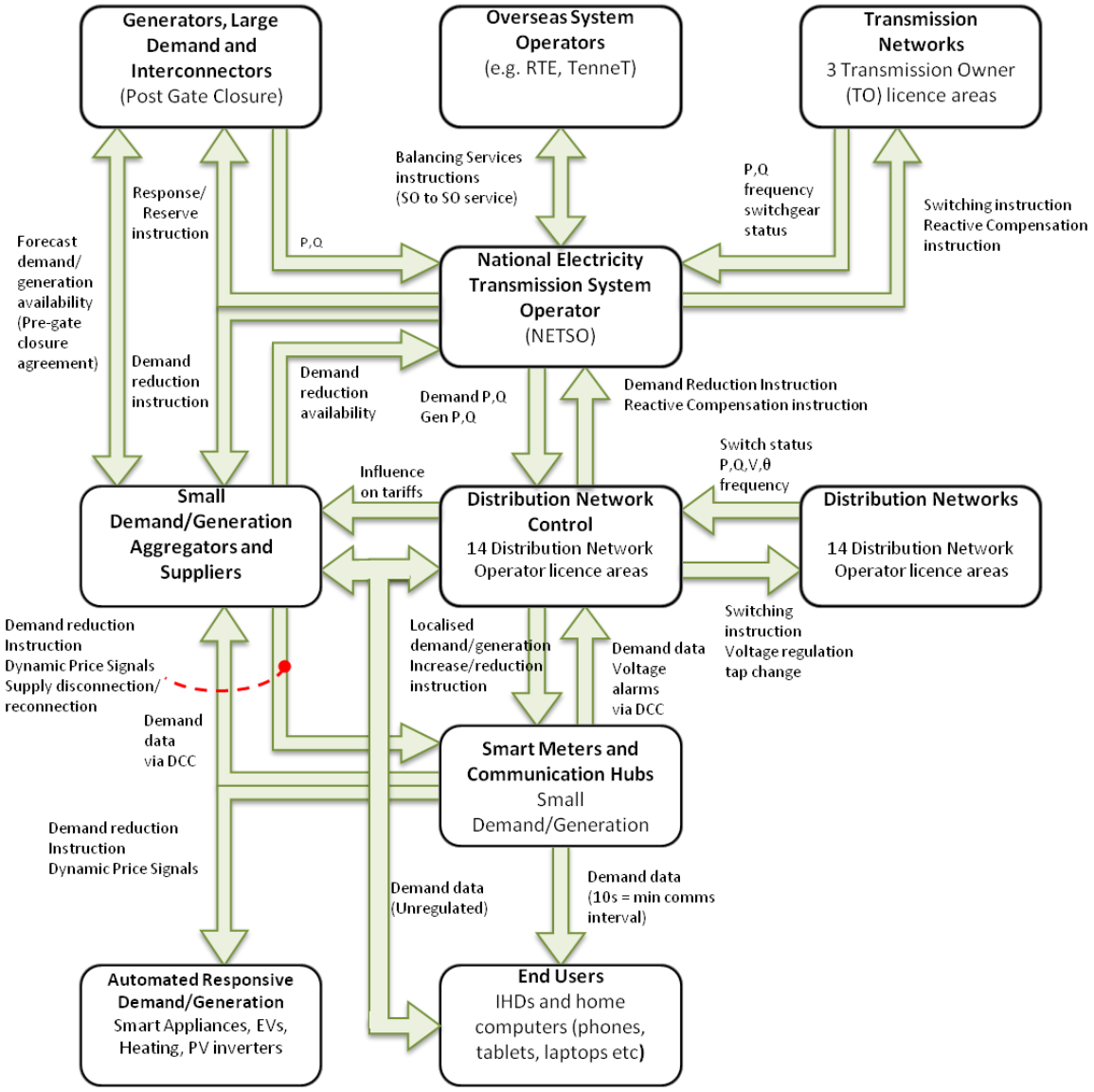


Figure 2.3 – Possible Future Information Flows for GB Electricity Network Operation

2.2 Distribution networks and distributed generation

Distribution networks operate at seven or fewer voltage levels; 132kV, 66kV, 33kV, 11kV, 6.6kV, 3.3kV and 0.4kV (phase to phase). Lakervi and Holmes [21] describe a number of network configurations that are used; Mesh, Interconnected, Link, Open loop and Radial (See Figure 2.4). Mesh networks are frequently used at voltage levels above 33kV. Link, Open Loop and Radial arrangements are more frequently used at 33kV, 11kV and 0.4kV.

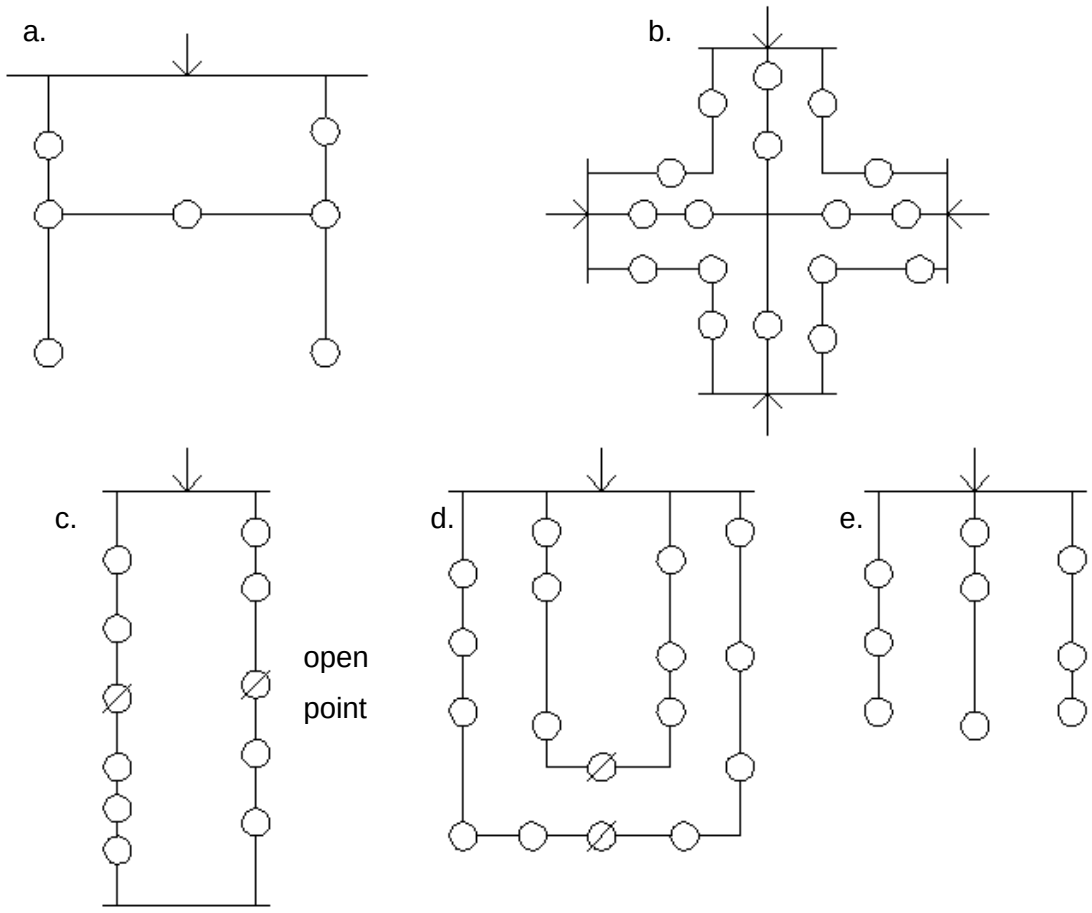


Figure 2.4 - Types of network configuration: a: Mesh, b: Interconnected, c: Link, d: Open Loop, e: Radial. From [21]

Urban distribution networks are characterised by underground cabling. Most consumers are connected to a single phase at the 0.4kV voltage level with many customers (hundreds) sharing a single ground mounted transformer. Larger commercial and industrial consumers have three phase connections and may be connected at a voltage level above 0.4kV. Open Loop networks are common in urban areas. In contrast, rural networks are typified by long radial/linked radial 11kV lines with a dedicated pole-mounted transformer for each customer (or small group of customers).

DNOs must ensure that the supply voltage level to 400V (230V single phase) connections does not vary above 440V (253V single phase) or below 376V (216.5V single phase) at the point of connection – this is +10%/-6% of the nominal voltage (400V line-line, 230V single phase) [22]. The voltage (at the 0.4kV level) varies with the variation of load. Therefore the voltage must be controlled or the system must be designed such that voltage excursions do not occur (e.g. line impedances low enough to avoid excessive voltage drop). Available apparatus and methods for voltage control in 11kV and 0.4kV networks include [21];

- On Load Tap Changers (OLTC) on HV:MV (e.g. 132:11kV, 66:11kV and 33:11kV) transformers
- Off Load Tap changers on MV:LV (e.g. 11:0.433kV) transformers
- Voltage regulators
- Shunt capacitors
- Control of demand/generation.
- Reconfiguration of the network (this acts to change the demand/generation on a given branch)

Presently there is limited use of automation to aid normal operation in distribution networks. Typical practice is to use automatic on-load tap changers on 33:11kV transformers and manual off-load tap changers on 11:0.4kV transformers [21]. Also, remotely controllable Ring-Main Unit switches are used at 11kV (and above). These are used to balance load across 11kV feeders and to reconfigure the network in the event of a fault or for maintenance [23].

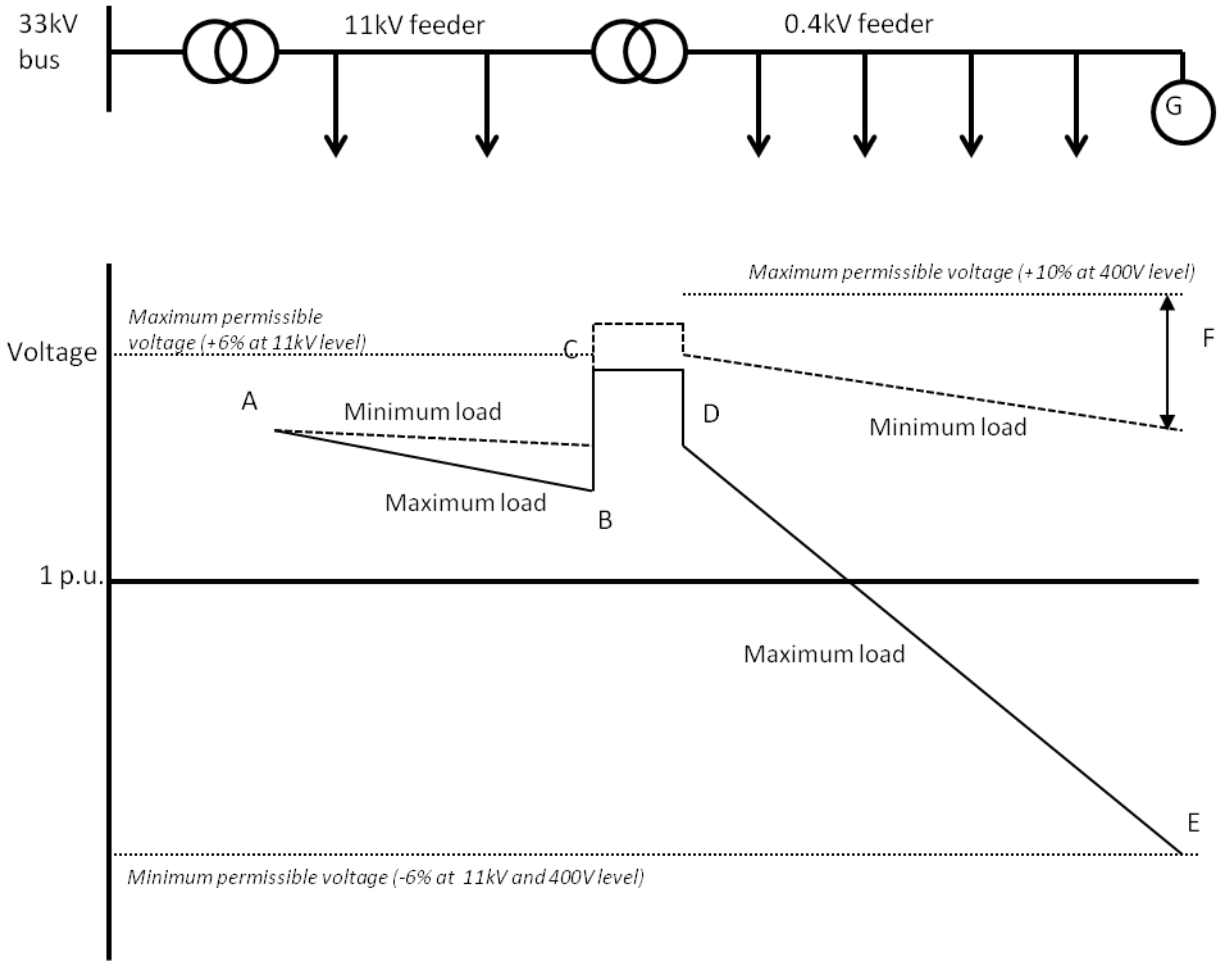
The amount of distributed generation that can be connected at the 400V level is limited by factors including [24]:

- Voltage constraints; a distributed generator (DG) may cause voltage rise which must be limited so that no customers experience a voltage outside limits.
- Fault level; the connection of DG may increase the expected fault level beyond the rating of existing equipment.
- Harmonics; typically caused by power electronic interfaces. These may cause resonances with the local transformer or other plant.
- Transient overvoltages or undervoltages;
- Thermal constraints. E.g. the current carrying capacity of the supply cabling will limit the DG capacity if it is not constrained by the above.

Strbac et al [25] found that network reinforcement costs required for increasing levels of electric vehicles and heat pumps are dominated by the low voltage (0.4kV) networks. They suggested that reinforcements in urban areas are driven by thermal overloads whereas, in semi-urban/rural and rural networks, it is mostly due to excessive voltage drops. Silversides et al [26] interpreted Strbac et al's investigation on this topic, saying that three quarters of heavily loaded medium density urban feeders are likely to be voltage-drop limited (as opposed to thermally limited).

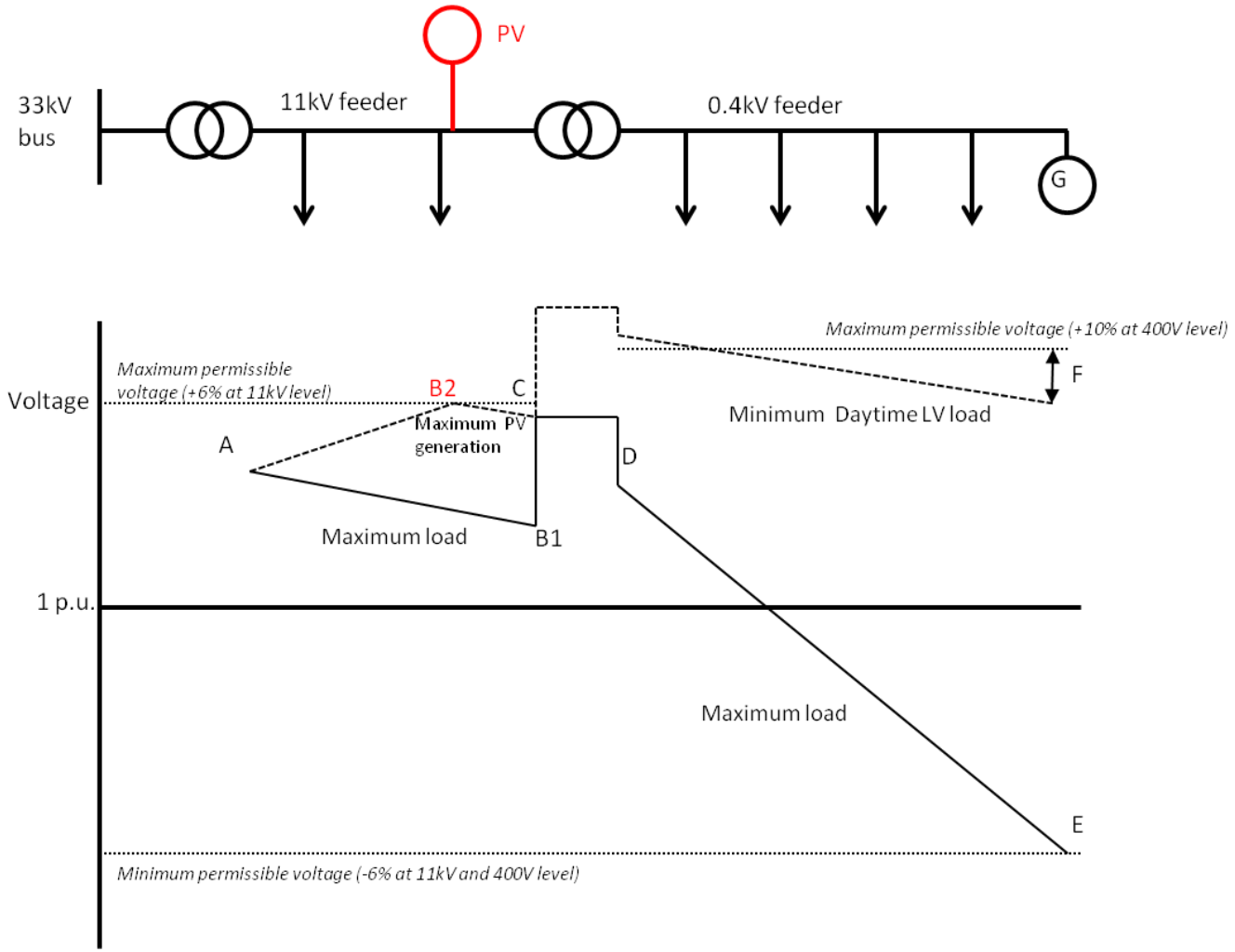
With the advent of distributed generation, traditional assumptions about voltage control at the 0.4kV level will change. A typical voltage profile along an urban 0.4kV distribution feeder cable is shown in Figure 2.5 [21], [27]. The voltage starts high and drops along the feeder until the furthest load is reached. The voltage at the 11:0.4kV substation is set (using manual tap changers - typically set only once, during substation commissioning.) so that the load nearest the substation does not experience an overvoltage and the load furthest from the substation does not experience an undervoltage (for maximum and minimum load conditions). However, connected generators will reduce voltage drop or even cause a voltage rise along the feeder. Therefore, if generators are connected, the possibility of an overvoltage at the furthest (or intermediate) loads must be considered. It is possible that no single transformer tap position will ensure that voltage excursions do not occur (see Figure 2.5). In Figure 2.5 the allowable output of generator G is limited by the voltage rise it causes at minimum load.

Furthermore, if generators are connected at 11kV level, any voltage rise caused at this level would reduce the voltage headroom at 0.4kV (see Figure 2.6). Proposed counter measures include the widening of voltage limits at 11kV, as is being investigated by at least one DNO (see Figure 2.7) and the adjustment of 33/11kV transformer tap change control schemes, based on time of day, to compensate for photovoltaic (PV) generation (see Figure 2.8). Both measures are presently being investigated in an LCNF project [28] in which Automatic Voltage Controller (AVCs – the relays controlling the operation of transformer taps) target voltages will be adjusted to increase capacity for distributed generation. Any resultant increase in capacity on the 11kV network may, however, reduce capacity at the LV level due to reduced voltage headroom on the LV network. The reduced LV headroom can be inferred from Figures 2.6 and 2.7by imagining that more generation is added at point B2, raising the voltage at F.



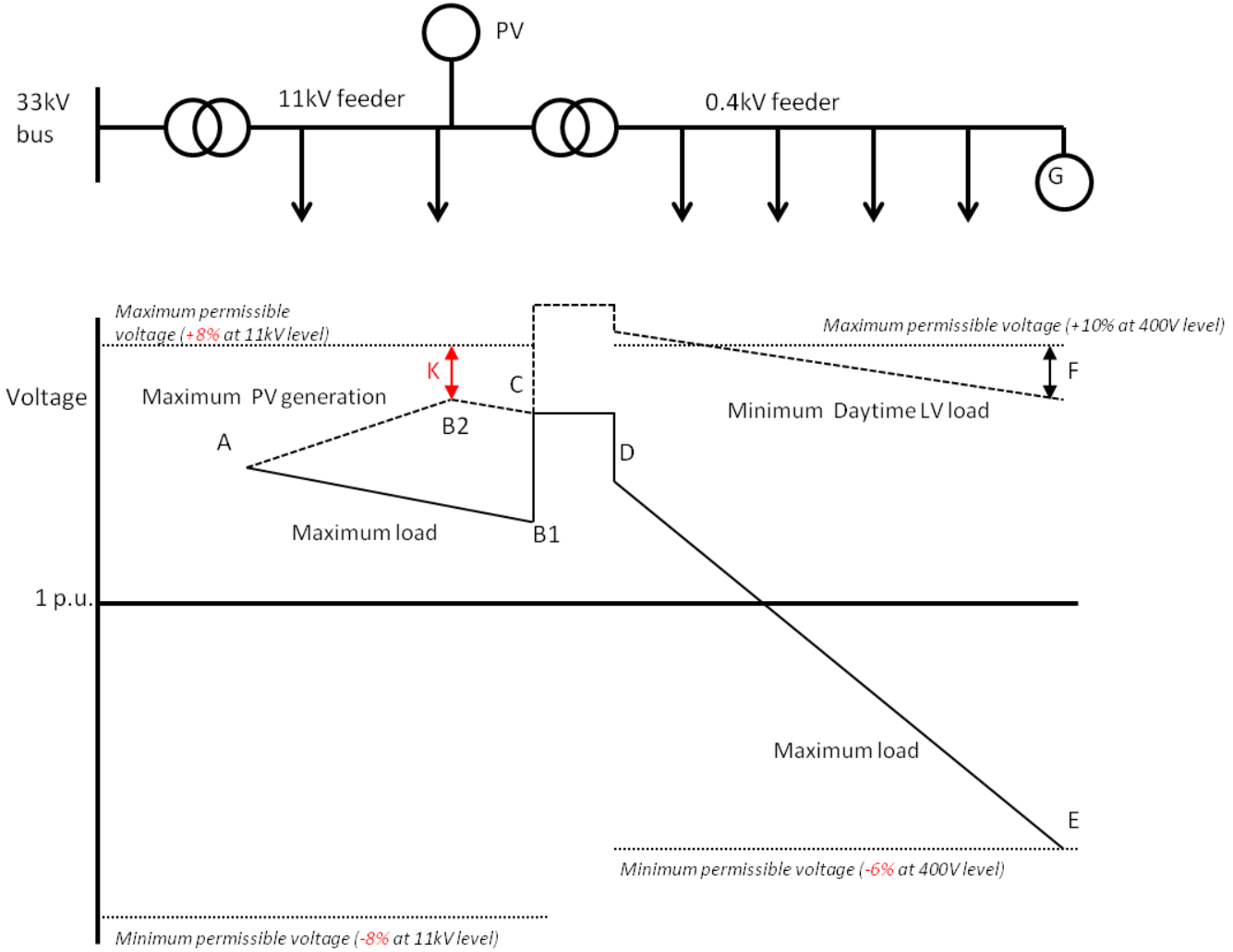
A: voltage held constant by tap-changer of 33:11kV transformer.
A-B: voltage drop to loads at 11kV level.
B-C: voltage boost due to taps of 11kV:0.4kV transformer.
D-E: voltage drop in 0.4kV feeder.
F denotes the allowable voltage rise for a distributed generator.

Figure 2.5 - Voltage variation down a radial feeder. From [27]**.**



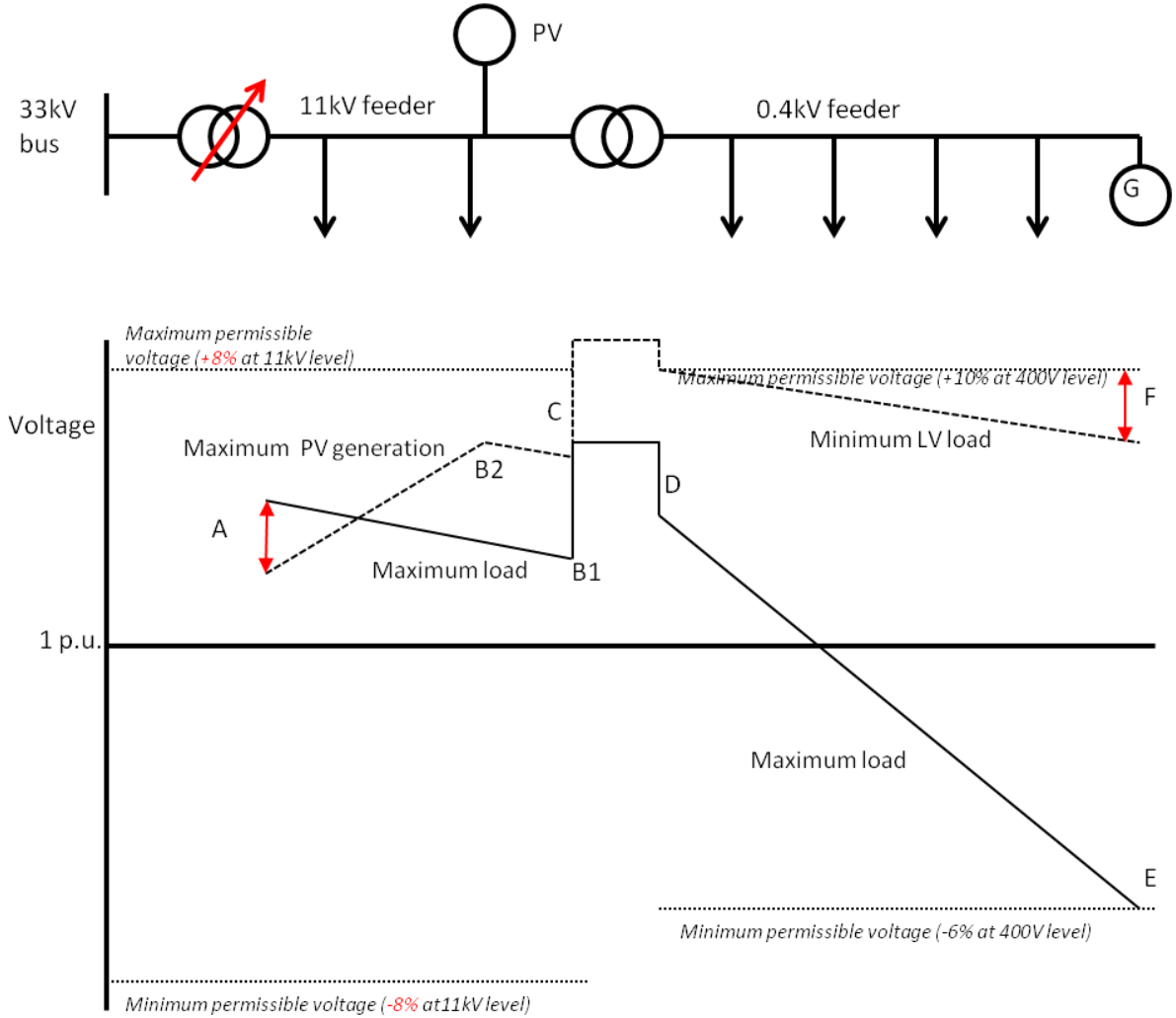
A: voltage held constant by tap-changer of 33:11kV transformer.
A-B1: voltage drop due to loads at 11kV level.
A-B2: voltage rise due to PV at 11kV level.
B-C: voltage boost due to taps of 11kV:0.4kV transformer.
D-E: voltage drop in 0.4kV feeder.
F denotes the allowable voltage rise for a distributed generator.

Figure 2.6 - Voltage variation down a radial feeder, with PV added at 11kV.



A: voltage held constant by tap-changer of 33:11kV transformer.
A-B1: voltage drop due to loads at 11kV level.
A-B2: voltage rise due to PV at 11kV level.
B-C: voltage boost due to taps of 11kV:0.4kV transformer.
D-E: voltage drop in 0.4kV feeder.
F denotes the allowable voltage rise for a distributed generator.
K: Increased headroom for PV if 11kV limits extended

Figure 2.7 - Voltage variation down a radial feeder, with PV added at 11kV and expanded voltage boundaries.



A: voltage changed by tap-changer of 33:11kV transformer to accommodate PV generation in daytime.
A-B1: voltage drop due to loads at 11kV level.
A-B2: voltage rise due to PV at 11kV level.
B-C: voltage boost due to taps of 11kV:0.4kV transformer.
D-E: voltage drop in 0.4kV feeder.
F denotes the allowable voltage rise for a distributed generator.

Figure 2.8 - Voltage variation down a radial feeder, with a tap change control scheme set up to increase PV hosting capacity.

Numerous techniques exist to increase the allowable Distributed Generation (DG) on a distribution network – referred to as the “Hosting Capacity” by Bollen [24] . Different methods apply depending on the factor limiting the hosting capacity for a particular area (e.g. voltage constraints, thermal constraints, fault level etc). The techniques include:

- Demand response [29];
    - Price based [30]
    - Direct control [31]
    - Frequency based [32]
- Use of power electronic converters;
    - Soft Open Points (SOPs) [33]
    - In-line phase rebalancers [26]
    - Fault current limiters [26]
    - PV inverter control [34]
    - Distributed Static Compensators, DSTATCOMs [35]
    - Active Distribution Transformers [36]
    - Mid Feeder Compensation [37]
- In-line voltage regulators [38]
- Generator control [39]–[41]
- Modified control of traditional controllers
    - Transformer tap changers (with AVCs) [42]–[47]
    - Reactive compensation [48]
    - Ring main units (Network reconfiguration[49])
- Change of Fixed tap /off load tap changing distribution transformers to OLTC [50]
- Increased meshing of networks [51]
- Use of energy storage [52], [53]
- Optimisation of fixed, passive settings (E.g. Transformer taps and generator power factor [54]).

Whilst DNOs do not usually interact with demand, they may need to in abnormal circumstances (e.g. insufficient generation, breakdown/operating problems). The following methods of demand control are set out in the Distribution Code [12]:

- Customer Demand reduction, including Voltage Reduction, initiated by the DNO.
- Customer Demand reduction instructed by National Grid

- Automatic low frequency Demand disconnection.
- Emergency manual Demand disconnection.

The DNOs provide automatic low frequency disconnection of at least 60% of the DNO's peak demand (40% in Scotland). This disconnection is performed in stages by tripping relays.

Distribution networks have traditionally been a passive means to pass power from the transmission network to customers [55]. Active Distribution Networks (ADNs) are being researched to increase the capacity, for both generation and demand, of existing distribution networks. ADNs are distribution networks that have systems in place to control a combination of distributed energy resources (generators, loads and storage) [56].

As more monitoring and control devices become available, the risk of unforeseen complex control interactions rises [57]. Therefore, systems for collating information, co-ordinating devices and producing necessary outcomes become important, as well as the necessary modelling platforms. Two extremes of control philosophy are centralised and de-centralised. At the decentralised extreme, each controller decides what action to take based on the information available to it. At the centralised extreme, a single central controller dictates the action of all local controllers.

Between the centralised and decentralised extremes, the controllers have some degree of autonomy and some degree of central co-ordination; there are numerous fields of study related to this type of system. Some of these are described in Table 2.1 [19]. Each of the concepts shown include elements, commonly referred to as agents, which are able to decide whether or not to perform actions based on available information. The application of multi-agent systems to power systems has been explored [58]–[61].

Field of study	Description
Autonomic Systems	“Computing systems that can manage themselves given high-level objectives from administrators”[62]
Autonomic Power System	“... provides flexible and adaptable control through fully distributed intelligence and control”[63]
Complex systems	“...systems with many parts that interact to produce global behaviour that cannot easily be explained in terms of interactions between the individual constituent elements.”[64]
Distributed Artificial Intelligence	“The subfield of AI concerned with co-ordinated, concurrent action and problem solving”[65] “... concerned with the study and construction of semi-autonomous automated systems that interact with each other and their environments.”[66]
Distributed System	“A software system whose constituent parts run on different computers connected by a network”[67]
Game Theory	“... studies interactions among self interested agents”[68]
Multi-agent Systems	“Multi-agent systems are those systems that include multiple autonomous entities with either diverging information or diverging interests, or both” [69]
Semantic Computing	“Semantic Computing addresses technologies which facilitate activities that allow users to create, manipulate or retrieve computational content based on semantics (“meaning”, “intention”), where “content” may be anything such as video, audio, text, software, hardware, network, environment, process, etc.”[70]
Smart Grid	A next-generation electrical power system that is typified by the increased use of communications and information technology in the generation, delivery and consumption of electrical energy [71].

Table 2.1 - Fields of study applicable to control philosophies for power networks.

2.3 The proposed GB smart metering system

The UK government aims to bring about the creation of an electricity and gas smart metering system for all domestic and small non-domestic properties by the end of 2020. The final system will allow Suppliers, Network Operators and other parties (e.g. demand response aggregators) access to smart metering data and demand response functions. The access will be controlled depending on the interests of the company accessing the data (i.e. Supplier or Network Operator) and subject to consumer consent.

New licensed bodies have been given responsibility for the operation of the smart metering system. The Data Communications Company (DCC) will take responsibility for the data and communication aspects of the smart metering system. It will do this with subcontracted Data Service Suppliers (DSPs) and Communication Service Providers (CSPs). The manufacture (to UK government specification) and installation of the smart meters is the responsibility of the Suppliers. A self governed multiparty contract, the Smart Energy Code (SEC), will define the DCC's provision of data communications services [72].

The structure of the smart metering system is shown in Figures 2.9 [73] and 2.10 [74]. These show that each household will have a dedicated Comms Hub to which the electricity and gas smart meters will connect. As well as a Supplier provided In Home Display (IHD), the Comms Hub will also allow connection of the consumer's devices (known as Consumer Access Devices – CADs), subject to a DCC controlled registration procedure. The Comms Hubs will interface between the communication network within the home, referred to as the Home Area Network (HAN), and the communication link with the DCC via what is referred to as the Wide Area Network (WAN). The HAN is likely to operate over low power radio, with a protocol based on the IEEE 802.15.4 standard. The technology for the WAN will be determined by the DSP and will vary based on location. It is expected, however, that cellular network based technology will be predominantly used (e.g. GSM - Global System for Mobile).

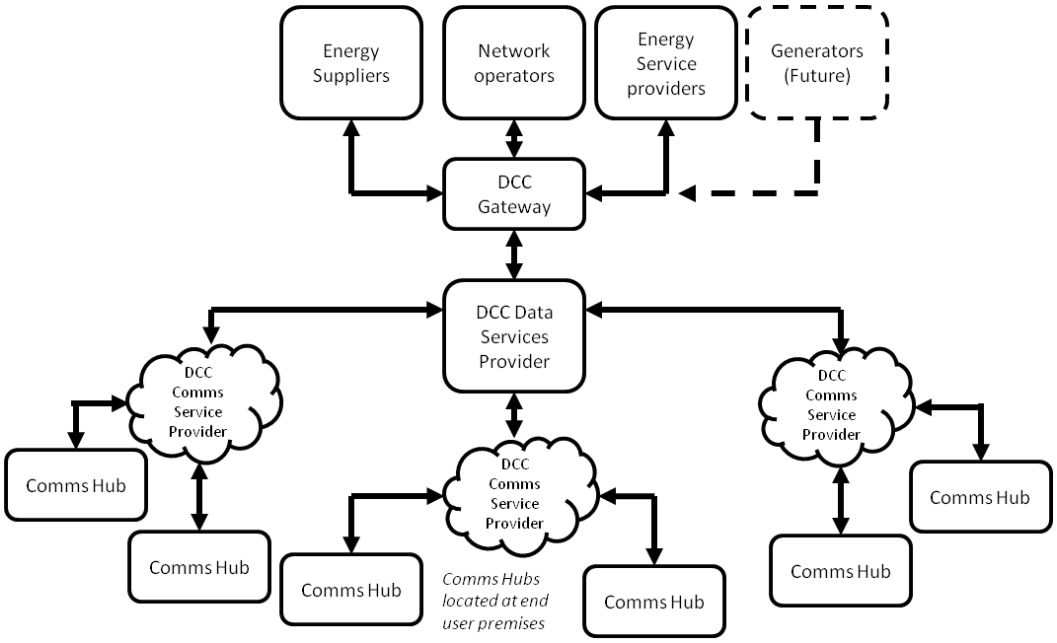


Figure 2.9 – Overview of the proposed GB smart metering system

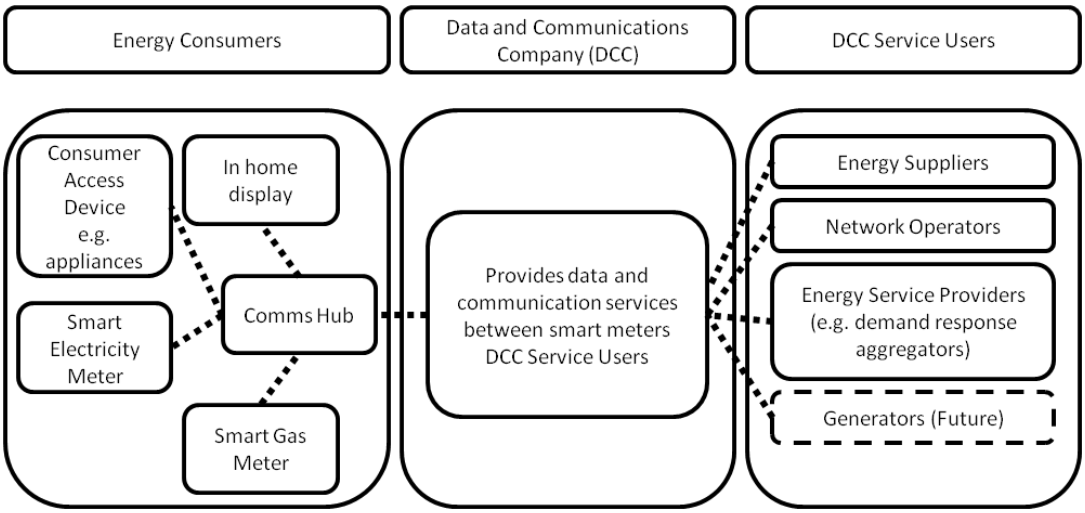


Figure 2.10 - Overview of the proposed GB smart metering system with in home devices

Communication between DNOs and smart meters can be categorised as follows:

Alert – smart meters sending message to DNO based on something it has measured (e.g. over voltage alert).

Configure – DNOs setting configuration parameters that influence the smart meter's operation (e.g. DNO setting overvoltage threshold at which it will receive an alert).

Read – performing a read of the smart meter's memory (or DCC record) (e.g. DNO requesting historical load data for households on a particular feeder).

Demand Response – messages to the Comms Hub intended to influence demand (e.g. requesting that a customer's appliance is switched off). Note, under present proposals, the planned control functions will not be directly accessible to DNOs.

It is proposed that the available alert functions, relevant to DNOs, will include the following [75]:

- Average RMS Voltage above Over Voltage Threshold
- Average RMS Voltage below Under Voltage Threshold
- RMS above Extreme Over Voltage Threshold
- RMS below Extreme Under Voltage Threshold
- RMS above Voltage Swell Threshold
- RMS below Voltage Sag Threshold
- Power Loss
- Supply Outage Restored
- Supply Outage Restored - Outage >= 3 minutes
- Outcome of Ad hoc Change to HAN connected Auxiliary Load Control switch
- Outcome of Calendar Based Change to HAN connected Auxiliary Load Control switch

The relevant proposed read options are shown in Table 2.2 [75]. These are classified as On Demand (DNO gets the information within 30 seconds) and/or Future Dated (DNO gets the information within 24 hours). The available information, relevant to network operators, relates to voltage monitoring (included as "Network" data in Table 2.2 [75]) and power and energy. A capability to record current or frequency is not included in the specification [76].

Service Name	Description	Future Dated (<24 hrs)	On Demand (<30 secs)
Read Instantaneous Import Registers	Read the specified import register or matrix on a specified meter as soon as the Command is received by the meter.	-	✓
Read Instantaneous Export Registers	Read the specified registers on a specified meter as soon as the Command is received by the meter.	-	✓
Read Profile Data	Return the specified date range of profile data from the profile data log for a specified meter.	✓	✓
Read "Network" Data	Retrieve stored power quality data from a Device for a specified Device ID.	✓	✓
Read Maximum Demand Registers	Retrieve the maximum demand register values recorded on a specified meter.	✓	-
Read Load Limit Counter	Retrieve the specified Load Limit Counter data on the specified meter.	✓	-
Read Active Power Import	Retrieve the specified Active Power Import values on the specified meter.	✓	✓
Retrieve Daily Consumption Log	Retrieve the specified Daily Consumption Log entry(s) on the specified meter.	✓	✓
Read Device Configuration	Retrieve the configuration data values for a specified meter.	-	✓

Table 2.2 – Selected proposed DCC Service Requests, applicable to electricity network operators.

The DNO will have some configuration access when it comes to voltage monitoring. Over and under RMS voltage magnitude alert thresholds will be able to be set along with the timeframe over which the threshold is applied. The smart meter capabilities relating to voltage monitoring include [76]:

- Ability to calculate the average value of RMS voltage over a configurable period and record the value calculated (including the UTC date and time at the end of the period to which the value relates) in the "*Average RMS Voltage Profile Data Log"*
- Ability to compare measured voltage to 6 configurable thresholds (3 high, 3 low); RMS over/under voltage detection, 'Extreme' over/under voltage detection and voltage sag/swell detection. Able to be done across 4 configurable timeframes; RMS period, 'Extreme' period, sag and swell periods).
- Ability to record events and send alerts when the voltage rises above high thresholds or falls below low thresholds for the related timeframe.
- Ability to record supply interruptions. Sends supply restoration notification if interruption is over 3 minutes.

The smart meter capabilities relating to power and energy monitoring include [76]:

- Ability to record energy import/export (kWh) on each of the 731 previous days.
- Ability to record half hourly data (kWh) for:
    - 13 months of Consumption;
    - 3 months of Active Energy Exported;
    - 3 months of Reactive Energy Imported; and
    - 3 months of Reactive Energy Exported.
- Ability to record maximum energy use measured over a half hour period (since last reset).
- Ability to compare active power to configurable thresholds ('Low-Medium', 'Medium-High' and 'Load-limit').
- Ability to record status of energy use as 'Low', 'Medium' or 'High' respectively.

The smart meter capabilities relating to demand response will not, at least initially, be available to DNOs. They include [76]:

- Time of use pricing; ability to store 48 half-hourly prices (beginning at 00 or 30 minutes past the hour).
- Ability to calculate an 'instantaneous cost' based on active power and tariff.
- Ability to read status of and send commands to up to 5 HAN connected auxiliary load control switches.
- Ability to store set of up to 200 'time-of-use switching' rules (in a 'calendar') for load switching (with a randomised offset); for changes in state across half-hours, days and dates.
- Ability to make ad-hoc requests following receipt of a command, that one or more HAN connected auxiliary load control switches (not all smart meters will include auxiliary load control switch(es)) change state.
- Ability to, on receipt of a command, disable or enable the supply.
- Capable of supply disablement if power rises above 'Load-limit' threshold.

2.4 Smart metering use cases for network operators

The Department of Energy and Climate Change (DECC) identifies smart metering as “a key enabler of the future Smart Grid” [77], the aim being to achieve the following:

- Decarbonising the electricity system through increasing connection of renewable generation and decentralised generation
- Improving asset utilisation
- Providing decision making support to engineers (in both operation and planning)
- Maintaining system stability and security of supply

Some of the anticipated uses of the smart metering system, for the network operators, are shown in Tables 2.3 [78] and 2.4 [79].

Use Case	Description	DNO Benefit(s)
Power Flow monitoring	Use of demand P and Q data from smart meters with state estimation algorithms, or smart meter voltage alarms, to identify the thermal capacity and voltage headroom across the distribution network.	Asset Utilisation
Network load forecasting	Predicting the impact of external factors (e.g. temperature, entertainment schedules, traffic) on the network by analysis of patterns in smart meter data.	Asset Utilisation
Condition Monitoring	Monitoring of the stress that equipment has been placed under – based on voltage and current derived from smart meter data. Comparison with similar assets elsewhere. Allows investment to be postponed without significantly compromising supply reliability.	Asset Utilisation
Phase identification / re-configuration	Use of smart meters to identify which phase loads are connected to (e.g. through examination of outage data or voltages). Analysis of load profiles to identify the optimal configuration of loads across the phases.	Asset Utilisation Generation/ Load Capacity
Impact assessment for new demand and generation	Using recorded smart meter data to simulate the impact of proposed increases in generation and demand in specific locations on the distribution network. Quantification of the network capacity across all network locations and timeframes.	Generation/ Load Capacity
Power quality monitoring	Use of smart meters to identify location and cause of power quality problems.	Power Quality
Maintenance outage planning	Minimisation of the impact of planned outages by review of smart meter profile data.	Reduced Outage Time
Reconnection Scheduling	Use of smart meter load switches to schedule reconnection in the event of a blackout.	Reduced Outage Time Contingency Planning
Latent demand calculation	Scheduling of demand ‘hidden’ by generation. The generation is more likely to trip due to anti-islanding measures. Smart meters can make latent demand known so that plans can be made to support it.	Contingency Planning
Outage reporting	Using smart meter data to demonstrate the performance of the network operator in resolution of an outage.	Performance Reporting

Table 2.3 Electricity network operator uses of smart metering data for planning.

Use Case	Description	DNO Benefit(s)
Outage Management	Use of smart meters to detect loss (and reinstatement) of supply.	Reduced Outage Time
Voltage Monitoring and Control	Use of smart meter measurements to influence the action of voltage control equipment (e.g. On load tap changers, reactive power compensation).	Asset Utilisation Generation/Load Capacity
Demand Response (DR)	Use of smart meters to bring about demand response. Methods include: • Price based (smart meter reacts to received price signals or tariffs), • Direct control (Direct remote operation of smart meter load switch) • Automatic (smart meter initiates demand response based on monitored voltage, current or frequency).	Asset Utilisation Generation/Load Capacity
DR for Energy Use Reduction	To find the optimum voltage for minimum total energy use.	
DR for Ancillary Services	To allow network operators to obtain ancillary services (such as frequency response), typically obtained from the supply side, from the demand side.	
DR for Local constraints	To manage local network voltage and thermal constraints.	
DR for Generation mix	To allow demand to follow availability of intermittent energy sources.	
Demand Response Verification	Use of smart meters to verify that demand response has taken place. This may be required if the customer is to be paid for automatic response (see above) or if the customer has entered in to a contract with an aggregator.	Asset Utilisation Generation/Load Capacity
Network Reconfiguration	Use of smart meter measurements to supplement network reconfiguration algorithms.	Asset Utilisation Generation/Load Capacity Reduced Outage Time Contingency Planning

Table 2.4 - Electricity network operator uses of smart metering data for operation.

2.5 The problem of phase imbalance in distribution networks

Voltage excursions are likely to increase as the phase imbalance of loads/generators increases. Phase imbalance occurs due to differing single phase load and generation magnitudes along the distribution feeder. As a result of imbalance, neutral currents flow. This results in a rise in neutral voltage - meaning that the voltage seen by the load (i.e. the voltage between the phase conductor and neutral) will change. Therefore, as more generation gets connected and as large loads such as heat pumps or electric vehicles become popular, the amount of imbalance is likely to (depending on connection and usage patterns) increase.

Phase imbalance is likely to be a limiting factor in the connection of distributed generation. A Western Power Distribution Innovation Funding Report [80] indicates that increased single phase distributed generation is likely to cause voltage unbalance problems. It also points out that this could limit the connection of three phase generation. Furthermore, The problem of phase imbalance has been identified in LCNF projects [26].

To demonstrate the effect of imbalance on load voltages, consider the circuit shown in Figure 2.11. To simplify, resistances are used (as opposed to impedances). The figure shows a three phase source emitting a current of 1 per unit per phase. Three single phase loads are also shown, each with a notional resistance of 1 per unit. The voltage across the load is therefore:

$V_{load} = R_{load} \times I_1 = 1 \times 1 = 1$ p.u.

The total power used, per phase, by the load is therefore:

$P_{phase,1} = V_{load} \times I_1 = 1 \times 1 = 1$ p.u.

This represents a balanced case. As it is balanced, no neutral current flows. Therefore the total line voltage drop for each phase is given by:

$V_{drop} = I_1 \times R_{line} = 1 \times 0.1 = 0.1$ p.u.

An unbalanced case was created where the total load power from the balanced case is combined on a single phase only (see Figure 2.12). The power used by that phase becomes:

$P_{phase,2} = 3 \times P_{phase,1} = 3 \times 1 = 3$ p.u.

Therefore, if the same voltage across the load is assumed, the current becomes:

$I_2 = P_{phase,1}/V_{load} = 3/1 = 3$ p.u.

In this extreme unbalanced case, the full phase current flows in the neutral. Hence the voltage drop is:

$V_{drop} = I_2 \times R_{line} + I_2 \times R_{neutral} = 3 \times 0.1 + 3 \times 0.1 = 0.6$ p.u.

This simplified case shows the total line voltage drop increasing by a factor of 6 in the case of extreme unbalance. On real networks, this effect may result in an increased likelihood of voltage excursions where loads/generators are unbalanced.

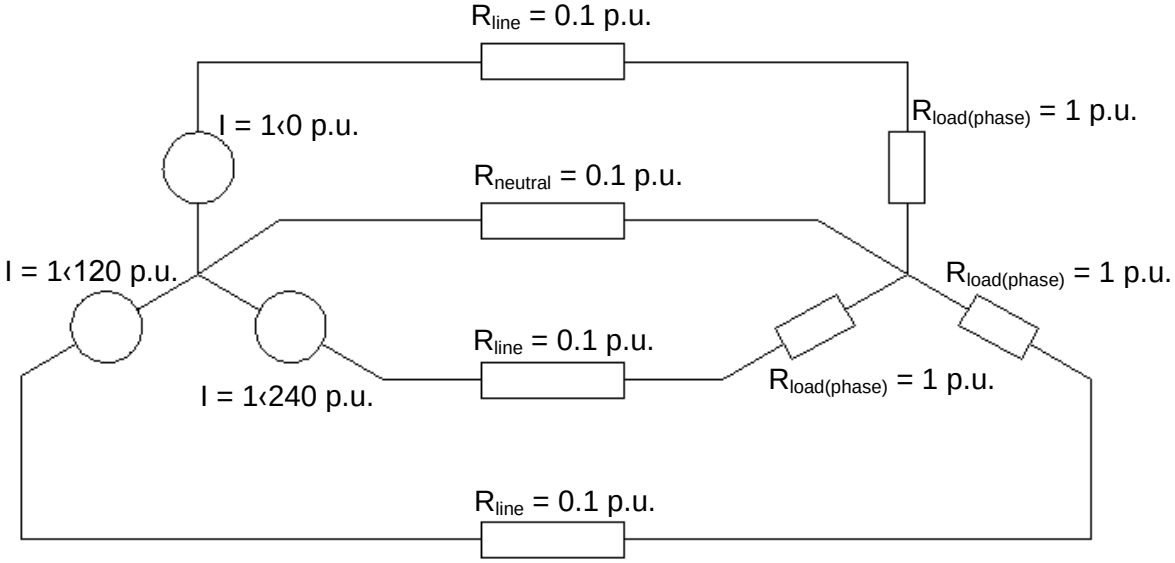


Figure 2.11 - Simple balanced network for demonstration of voltage unbalance

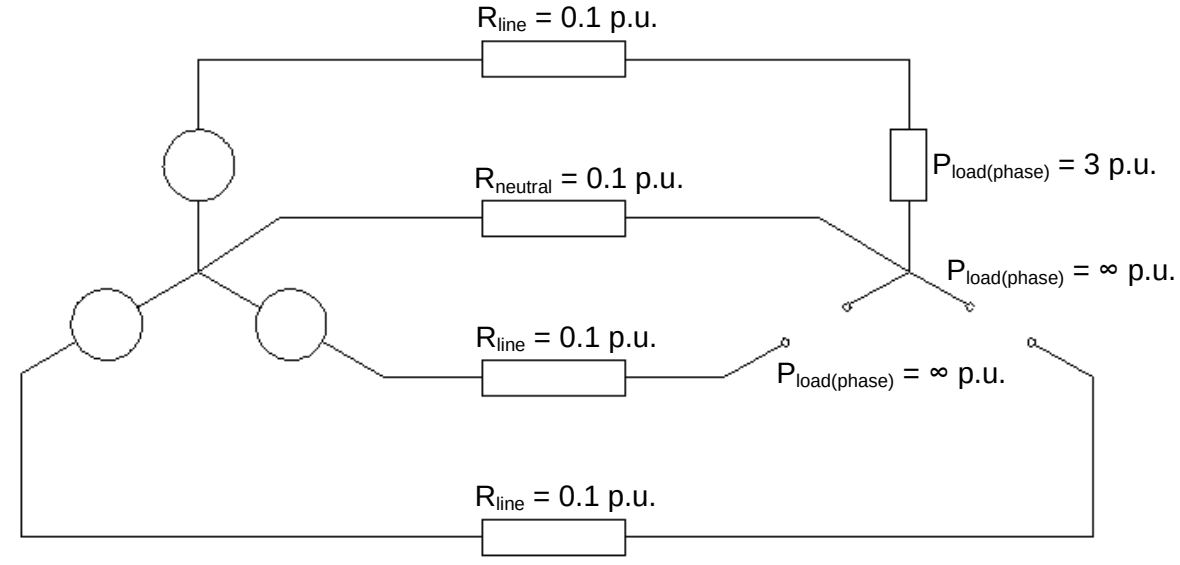


Figure 2.12 - Simple unbalanced network for demonstration of unbalance

In the case of unbalanced star connected loads, the neutral voltage rise that occurs during unbalanced loading is related to the impedance of the neutral return path.

This is related to the cross sectional area of the neutral conductor, the proximity to phase conductors, the earthing arrangement and the impedance of earth (if the return path goes through earth). Balanced loads and parallel return paths reduce the need for a ‘full-size’ neutral conductor (of the same cross sectional area as the phase conductor). The cross sectional area of the neutral conductor is sometimes under-sized in relation to the phase conductor.

Low voltage (400V) earthing arrangements in the UK are usually defined as TN-C-S, TN-S, TT or IT. Where T stands for Terra (Latin for earth), N for Neutral, C for Combined, S for Separate and I for Isolated. When read from left to right these acronyms describe the earthing and neutral arrangement from the distribution transformer (or generator) to the load within a premises (see Figure 2.13 [81][82]). For instance, TN-C-S (Terra-Neutral-Combined-Separate) implies a transformer with an earthed star point and a conductor that acts as both neutral and earth until it reaches the premises at which point separate neutral and earth conductors are used. Similarly, TN-S implies that separate neutral and earth conductors are used all the way from the transformer star point to the load. The TT system, Terra-Terra, describes a system with earths at both the transformer and load and with no earth or neutral conductors. Finally, IT (Isolated Terra) represents a system with no earth connection at the transformer but with an earth at the load.

There are two common variants of TN-C-S systems; Protective Multiple Earth (PME) and Protective Neutral Bonding (PNB). In PME, one or more additional earth points are located along (or at the end of) the combined neutral and earth conductor. When a system is described as TN-C-S it is implied that it is also a PME system[82]. It is the policy of many Distribution Network Operators to use PME where reasonably practical [81], [83], [84].

Another variant of TN-C-S is Protective Neutral Bonding (PNB). In PNB, the neutral is only earthed at one point. The earth is usually located nearer to the load than to the transformer. PNB is only used for single, or small groups of, loads (e.g. a pole mounted transformer supply) [81].

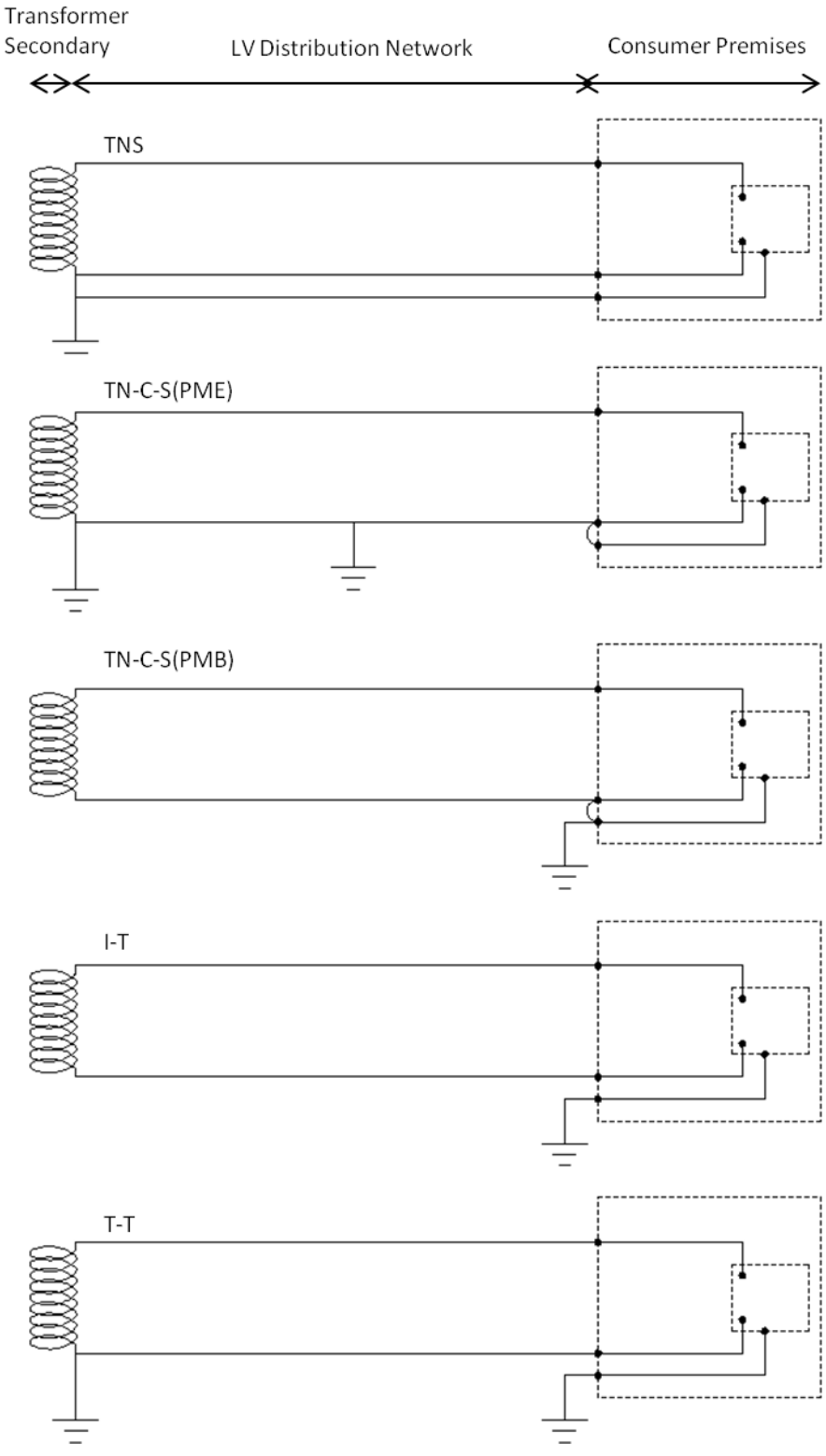


Figure 2.13 - Earthing and Neutral arrangements commonly used in LV networks.

2.6 Representing imbalance

Symmetrical component theory is widely used to represent unbalanced sets of phasors. In essence, symmetrical component theory says that any set of three phasors (e.g. representing three voltages) can be represented by two sets of three symmetrical phasors and one set of three identical phasors. These sets are referred to as positive sequence, negative sequence and zero sequence components. To illustrate, a set of unbalanced phasors (Figure 2.14) are split into constituent symmetrical components in Figure 2.15.

Lyon [85] describes an observation underlying symmetrical component theory (quotation abridged), "… a stationary sinusoidally distributed field, the magnitude of which varies harmonically in time, is equivalent to two equal and constant sinusoidal fields that move at the same singular speed, but in opposite directions." In three phase systems, a similar effect is achieved by changing the sequence of rotating vectors, i.e. A-B-C (positive sequence) to A-C-B (negative sequence). It was found that some unbalanced sets of phasors could be resolved into the two symmetrical sets of phasors. Incidentally, according to Lyon, Stokvis [86] had got this far in 1915 before Fortescue [87], in 1918, proposed the zero sequence components, which can be imagined as an offset of the positive and negative components.

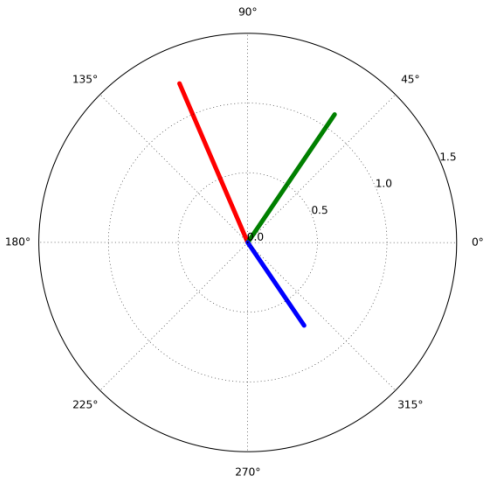
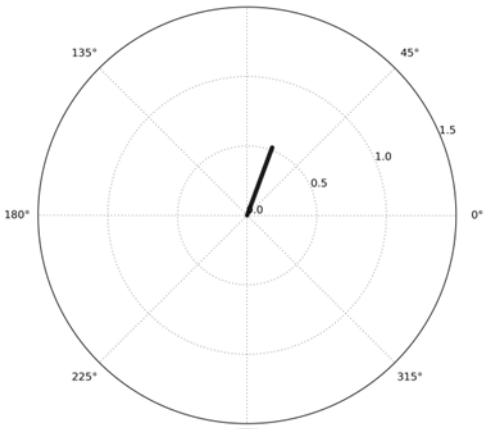
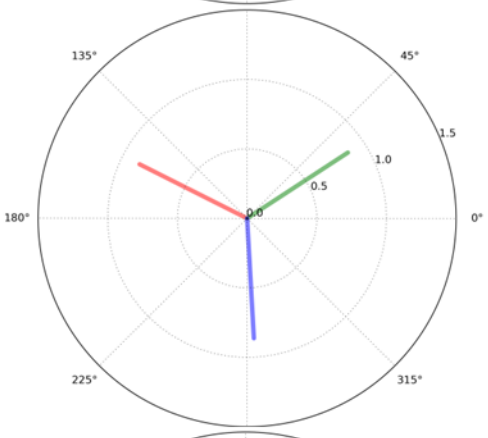


Figure 2.14 - Example three phase unbalanced phasors (could represent voltage or current)

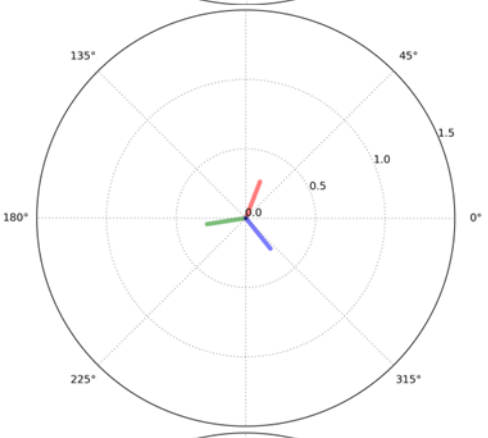
Zero sequence components:



Positive sequence components :



Negative sequence components :



All components and original phasors:

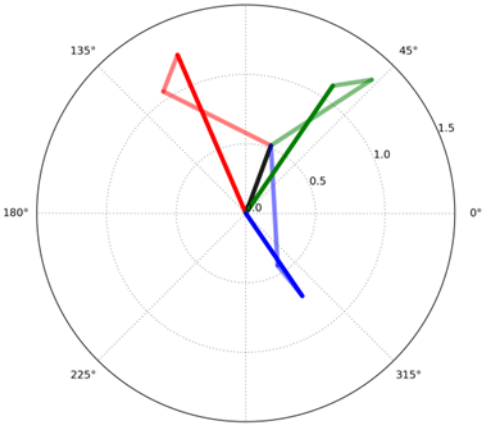


Figure 2.15 - The sequence components of the example phasors

Bollen [88] highlighted the different definitions of voltage unbalance. In particular, he emphasises the two different IEEE definitions; Std.936 ("the difference between the highest and the lowest RMS voltage, referred to the average of the three voltage (sic)") and Std.112 ("the maximum deviation from the average phase voltage, referred to the average of the phase voltage"). Bollen concludes that the two definitions can give significantly different results, especially if zero-sequence voltages are present. The "true" definition of voltage unbalance is defined as the ratio of the (magnitude of the) negative sequence voltage component to the (magnitude of the) positive sequence component [88], [89]. This is commonly referred to as VUF (Voltage Unbalance Factor) and is widely used in the literature as a measure of unbalance, often expressed as a percentage:

$$VUF_{\%} = \frac{|V_{neg}|}{|V_{pos}|} \times 100$$

Voltage unbalance tends to increase heating in three-phase induction machines and causes thermal stress in power electronic equipment. Engineering Recommendation P29 recommends that the VUF does not go above 1.3% [90] (after [91]). However, in low voltage networks feeding residential areas it is unlikely that three phase equipment will be connected. In these cases, therefore, the main problem associated with voltage unbalance will be voltage excursions or increased line currents.

2.7 Modelling unbalanced distribution networks

In Zimmerman's comprehensive review of distribution power flow [92], distribution power flows were grouped into three categories (The references listed here are as per Zimmerman's Thesis. Some of them were not available or not found. These are listed as [Not available] in the references) :

- Network reduction methods [93]
- Backward/forward sweep methods [94]–[98]
- Fast decoupled methods [99]–[101]

Most of the techniques identified by Zimmerman are suited to balanced three phase analysis and are therefore not applicable to unbalanced distribution network analysis. One exception to this is the network reduction method used by Berg, Hawkins and Pleines [93].

In his PhD thesis, Zimmerman goes on to present a number of methods for distribution load flow, each based on a basic building block, See Figure 2.16. Zimmerman focuses on three wire systems and did not allow for the explicit definition of neutral impedance or return path. Rather, he defines Y connected loads with a grounded (0V) star point.

A similar building block is used by Broadwater, Chandrasekaran, Huddleston and Khan's [102]. They also do not allow for the explicit definition of neutral impedance, meaning that neutral current cannot directly be calculated. In both cases, it is possible, however, that the building blocks could be created using Carson's equations [103] to find the self and mutual impedances including the effect of ground [104] and Kron reduction [105], as suggested by Kersting [106] and Srinivas [107]. However, as suggested by Ochoa et al [108], the use of reduced matrix resistances (i.e. using Kron reduction) can lead to inaccuracy. One reason for this is that neutral currents are not defined and therefore the neutral I^2R losses do not get accounted for and neutral voltage are not calculated.

Teng's [109]–[112] three phase unbalanced power flow for radial distribution networks does not include explicit definition of neutral impedance. Again the neutral and ground are included using Kron reduction [105], as suggested by Kersting [106] and Srinivas [107], then neutral impedance can be input and neutral current calculated.

Da Costa, de Oliveira and Guedes [113] build on Garcia, Pereira, Caneiro, Da Costa and Martins'[114] Newton-Raphson based Three phase Current Injection Method (TCIM) method. Araujo, Penido, Carneiro, Pereira and Garcia [115] compare TCIM to the forward-backward technique, concluding that, for large systems or highly meshed systems, TCIM required fewer iterations. However, in a later paper, they note that ground and neutral representation is not allowed for with TCIM [116].

Khushalani , Schulz [117] and Solanki [118] extend the work of Cheng and Shirmohammadi [94] to include three phase distributed generators. Khushalani and Schulz highlight the difficulty in modelling PV bus (constant real power, constant voltage magnitude) generators in radial power flow programs. As with previously identified literature, this work does not appear to allow for the explicit definition of neutral impedances. Other published techniques, that do not allow for an explicitly stated neutral [97], [98], [101], [119]–[142], or do otherwise not allow for unbalanced analysis [143], [144] exist.

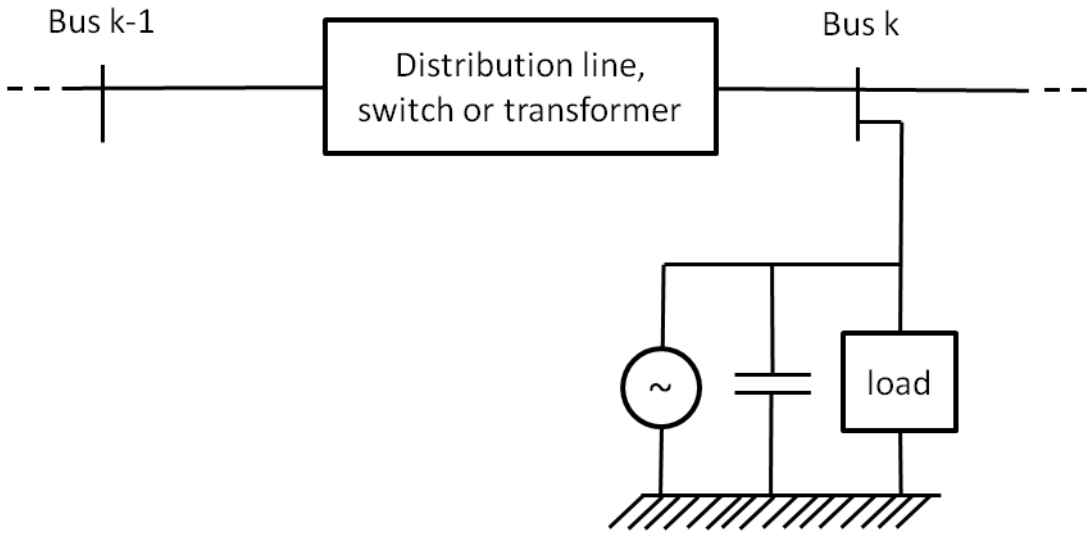


Figure 2.16 - Basic building block used for unbalanced load flow by Zimmerman [92]

Ciric, Feltrin and Ochoa [145] present an unbalanced load flow method in which, unlike the majority of other methods, neutral impedances are explicitly defined. They expand on Cheng and Shirmohammadi's [94] backward-forward method, expanding the 3 x 3 matrix network representation to a 5 x 5 matrix (to represent the three phases, the neutral conductor and an assumed earth path).

Penido, de Arujo, Carneiro Jr, Pereria and Garcia's Newton Raphson based Four conductor Current Injection Method (FCIM) [116] has the advantage, like Ciric, Feltrin and Ochoas' method, that the neutrals and ground connections are explicitly represented. They also assert that their FCIM method is not limited to radial topologies, unlike Ciric and Feltrin's. Penido et al go on to show results from a simulated 220V underground network, including currents flowing in the neutral grounding impedances of the transformers.

Peralta, deLeon and Mahseredjian [146] represent neutral and ground wires as positive-sequence lines (referencing Ciric, Feltrin and Ochoa). They introduce a zero power, zero voltage PV bus at the grounding point, validated their method using EMTP-RV and noted the growing interest in modelling of unbalanced power systems.

In their 2011 review of load flow calculations in distribution systems with distributed resources, Martinez and Mahseredjian [147] highlight the requirement for three phase modelling of distribution systems and group load flow methods into two main categories; deterministic and non-deterministic. Within the deterministic category

they list the following sub-categories; backward/forward sweep methods, Newton-Raphson type, Gauss-Siedel or fixed-point and EMT initialisation. In the non-deterministic category they list; Numerical, Analytical, probabilistic for distribution. Martinez and Mahseredjian say that the backward/forward method is the most popular. They categorise Berg, Hawkins and Pleines' [93] method under the backward/forward category, unlike Zimmerman [92] – who creates a separate "network reduction" category for that method. Furthermore, Martinez and Mahseredjian go on to say that the Berg, Hawkins and Pleines paper is where the backward/forward method was first proposed.

Backward/forward, or Ladder-Iterative [106], techniques work by first initialising the node voltages and then sweeping from the most remote (with respect to the source) node back to the source node, calculating the current at each step and the voltage at the node next closest to the source. After the source node is reached, another sweep is performed, working away from the source node, in which the voltages are calculated using the currents from the previous step. These voltages are then used in a repeat backward sweep. The process continues until convergence stopping criteria are reached.

Newton based and Gauss-Siedel/fixed point techniques involve creating a set of simultaneous equations representing variable relationships at each node – nodal equations. Typically, the variables are selected from voltages, admittances, real power, reactive power, voltage angle and current at each node and/or within each branch. The formulation and solution technique depend on the chosen algorithm. An iterative process is always used.

For instance, in the Newton Raphson technique [148], partial derivatives of the nodal equations are taken and put into matrix form – known as the Jacobian. This is then used to linearly relate the known values (mismatches) to the unknowns (corrections). These linear relations are then solved to find the unknowns. The unknowns are then passed back to the non-linear nodal equations. The process repeats until the unknowns converge (to within some error value) or until it is deemed to have failed (reached a large number of iterations, e.g. ~100).

In performing network power-flow analysis for high penetration distributed micro-generation, Thomson and Infield [149] performed unbalanced power flow analysis using the backward/forward technique developed by Kersting [106]. Thomson and Infield note that there further verification of the modelling technique is required. In

an earlier, 2003, paper they, and others, suggest that unbalanced analysis at low voltage was rarely done [150].

In 1968, MA Laughton [151], [152] made the observation that, with the advent of digital computing, calculations using symmetrical component transformations are not required. However, Laughton also noted that symmetrical components should be retained as measures of system unbalance. Laughton's papers show that it is possible to solve unbalanced network problems using phase co-ordinates. In both of the papers, Laughton assumes that star point neutral points are either solidly earthed or earthed via some impedance, where earth is a 0V plane. Dugan [153] also recommends that symmetrical components are not used in software based network modelling.

Probabilistic load flow takes into account uncertainty of node data and will typically give results as probabilities (e.g. the probability that a branch flow will exceed a capacity limit) [154]. Many of the published papers on probabilistic techniques do not explicitly consider unbalanced networks [155]–[165]. Exceptions to this include Hatziagyrou and Karakatsanis [166], Caramia, Carpinelli, Pagano and Varilone,[167] and Bracale, Carpinelli, Caramia, Russo and Varilone [168]. However none of these examples explicitly consider a neutral, probably because their focus is at a medium voltage level (11kV-33kV) where use of neutral conductors is less common. There is no reason that the probabilistic technique could not be applied to low voltage networks with neutral return paths.

In 2000, Wang and Yang [169] noted that random variation of voltage unbalance is rarely considered in the literature. They used a monte-carlo technique to produce Gaussian distributions of line to line voltages. However, they did not discuss the theoretical background.

In a section titled "Modeling of distribution feeders", Schneider, Chassin, Chen and Fuller [170] say they have constructed 25 radial distribution feeders in their software, GridLAB-D. GridLAB-D uses a multi-agent philosophy to concurrently run modules (e.g. thermal, power flow models). Its power flow module uses either a forward-backward sweep or a current injection method [171].

Shahnia et al [35], [172] describe a formulation for unbalanced load flow on a low voltage feeder where each single phase load is taken into account, and where neutral impedances are explicitly allowed for. The technique also allows for the introduction of fixed power photovoltaic generation in parallel with each load.

From the papers reviewed, 4 techniques appear to be directly (i.e. without modification) suitable for load flow on an unbalanced radial feeder with a separate neutral conductor. These are Berg, Hawkins and Pleines [93], Ciric, Feltrin and Ochoa [145], Penido et al [116] and Shahnia et al [35], [172].

OpenDSS is an open-source software program aimed at modelling distribution networks. It was written in Dephi and developed by the Electric Power Research Institute (EPRI), each element in the system is represented by a nodal admittance matrix. These matrices are combined into a single system admittance matrix and solved using sparse matrix solvers [173]. An iterative process (either fixed point or newton) is used to find the system voltages and the currents of power conversion elements are updated at every iteration.

OpenDSS is able to perform unbalanced power flow with an explicitly defined neutral. Whilst a Linux version is in development, at the time of writing, OpenDSS is only available for the Windows operating system [174]. This is problematic if supercomputing resources, which are typically Unix based, need to be used to run a large number of scripts. However, as OpenDSS is open source, there is little restriction in adapting the existing code.

1.2 Domestic demand modelling review

Samarakoon [31] divides load modelling methods into two main categories; "Load forecasting" and "End-use models". Load forecasting is the estimation of future demand based on previously recorded data (e.g. billing data, weather data). End use models combine appliance power-time profiles and appliance usage probabilities to build total household (or other building) load profiles.

Load forecasting techniques typically aim to predict of the aggregated sum of large numbers of loads. They are therefore ill-suited for use in system modelling where individual appliance and household demand profiles must be considered. Hahn, Meyer-Nieberg and Pickl [175], Alfares and Nazeeruddin [176] and Suganthi and Samuel [177] review and classify various load forecasting techniques. Notably, Alfares and Nazeeruddin observed a trend towards stochastic methods.

Samarakoon [31] found three variants of end-use models; Capasso, Grattieri, Lamedica and Prudenzi [178], Paatero and Lund [179] and Richardson, Thomson, Infield and Clifford [180]. These end use models are based on time use survey data – these are limited in the sense that some socio-economic groups are missed. For

instance, the time use survey on which Richardson et al's model is based does not include retirees.

Richardson et al's model has the following limitations, identified in their paper [180]:

- The model over-represents active occupancy in dwellings with low energy use, i.e. houses that are not occupied for several days are not captured by the time use survey.
- Correlations in appliance ownership across socio economic groups is not taken into account
- No account is taken of consumer attitudes toward energy
- Underestimated seasonal variation – the occupancy model (simulates number of people within a household) is not seasonal (seasonal effects are taken into account in other ways e.g. lighting, heating and usage likelihood)
- A possible over estimate of demand after 1800 hrs as less multi-tasking at this point is not represented.
- Most appliances are treated as constant demand – in reality they may vary.
- Some behaviours are not included. E.g. re-boiling of kettles.
- Small loads (e.g. phone chargers) are not included.

Inspection of the code also revealed:

- The mean cycle time for electric showers is three minutes – this is perhaps a slight underestimate [181].
- The lighting component of the model assumes all lamps are incandescent (ratings from 40W to 100W) – it is likely that as lighting technology will change (to fluorescent and LED) [182] and that these ratings will therefore also change.

Notwithstanding the above, the model was validated, by Richardson et al, as being in agreement with recorded data. After the individual appliance profiles have been combined, the final stage of the model involves a scaling of the demand in accordance with available aggregate data. In this way omitted appliances, socioeconomic groups or behaviours are, indirectly, allowed for. Furthermore, the code behind the model has been made available under a permissive licence – making it convenient for other researchers to make use of.

Capasso et al's [178] method uses Monte-Carlo extraction for creation of domestic load profiles and is based on Italian time use surveys (in which participants kept

diaries recording their activities). The method differs from Richardson et al's in that it takes Socio-economic grouping, gender and age of occupants into account. Notably, Capasso et al also express their desire to incorporate weather dependence, heating, ventilation and demand response into the model. They compare predicted and recorded load data and note that they approximately agreed.

Paatero and Lund's [179] model produces hourly household demand profiles. Seasonal effects are excluded from the model. The model was based on hourly probability factors for appliances or groups of appliances – these factors were taken from outputs of the SAVE and EURECO European projects. Whilst the data is reported in an appendix, the source data (from the SAVE and EURECO projects) could not be found. Also, Paatero and Lund's model did not include any form of heating or air conditioning.

Samarakoon [31] combined the method of Richardson et al [180] and the appliance demand profiles from Paatero and Lund [179] to construct grouped appliance demand profiles for all appliances in Great Britain. The appliance groups were; Fridges and freezers, clothes washers and dryers, and hobs and ovens. Based on this, Samarakoon estimated that 1.6GW of controllable load is available from fridges and freezers (at any time), 1 to 4GW from washers/dryers (between 1000 and 2200 hrs) and 0.5 to 2GW from hobs/ovens (between 1000 and 2200 hrs).

In all of the reviewed end use load models, there is scope to improve the way that temperature dependence (of each appliance) is modelled. Neither Capasso et al's or Paatero and Lund's models use temperature to directly influence the appliance demand profiles (it can be said that it is indirectly allowed for in the source survey data which is recorded separately in different seasons. Richardson et al do use temperature modifier factors, derived from Met office data, to influence the likelihood that electric heating is used. However, using thermal models of relevant appliances (e.g. fridges) would improve the accuracy. Cheng [183] details an example of how this could be achieved. Despite the potential improvement, the probabilistic method detailed by Richardson et al does provide configurable variation in on/off cycle times across a population of fridges giving a similar overall profile to the thermal model based population. Thermal modelling becomes more important, however, where demand response is used and the accessible "stored" energy, based on the temperature differential and thermal mass of the fridge contents, must be considered.

Thermal models of entire houses will become important, especially as electrified heating, with heat pumps, is predicted to become more widespread. Bagdanavicious and Jenkins [184] detail a possible approach for including thermal models. Their paper highlights the large variation in eventual heating demand based on property size and insulation quality. Therefore, if such models were to be introduced into the stochastic end-use models reviewed here, significant research into the distributions of property size and insulation quality within populations of households would need to be undertaken. This is important as heat pumps are a promising candidate technology for demand response. However, so long as the proportion of households that have electrified heating remains low, the thermal modelling problem is not significantly problematic for end use models.

2.8 Investigation of phase imbalance with distributed generation

Shahnia et al [172] performed an analysis of voltage imbalance in residential low voltage distribution networks with photovoltaics. They used a Monte Carlo Simulation technique to pick the PV rating, the PV location and the number of households with PV. They performed load flow analysis and reported a significant increase in voltage imbalance at the end of the feeder when PV is added to the feeder, dependent on the location and rating of the PV. Shahnia et al used VUF, along the feeder to measure the imbalance. They did not appear to look at the impact on the imbalance on the magnitude of the load voltages.

Imgram et al [185] produced a generic UK distribution network and examined the effect of increasing embedded generation on voltage unbalance. They increased the generation on a single phase of the low voltage system and observed the unbalance exceeding the P29 limit of 1.3% when there is a 150% penetration of 1.1kW generators. Trichakis et al [186] modelled, using PSCAD, a network based on Ingram et al's and agreed with their results.

In a later paper, Trichakis et al [91] predict a per-phase allowable volume in kW, assuming the voltage unbalance factor as the sole limit. They did this for varied generation and load power factor and varied network impedance. Varying the impedance had the greatest effect. They also noted the significance of the four wire system and neutral impedance, describing a factor of 6 voltage rise increase if generation is placed on one phase as opposed to being balanced across three.

Thomson and Infield [187] performed unbalanced load flow on a distribution network model, using 1-minute interval, stochastically created, load data. Their results

suggest that no significant problems occur, even with high PV penetration. However, they did not vary the number of households connected per phase.

Navarro et al's [188] "Monte Carlo based assessment of PV impacts on real UK Low Voltage Networks" used 5 minute resolution load data and varied the amount of PV. They showed that there are no voltage problems for small feeders (less than 35 customers and shorter than 1km). For longer or more heavily loaded feeders, voltage problems started at 40% PV penetration (the proportion of households with PV).

Su, Masoum and Wolfs [34] note that "The continuous expansion of consumer-driven installations of residential rooftop photovoltaic (PV) systems causes serious power quality, notable voltage variations and unbalance, which limit the number and capacity of the further connections". To reduce these problems, they go on to propose a method for control of PV inverters based on Optimal Power Flow.

Bollen and Hassan [24] stated that the probability that N single phase generators will be connected on the same phase will be $1/(3^N -1)$. This assumes that there is no existing imbalance in single phase connections. They derive an expression using the negative sequence voltage as the limiting factor in the hosting capacity, as it will limit the connection of three phase load.

2.9 The phase identification problem

DNOs commonly do not know which phase houses are connected to. Phase connection information would be valuable to DNOs as it would enable them to better find, analyse and fix networks with high levels of imbalance. Knowing the configuration and network data will allow network operators to better estimate the hosting capacity (for demand and generation) of their LV networks. As noted by Wen at al [189], the body of work relating to phase identification using smart metering is relatively small. The available solutions can be roughly divided into three areas:

- Voltage correlation (similarities between network voltages and smart meter voltages). [189]–[193]
- Comparison of feeder input power measurements and smart meter power measurements. [194]–[197]
- Per-phase signal generation at non power-frequency and detection at the customer's premises. [198]
- Clustering of time series voltage profiles from smart meters. [199], [200]

Pezeshki and Wolfs [190] demonstrated that the phase transformer voltage profile is correlated to the voltage profile recorded at houses connected to the substation feeder. Based on this, they used a cross-correlation method to correctly identify the phase connection of 75 households in Perth, Australia. The technique uses 15 minute voltage measurements from smart meters and voltage measurements at the transformer – measurements over a period of a week. As voltage measurements are required at the substation, a physical connection to the phase conductors at the substation would have to be made (assuming traditional voltage measurement techniques are used), requiring a service interruption. It would be preferable to use a technique based on current measurements at the substation as these could be obtained non-intrusively. It may be that Pezeshki and Wolfs' technique will work adequately with an estimated voltage using current measurements, this would require further investigation, however.

Seal and McGranaghan [191] put forward a phase identification method that uses distribution network SCADA voltage measurements (15 minute intervals) and smart meter voltage measurements (15 minute intervals). The method looks for unique voltage changes on a phase at from the SCADA data and then looks for equivalent changes in the smart meter data. They reported success on an example feeder. However, to achieve this, they had to perform control actions (using a voltage regulator or capacitor) on a single phase. They indicated that more work would be needed to get the algorithm to work with realistic non-deliberate smaller voltage changes.

Like Pezeshki and Wolfs [190], Short's [192] phase identification technique uses correlation between network voltages and smart meter voltages. It uses a Linear Regression based on the substation per phase voltages, the substation mean power, the meter voltages and the meter mean power measurements. Short reported an accuracy of at least 95%. Short also uses this approach to estimate the circuit parameters (resistance and reactance), after the phase connections have been established.

Arya et al [195] used smart meter power measurements and transformer power measurements to identify the phase connection of households. They interpret the problem as a subset sum problem in their patent [194]. They state that the simplest option is to iterate over all possible solutions.

Cormen et al [201] define the subset sum problem as follows; "… we are given a finite set $S \subset \boldsymbol{N}$ and a target $t \in \boldsymbol{N}$. We ask whether there is a subset $S' \subseteq S$ whose elements sum to t." When applied to the phase identification problem, the subset sum problem can be stated as; Given a set of numbers (say current measured by smart meters) and a smaller set of totals (say the total current entering the feeder at the substation end, per phase) find the subsets that sum to give the totals (these sets would then be the groupings of meters connected to each phase). In reality, the subsets would not exactly be equal to the totals due to, for example, measurement errors, synchronisation errors (different clock setting across smart meters) and (if power measurements are used) line losses.

Arya et al formulate and solve the problem using Integer Linear Programming and Integer Quadratic Programming techniques. These appear to work well, for low levels of losses, provided that the number of independent measurements is greater than half the number of meters. However in cases with losses or clock skew error, their algorithm was not completely successful. Given enough measurements however, these errors may be overcome [195]. The tests performed by Arya et al did not include a network model. Instead, they sampled percentage errors for each load from a Gaussian distribution with a 1-3% standard deviation.

Like Pezeshki and Wolfs, Arya et al rely on voltage measurements at the substation. Again, this can perhaps be avoided if the subset sum problem were to be based on currents rather than power (as used by Arya et al). This has the further advantage that inaccuracies due to line losses are avoided. However, inaccuracy may be introduced due to the necessary calculation of currents at each smart meter using averaged values.

Fan et al [196] assume that smart meters will record power factor and that the real and imaginary components for voltage and current can be calculated. From this, they use a formulation similar to that of Arya et al to perform phase identification. The algorithm did not correctly predict the connection of all meters in any of their test cases.

Caird's patent [198] for meter phase identification discusses the use of a low frequency (<60Hz) signal generator on the distribution. The intention is that distinguishable signals are injected on each phase at, say, the local substation. Smart meters must then be capable of interpreting these signals and communicating which of them they have received. From this, the phase connection

of the meters can be established. The method has the drawback of requiring signal generation equipment. Also, smart meters are not expected to monitor and report frequency. However, the British smart meters will be capable of reporting a total harmonic distortion value – which is dependent on frequency content. It is therefore possible that Caird's method could be modified to make use of this functionality.

Dilek [197] uses system measurements and customer load information with a tabu search algorithm to perform phase prediction. Tabu search is described by its creator, Fred Glover [202], as "a meta-heuristic that guides a local heuristic search procedure to explore the search space beyond local optimality". Heuristics are widely understood to be "criteria, methods or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal" [203]. The term meta-heuristic "refers to a master strategy that guides and modifies other heuristics to produce solutions that are beyond those that are normally generated in a quest for local optimality" [203]. Dilek's algorithm, therefore, does not search all possible phase configurations; it instead trims down the search area based on pre-defined expectations. Therefore the true solution may lie outside the tested combinations. In Dilek's test runs, the algorithm was successful in 21 out of 24 cases.

Chen, Ku and Lin [193] performed phase identification by comparison of GPS synchronised voltage angle measurements at the load and on the network. Similarly, Wen et al [189] proposed a method that uses correlation between voltage magnitudes and similarities in voltage angle changes to identify phases. The method uses micro synchrophasor measurement units to obtain voltage magnitudes and angles in the distribution network. The method relies on obtaining voltage phasor measurements in the customer's networks – the authors suggest the use of a plug-in (to a standard socket outlet) to achieve this. As both methods require voltage angle measurements and, in the case of Chen et al, GPS synchronisation, they are unlikely to be usable with smart meters (based on the smart metering proposals in Britain).

Another approach, put forward by Arya and Mitra [200], uses clustering of smart meter time series voltage profiles. Their approach uses the K-means clustering algorithm and accurately grouped the meters per phase in their experiments. The time series profiles used were at 10-300 second sampling intervals over a (up to and including) 5 hour timeframe. They observed that "averaging over longer time intervals can lead to loss of accuracy as the variations in voltage may decrease".

In a following paper, Arya, Mitra et al [199] used the K-means clustering algorithm on 5 minute mean RMS voltage time series data spread over a up to 5 days. They examined 1500 single phase loads and correctly inferred the phase connection for 93% of them. Whilst not accurate in all cases, this technique has the advantage that no supporting network measurements are required.

The K-means algorithm is described by MacKay [204] . It is an algorithm for sorting N data points into K clusters. When applied to the phase identification problem, there are N voltage *profiles* (sets of data points collected over a given timeframe) rather than data points. The number of clusters, K, is 3 (assuming a three phase system with no “empty” phases).

The K-means algorithm begins with an initialisation of the K means – these are typically randomly assigned. The algorithm is then an iterative two-step algorithm consisting of assignment and update steps [204]. Paraphrasing Mackay, in the assignment step, each profile is assigned to the nearest mean. In the update step, the means are adjusted to match the sample means of the data points that they represent [204]. This assignment and update approach is repeated until the assignments do not change.

The outcome of the K-means algorithm is dependent on the initially selected means. Repeat runs may therefore be needed to get confidence in the result. Mackay [204] also points out that, in practice, one might want points nearer the boundary of a cluster to have a lower weighting in determining the mean.

The K-means algorithm is classified as a distance based algorithm. Aggarwal and Reddy [205] categorise distance based algorithms as either flat or hierarchical. In flat algorithms, such as K-means, the clustering is done without iterating. Whereas, in hierarchical clustering it is done iteratively; either starting from the data points and clustering the closest step-by-step – agglomerative, or putting the data points in a single cluster and dividing it step-by-step – divisive. Distance based algorithms are popular as they are relatively simple, easy to implement and flexible (able to be used with a wide range of data types) [205].

3 Modelling of an Unbalanced Low Voltage Feeder

3.1 Introduction

A method for modelling an unbalanced Low Voltage (LV) feeder, allowing for flexible definition of connection imbalance, was developed. The method was validated using PSCAD. The method adapts a feeder model put forward by Ingram, Probert and Jackson [185], to allow for the connection of individual households to different phases. This allows investigation of different configurations and, therefore, different levels of imbalance in the numbers of households per phase. The load-flow method put forward by Berg, Hawkins and Pleines [93] was adapted to accept the different configurations and process the output. The method was implemented using Numerical Python. The resulting software model accepts the per-household real and reactive power load and/or generation. It outputs voltages and currents at the dwelling meters and the losses in the feeder.

The approach is useful because it can be used for repeated load flow solutions, of the order of millions, on a supercomputer – which is done in Chapter 4. From this, realistic smart meter voltage profiles, which include the effect of imbalance due to variation in both demand (and/or generation) and number of houses per phase, were obtained. Real data of this type is not readily available to researchers. The model makes possible the assessment of distributed generation hosting capacity (Chapter 4) and the use of smart meters for phase identification (Chapter 5).

3.2 Basis for the LV feeder model

The LV distribution feeder model produced by Ingram, Probert and Jackson [185] is commonly used for studies of LV systems [91][206][207]. The LV feeder consists of a 300m long trunk cable with service cables tapped off to supply 96 dwellings. The trunk cable is tapered and the loads are evenly distributed along it. Each of the 96 loads has the same 1.3kVA value (the given After Diversity Maximum Demand). Also, each of the 96 loads is single phase. However, Ingram et al model them as 32 balanced star-connected three phase loads. An exception to this is when they consider imbalance in a simulation where the load is entirely removed from one phase. The LV feeder model forms part of a larger model produced by Ingram et al. A diagram of the model is reproduced in Figure 3.1 [185].

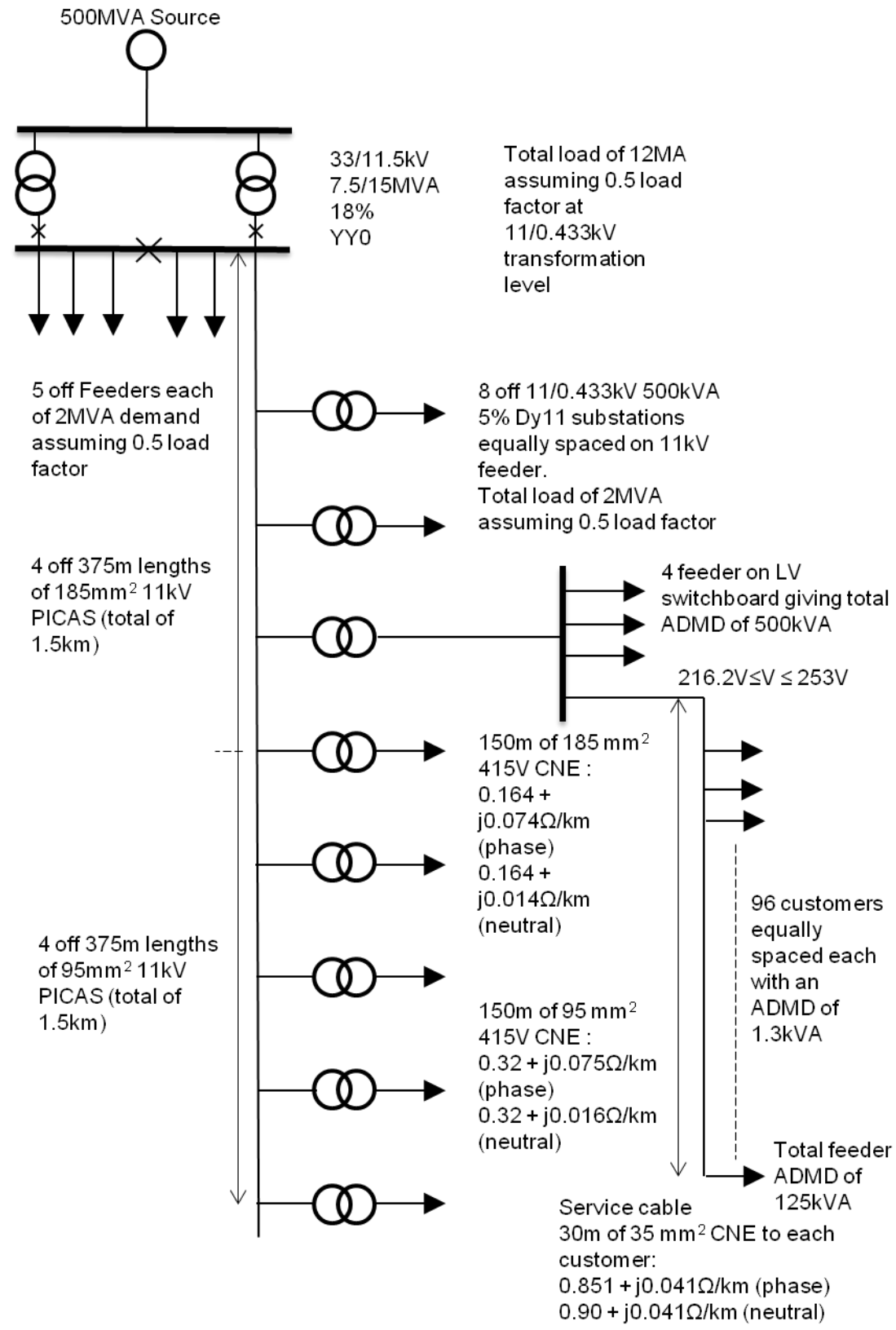


Figure 3.1 - Test network used by Ingram, Probert and Jackson.

The LV portion of Ingram et al's model was interpreted as shown in Figure 3.2a. The feeder was divided into 32 "segments", each consisting of a portion of the trunk cable impedances ($\mathbf{Z}_{Li}$ and $\mathbf{Z}_{N}$), the service cable impedances ($\mathbf{Z}_{serv}$) and the impedance of all the demand (and generation) within 3 households ($\mathbf{Z}_{load}$ and R_{gen}). For ease of reference, the impedance of all the demand (and generation) within a household is referred to as "load" impedance. Each load impedance is in series with a service cable line impedance and a service cable neutral impedance – this is referred to as a "leg". In Figure 3.2a, the legs are connected in star arrangement. The star point represents the point at which the three service cable neutral conductors meet the neutral of the main trunk conductor. The feeder model represents one feeding an urban street with terraced, or closely packed, housing.

The demand from any given three dwellings is unlikely to be identical. Furthermore, there is evidence that many LV networks in Britain have large imbalances in the connection of single phase loads. This is likely to be because cable jointers are (or were) more likely to connect to the top wire (and neutral) within multicore cables [8]. As the top wire is likely to be one of the three phases, it is likely that more of the loads are connected to one phase than either of the other two.

3.3 Connection Imbalance

To consider imbalance in single phase connections, or "connection imbalance", each of the segments within Figure 3.2a can be rearranged so that some phase conductors support more loads than others – an example of this is shown in Figure 3.2b. Figure 3.3 shows the 10 possible arrangements for each segment. Each diagram represents a possible arrangement of 1 segment (see Figure 3.2). The numbers in brackets are the configuration codes assigned to the adjacent segment. Each arrangement is given a code related to the number of households connected to each phase. A balanced arrangement (one household on each phase) is given the code [1,1,1]. Another example might be where phase 1 supports two households, phase 2 one household and phase 3 zero households. This would be given the code [2,1,0]. The number of loads per segment is fixed at 3. Modelling variation in connection imbalance was structured in this way.

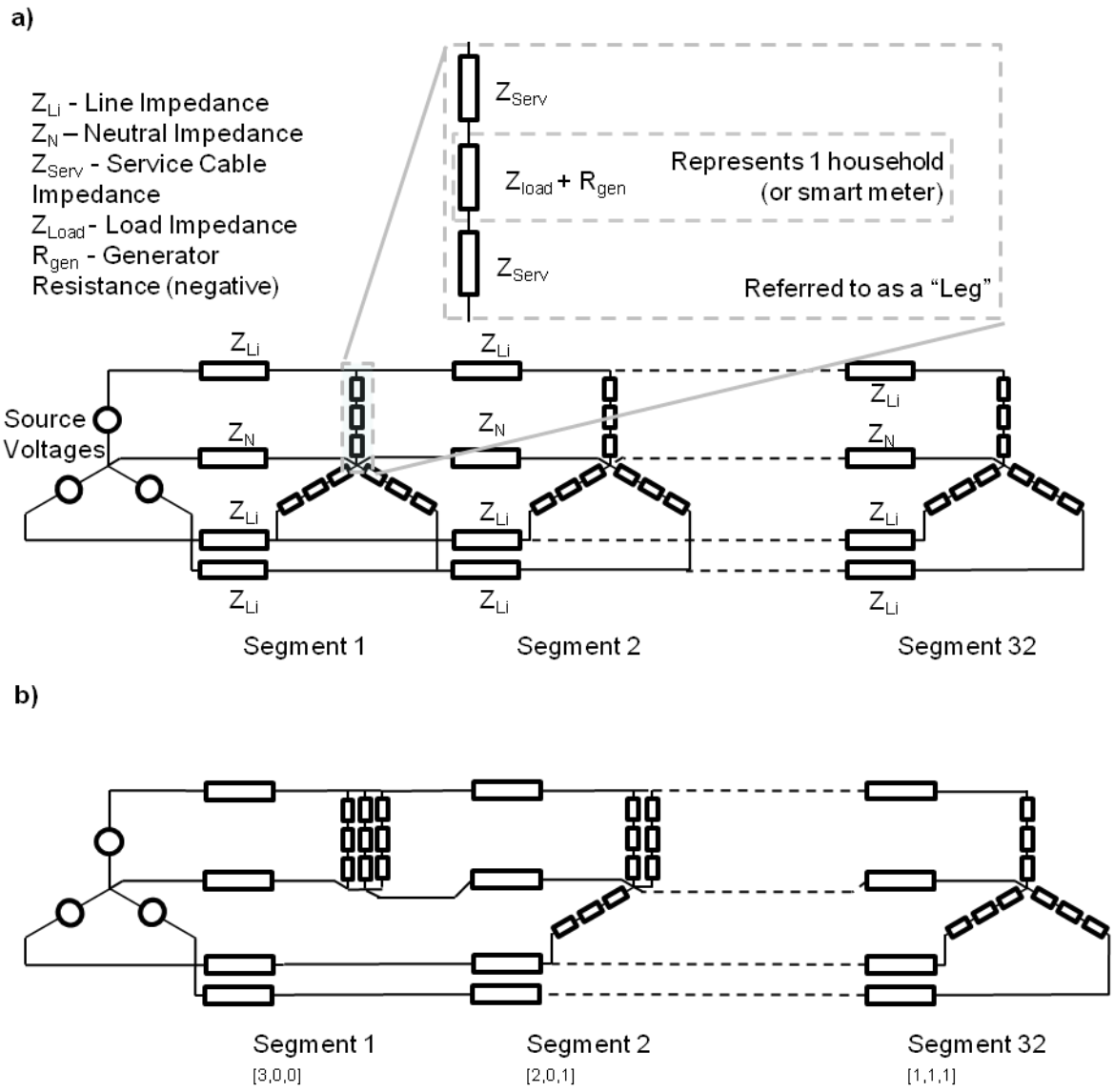


Figure 3.2 - The LV feeder model. a) based on Ingram, Probert and Jackson [185] **and b) illustrating the approach with which connection imbalance is modelled.**

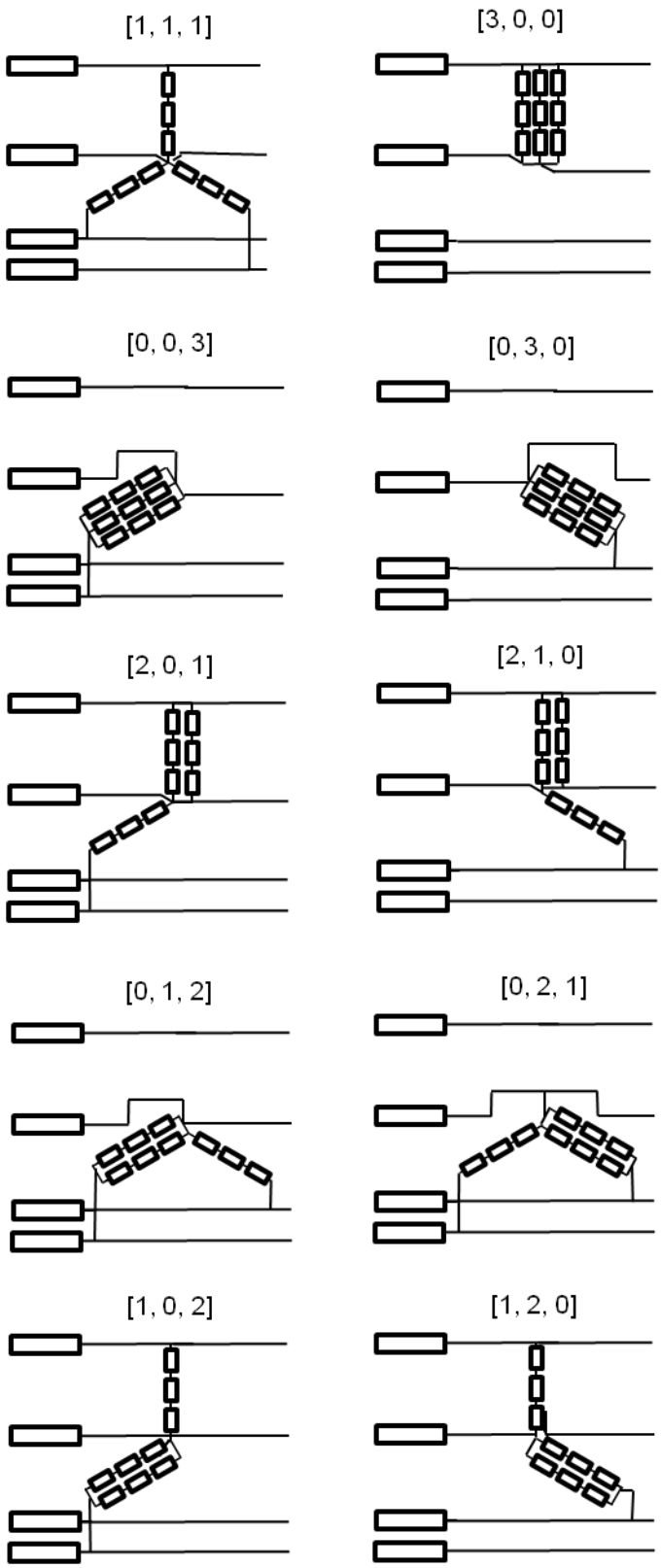


Figure 3.3 - Phase configuration codes used in the LV feeder model.

3.4 Finding the voltages and currents in the LV feeder

The iterative method put forward by Berg, Hawkins and Pleines [93] was used for the solution (calculation of voltages at all points for given supply voltage, line admittances and load admittances) of the feeder. An overview of the method is illustrated, for a four segment feeder, in Figure 3.4. The method relies on a series of star to mesh circuit reductions[1] followed by a sequence of nodal analysis[2] and circuit re-expansions. The end result is that the voltages are known at all junction points on the feeder. From this other information, such as line currents, load currents, load voltages and losses, can be calculated.

If the load admittances are known, the four segment example feeder can be solved in 7 steps as shown in Figure 3.4. If, however, the loads are given in terms of real power, P, and reactive power, Q, then the load admittances must first be approximated using a nominal voltage (the known supply voltage is used), see Equation 3.4. After the feeder has been solved (steps 1-7 in Figure 3.4) the load admittances are recalculated using the voltages from the previous step. The feeder voltages are then calculated again using the updated load admittances. This solution of the feeder, with iteratively updated voltages and load admittances, is repeated until all the load voltage magnitudes converge to stopping criteria, e.g. all meter voltage magnitudes (voltage magnitude at each household's point of connection) within 0.001% of the value from the previous step.

Generators are included in the model through the use of negative resistance. Typically, generators will be specified in terms of real power, P. A simplified illustration of the procedure with a generator is shown in Figure 3.5. The Maximum Power Point Trackers (MPPTs) installed with photovoltaic (PV) generators mean that PV generation can be approximated as a negative fixed power load (i.e. the power output is kept approximately constant over a range of AC terminal voltages). Furthermore, it was assumed that the associated inverters ensure a unity power factor – so any reactive power from generators was ignored, this is commonly done in the literature [187], [188], [208] as PV inverters are frequently set up to operate at unity power factor [209]–[211].

[1] See Appendix 3.1
[2] See Appendix 3.2

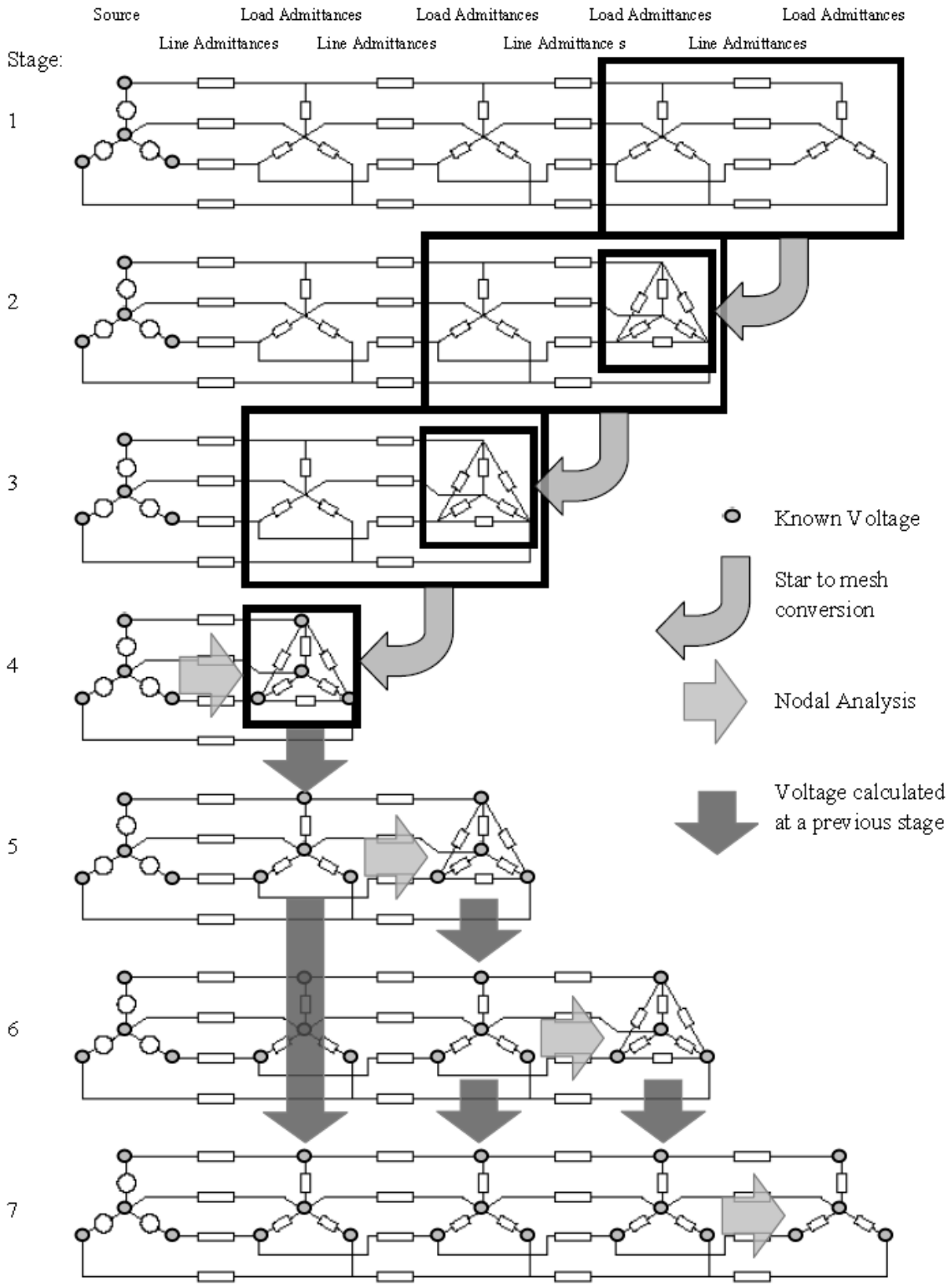
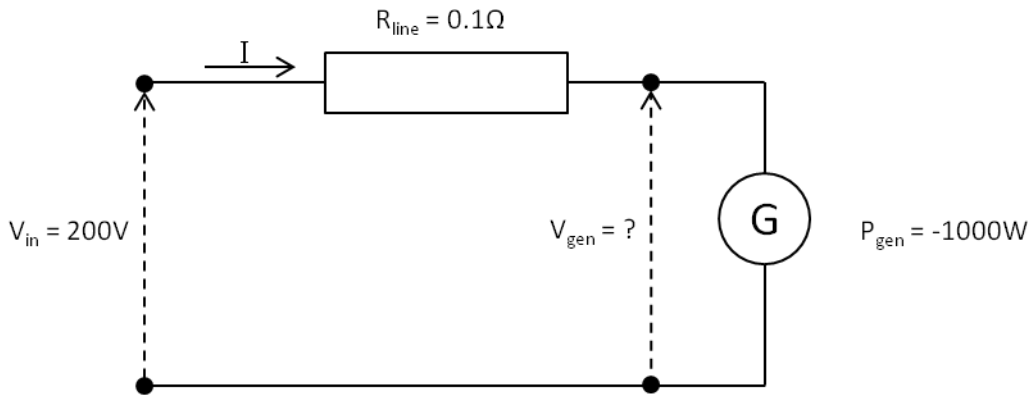


Figure 3.4 - Illustration of the load flow procedure described by Berg, Hawkins and Pleines [93]**, using an example with 4 sets of 3.**



Assume V_{load} = 200V

Convergence error threshold = 0.001 V

Iteration 1:

$R_{gen} = V_{gen}^2/P_{gen}$ = -40 Ω

$I = V_{in}/(R_{line} + R_{gen})$ = -5.012531328 A

$V_{gen}^{(1)} = I \times R_{gen}$ = 200.5012531 V

Iteration 2:

$R_{gen} = V_{gen}^2/P_{gen}$ = -40.20075251 Ω

$I = V_{in}/(R_{line} + R_{gen})$ = -4.987437579 A

$V_{gen}^{(2)} = I \times R_{load}$ = 200.4987438 V

$|V_{gen}^{(2)} – V_{gen}^{(1)}|$ = 0.0025093 V

Iteration 3:

$R_{gen} = V_{gen}^2/P_{gen}$ = -40.19974625 Ω

$I = V_{in}/(R_{line} + R_{gen})$ = -4.987562733 A

$V_{load}^{(3)} = I \times R_{gen}$ = 200.4987563 V

$|V_{gen}^{(3)} – V_{gen}^{(2)}|$ = 0.0000125 V → Less than error threshold (0.001 V) therefore 200.4987563V selected as the final value for V_{gen}..

Figure 3.5 –Iterative method for solution of networks with fixed power generators

The input data for a balanced segment, [1,1,1] in Figure 3.4, are shown in Figure 3.6. The data are the trunk cable line impedances (Z_{line}), the trunk cable neutral impedance (Z_{neut}), the service cable impedances ($Z_{service}$) , the load real (P_{load}) and reactive (Q_{load}) power and the generator real power (P_{gen}). The first step is to convert the load and generator Ps and Q to an impedance value. Following this, to avoid the use of infinity values (for open circuits) in the star to mesh conversion and to simplify the procedure later, the impedances were converted to admittances, see Equations 3.1 to 3.6.

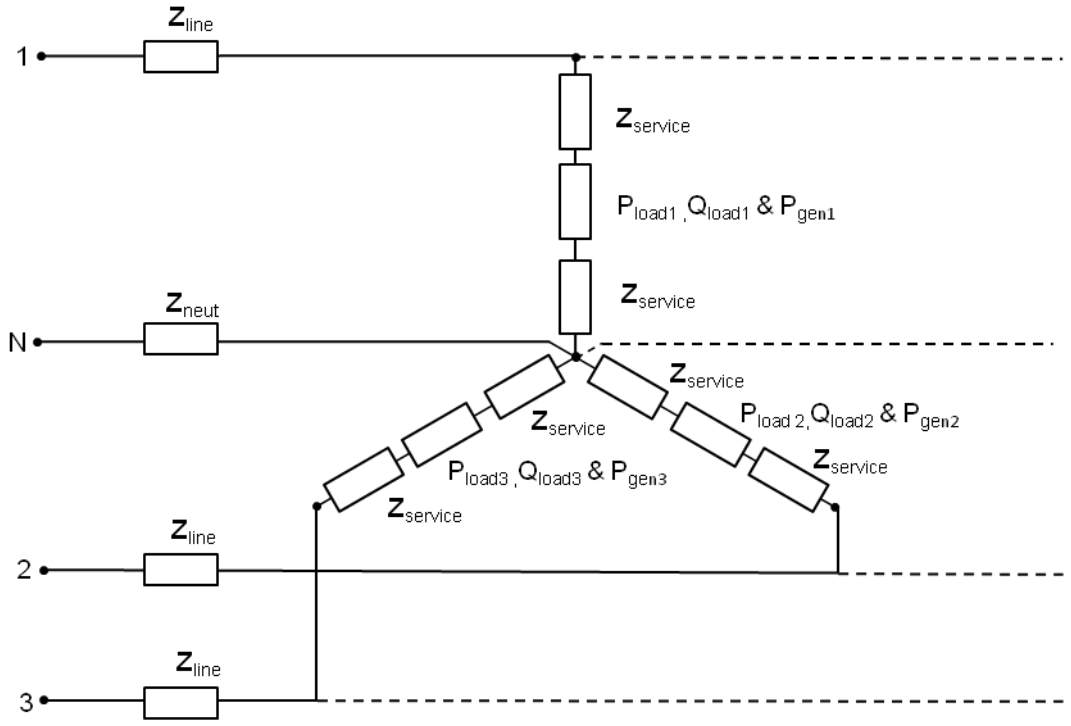


Figure 3.6 - The input data to the model for a balanced segment.

$$\boldsymbol{Y}_{line} = \frac{1}{\boldsymbol{Z}_{line}} \qquad \textbf{Equation 3.1}$$

$$\boldsymbol{Y}_{neut} = \frac{1}{\boldsymbol{Z}_{neut}} \qquad \textbf{Equation 3.2}$$

$$\boldsymbol{S}_{meter\varphi} = P_{load\varphi} - P_{gen\varphi} + jQ_{load\varphi} \qquad \textbf{Equation 3.3}$$

Where S_{meter} is the apparent power seen by the household and φ is the phase number (1, 2 or 3).

$$\boldsymbol{Z}_{meter\varphi} = \frac{\boldsymbol{V}_{slack\varphi}{}^{2}}{\boldsymbol{S}_{meter\varphi}} \qquad \textbf{Equation 3.4}$$

The input slack voltage, $\boldsymbol{V}_{slack\varphi}$ (the line to neutral voltage at the feeder terminals in the 11:0.433kV substation), is used for the initial calculation. In subsequent iterations the calculated voltages from the previous iteration are used.

$$\boldsymbol{Z}_{leg\varphi} = \boldsymbol{Z}_{meter\varphi} + 2\boldsymbol{Z}_{service} \quad \textbf{Equation 3.5}$$

$$\boldsymbol{Y}_{leg\varphi} = \frac{1}{\boldsymbol{Z}_{leg\varphi}} \quad \textbf{Equation 3.6}$$

Where $\boldsymbol{Z}_{leg\varphi}$ and $\boldsymbol{Y}_{leg\varphi}$ are the equivalent impedance and admittance between the cable joints on the phase conductor and neutral conductor in the trunk[3] cable. Note the 2 $\boldsymbol{Z}_{service}$ values used in Equation 3.5 – this represents the phase conductor (the cable core from the trunk cable to the supply, or "live") and the neutral conductor (the cable core from the meter point and the cable core returning from the neutral meter point to the trunk cable). The result is represented in Figure 3.7.

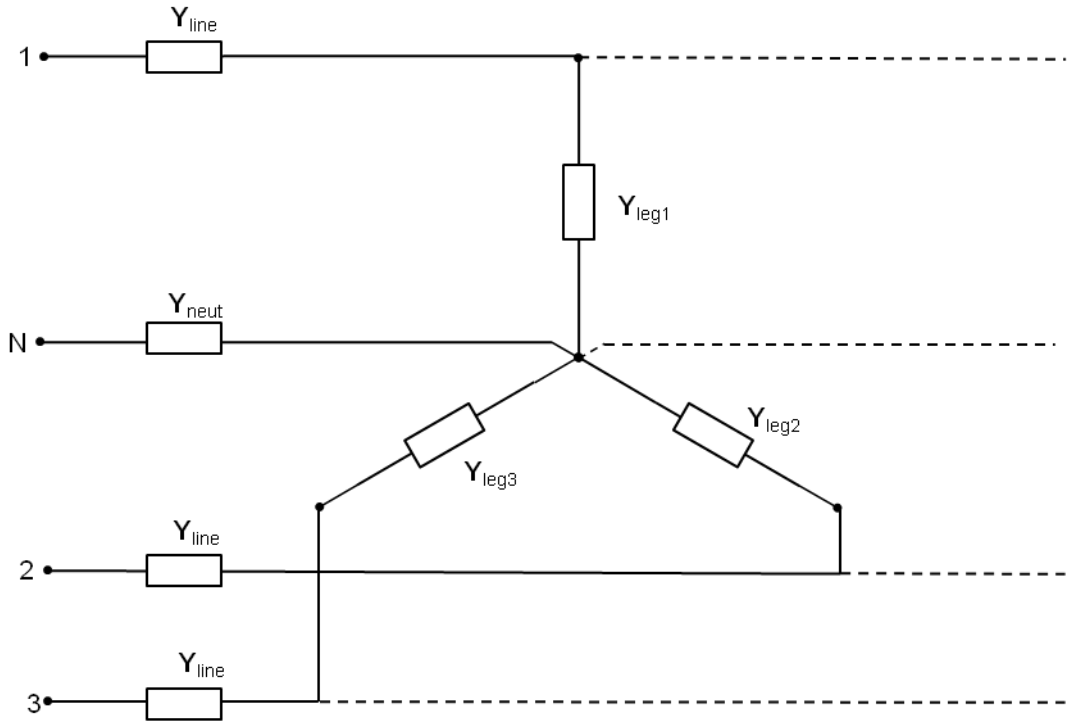


Figure 3.7- A balanced segment after calculation of "leg" admittances.

If the phase configuration of a segment (as defined in Figure 3.3) is unbalanced, the admittances must be adjusted. Firstly, each load (and/or generator) impedance and service impedances are combined (added, as they are in series) and inverted to give a total admittance value for a circuit for each household; from the trunk cable phase conductor to the household and back to the trunk cable neutral conductor –

[3] Trunk cable refers to the main conductor from which households are connected via service cables. E.g. A cable running beneath a road on an urban street.

referred to using the “leg” suffix. If the phase configuration is balanced, [1,1,1] in Figure 3.3, then the calculated admittances can be used in the load flow directly (as per Figure 3.4).

However, if there is an unbalanced phase configuration then, depending on the configuration, at least one of the input admittances will be set to zero and at least two of the admittances will be added as they represent parallel connections to the same phase. For example, this occurs if two loads are on phase 1, one is on phase 2 and zero are on phase 3, as in Figure 3.8. Note that this configuration is assigned the code [2,1,0], see Figure 3.3.

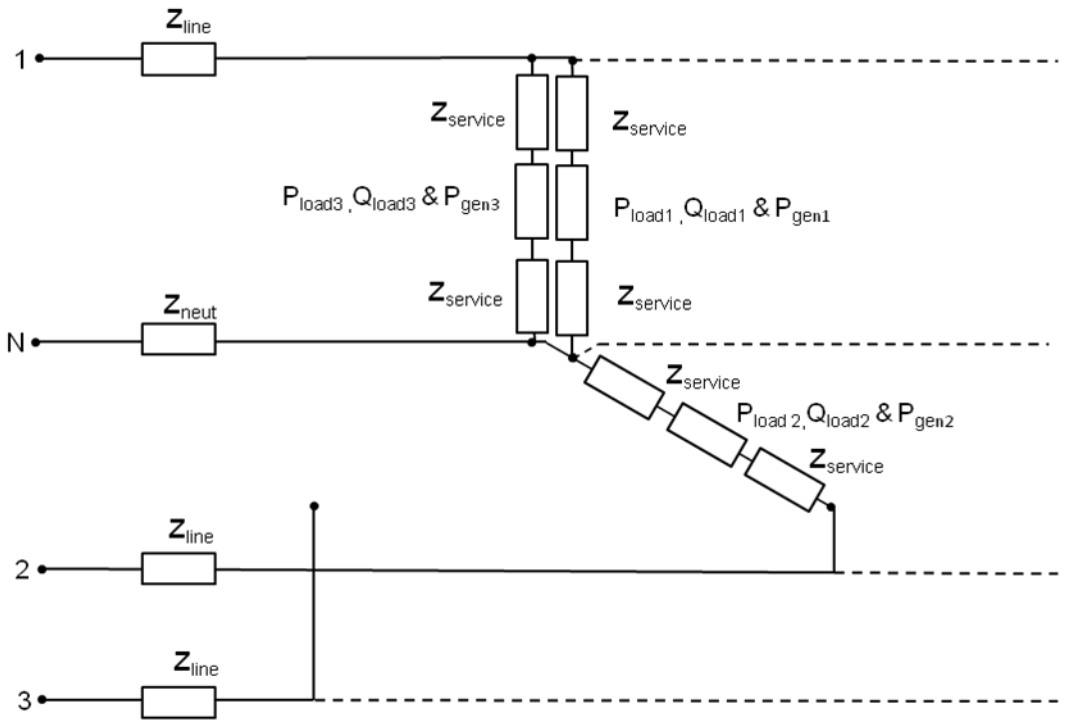


Figure 3.8 – Input data for an unbalanced segment with configuration code [2,1,0].

The calculation of the leg admittances proceeds as in the balanced, [1,1,1], case. However, the admittance for leg 3 must be transferred to phase 1, this is done by adding the calculated leg 3 and leg 1 admittances, see Equation 3.7.

$$\boldsymbol{Y}_{leg1}{}' = \boldsymbol{Y}_{leg1} + \boldsymbol{Y}_{leg3} \qquad \textbf{Equation 3.7}$$

Where ′ denotes that the admittance has been adjusted to allow for an unbalanced phase configuration. Note that an input is still required for phase 3. As, in this

example, there is nothing connected to phase 3, the admittance value is set to zero (open circuit), see Equation 3.8. The resulting segment is shown in Figure 3.9.

$$\boldsymbol{Y}_{leg3}{}' = 0$$ **Equation 3.8**

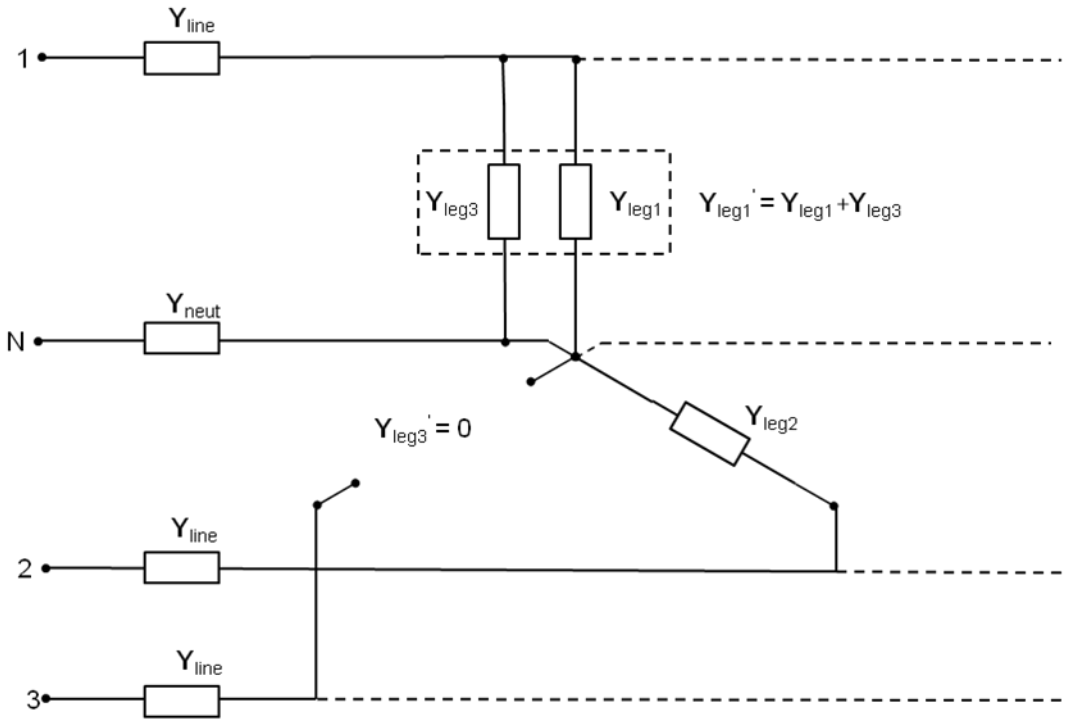


Figure 3.9 - An unbalanced segment, with a configuration code of [2,1,0], after calculation of "leg" admittances.

The meter voltages are calculated using circuit analysis. The circuit analysis involves the calculation of the total voltage drop across each “leg”. Initially, the voltages at the trunk cable tap-off points are output, see Figure 3.10.

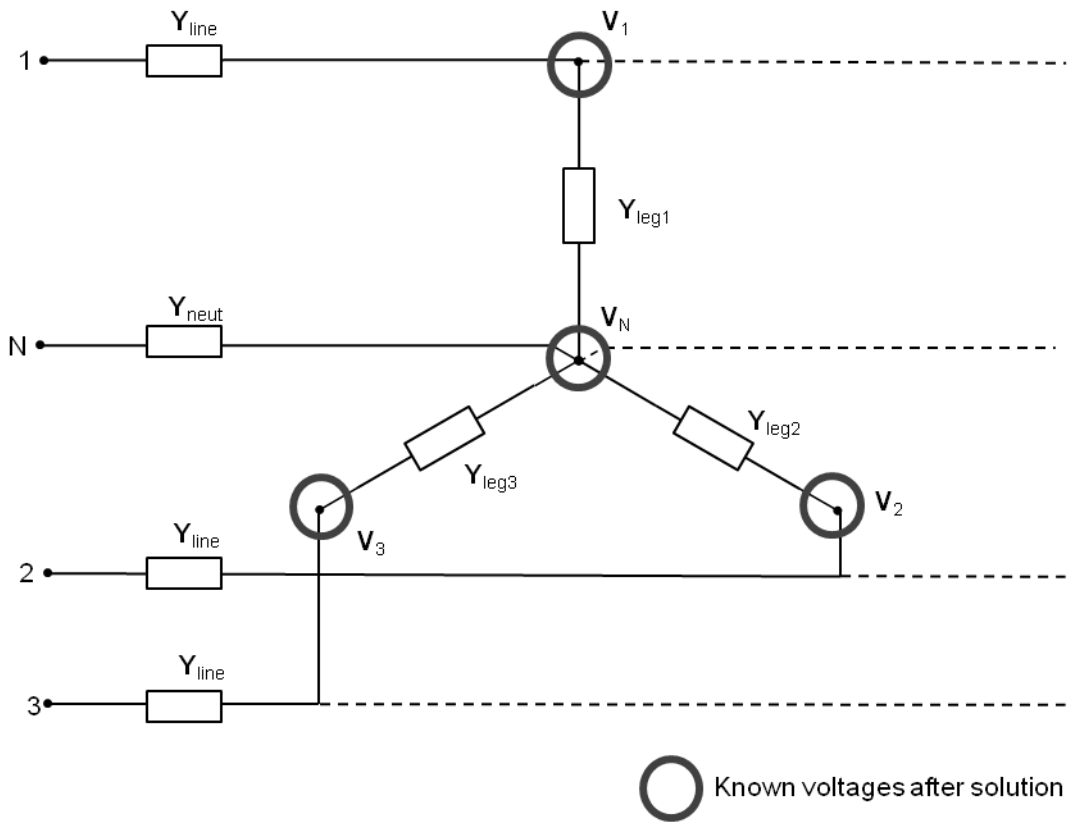


Figure 3.10 – Known voltages calculated for a balanced segment.

The leg voltages are calculated by subtracting the calculated neutral voltage at the trunk cable tap-off point, $\mathbf{V}_N$, from the appropriate phase voltage at the trunk cable tap-off point ($\mathbf{V}_1$, $\mathbf{V}_2$ or $\mathbf{V}_3$), see Equations 3.9 to 3.11.

$$\boldsymbol{V}_{leg1} = \boldsymbol{V}_1 - \boldsymbol{V}_N \qquad \textbf{Equation 3.9}$$

$$\boldsymbol{V}_{leg2} = \boldsymbol{V}_2 - \boldsymbol{V}_N \qquad \textbf{Equation 3.10}$$

$$\boldsymbol{V}_{leg3} = \boldsymbol{V}_3 - \boldsymbol{V}_N \qquad \textbf{Equation 3.11}$$

When an unbalanced phase configuration is used, the voltages used to calculate the leg voltages are decided by the phase configuration. In the [2,1,0] example, shown in Figure 3.11, this would be as shown in Equations 3.12 and 3.13:

$$\boldsymbol{V}_{leg1} = \boldsymbol{V}_{leg3} = \boldsymbol{V}_1 - \boldsymbol{V}_N \qquad \textbf{Equation 3.12}$$

$$\boldsymbol{V}_{leg2} = \boldsymbol{V}_2 - \boldsymbol{V}_N \qquad \textbf{Equation 3.13}$$

Note the use of the phase 1 voltage for the third leg as, in this case, the third meter is connected to phase 1.

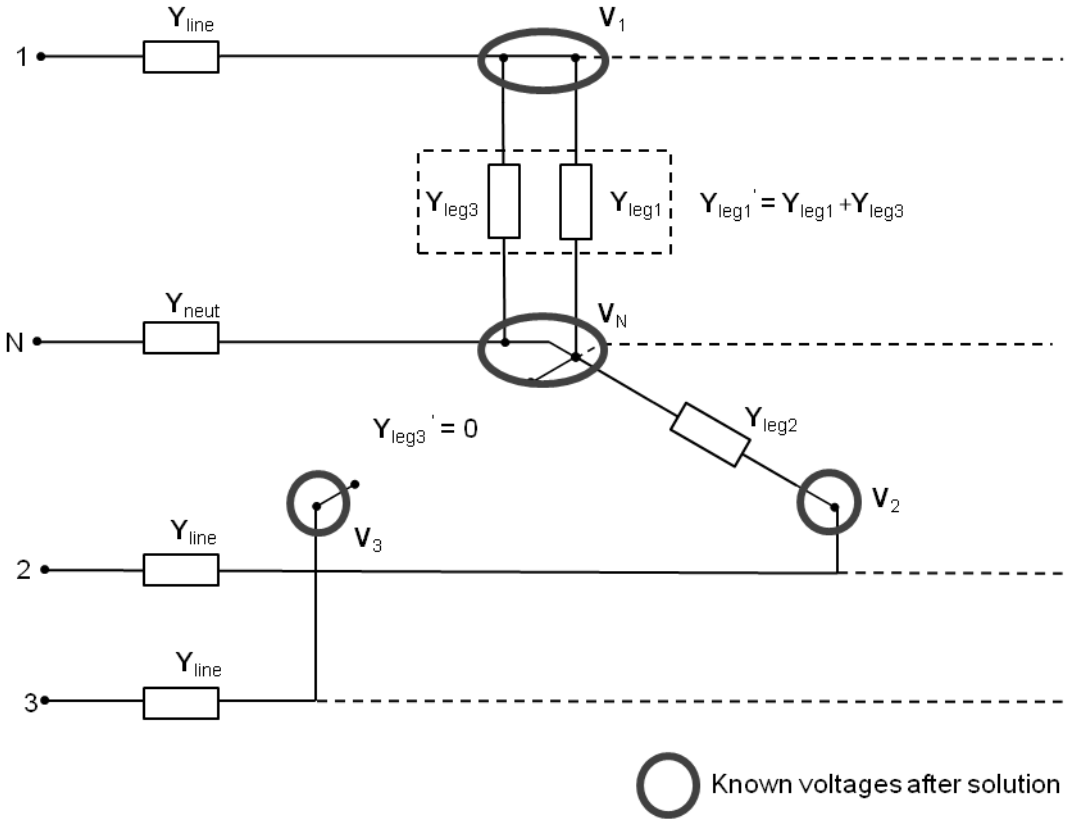


Figure 3.11 – Known voltages calculated for an unbalanced segment with a configuration code of [2,1,0].

Recall that, in Equation 3.4, $\boldsymbol{Z}_{meter\varphi}$ was calculated. This is used, in Equation 3.14, to calculate the meter voltage:

$$\boldsymbol{V}_{meter\varphi} = \boldsymbol{Z}_{meter\varphi} \frac{\boldsymbol{V}_{leg\varphi}}{\left(\boldsymbol{Z}_{meter\varphi} + 2\boldsymbol{Z}_{service}\right)} \qquad \textbf{Equation 3.14}$$

The process is iterative. After calculation, the $\boldsymbol{V}_{meter\varphi}$ values for all segments are compared to the previous guesses. If they are all found to be within 0.001% of each other, convergence has been reached. Else, $\boldsymbol{Z}_{meter\varphi}$ is recalculated using Equation 3.15:

$$\boldsymbol{Z}_{meter\varphi} = \frac{\boldsymbol{V}_{meter\varphi}{}^{2}}{\boldsymbol{S}_{meter\varphi}} \qquad \textbf{Equation 3.15}$$

The process repeats until all the meter voltages converge or until 100 iterations are reached and it is deemed to have failed. When the voltages have been successfully

obtained, the losses and currents are calculated. To calculate the metered current, the leg voltages and admittances are multiplied, see Equations 3.16 to 3.18:

$$\boldsymbol{I}_{leg1} = \boldsymbol{V}_{leg1}\boldsymbol{Y}_{leg1} \quad \textbf{Equation 3.16}$$

$$\boldsymbol{I}_{leg2} = \boldsymbol{V}_{leg2}\boldsymbol{Y}_{leg2} \quad \textbf{Equation 3.17}$$

$$\boldsymbol{I}_{leg3} = \boldsymbol{V}_{leg3}\boldsymbol{Y}_{leg3} \quad \textbf{Equation 3.18}$$

The leg current calculations are the same regardless of the phase connection. Note that the original leg admittances are used (i.e. before admittances were transferred from one phase to another – denoted by the absence of the ′ superscript in Figure 3.9).

The line and neutral currents were calculated using Kirchhoff's current law. The current calculations are done segment by segment, starting from the segment furthest from the substation – the "end" segment. The calculations for the line and neutral currents are shown in Equations 3.19 to 3.22, see also Figure 3.12.

$$\boldsymbol{I}_{line1} = \boldsymbol{I}_{leg1} \quad \textbf{Equation 3.19}$$

$$\boldsymbol{I}_{line2} = \boldsymbol{I}_{leg2} \quad \textbf{Equation 3.20}$$

$$\boldsymbol{I}_{line3} = \boldsymbol{I}_{leg3} \quad \textbf{Equation 3.21}$$

$$\boldsymbol{I}_{neutral} = \boldsymbol{I}_{leg1} + \boldsymbol{I}_{leg2} + \boldsymbol{I}_{leg3} \quad \textbf{Equation 3.22}$$

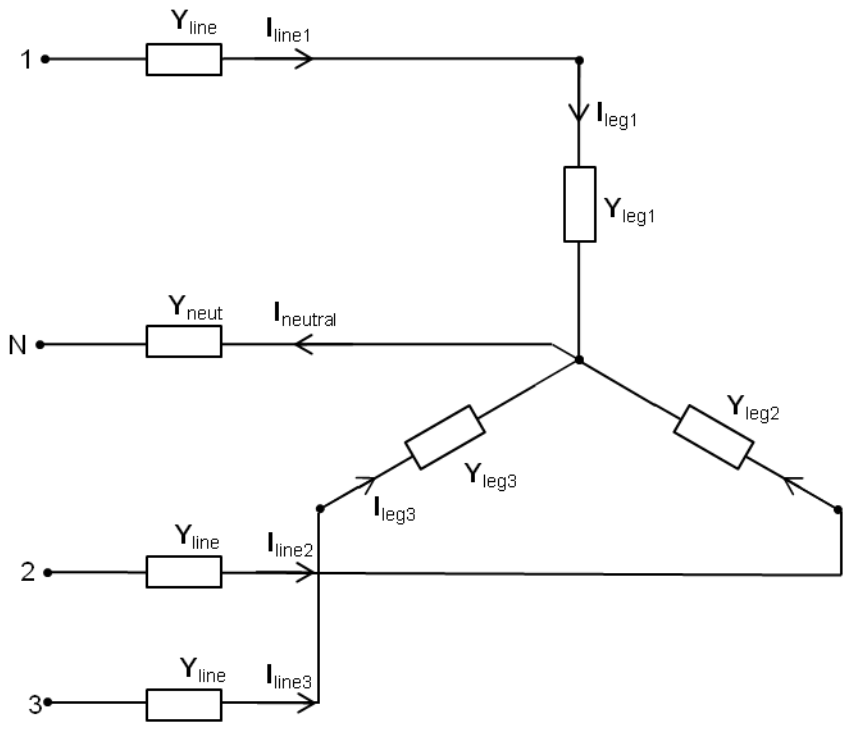


Figure 3.12 - Current labels in a balanced end segment.

For segments other than the end segment the currents flowing to the previous segment must be considered, see Equations 3.23 to 3.26 and Figure 3.13.

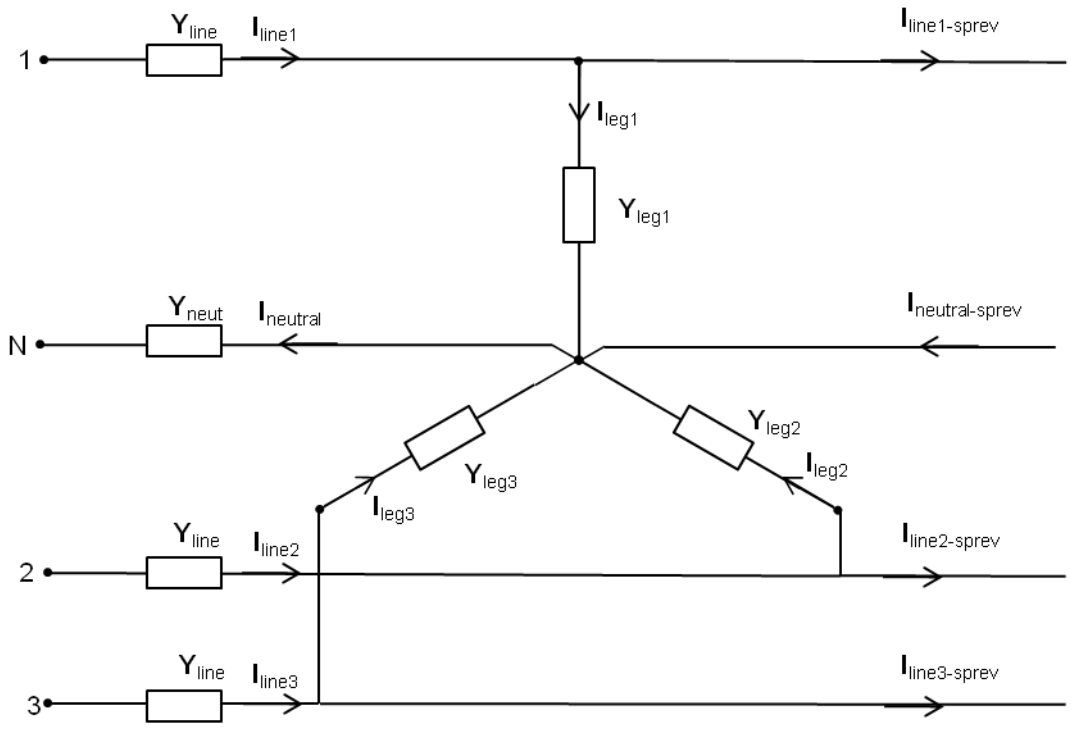


Figure 3.13 - Current labels in a balanced segment.

$$\boldsymbol{I}_{line1} = \boldsymbol{I}_{leg1} + \boldsymbol{I}_{line1-sprev} \quad \textbf{Equation 3.23}$$

$$\boldsymbol{I}_{line2} = \boldsymbol{I}_{leg2} + \boldsymbol{I}_{line2-sprev} \quad \textbf{Equation 3.24}$$

$$\boldsymbol{I}_{line3} = \boldsymbol{I}_{leg3} + \boldsymbol{I}_{line3-sprev} \quad \textbf{Equation 3.25}$$

$$\boldsymbol{I}_{neutral} = \boldsymbol{I}_{leg1} + \boldsymbol{I}_{leg2} + \boldsymbol{I}_{leg3+\boldsymbol{I}_{neutral-sprev}} \quad \textbf{Equation 3.26}$$

The current calculations for different phase configurations (as shown in Figure 3.3) must be adjusted accordingly. For example, for a configuration of [2,1,0] the calculations shown in Equations 3.27 to 3.30 are used:

$$\boldsymbol{I}_{line1} = \boldsymbol{I}_{leg1} + \boldsymbol{I}_{leg3} + \boldsymbol{I}_{line1-sprev} \quad \textbf{Equation 3.27}$$

$$\boldsymbol{I}_{line2} = \boldsymbol{I}_{leg2} + \boldsymbol{I}_{line2-sprev} \quad \textbf{Equation 3.28}$$

$$\boldsymbol{I}_{line3} = + \boldsymbol{I}_{line3-sprev} \quad \textbf{Equation 3.29}$$

$$\boldsymbol{I}_{neutral} = \boldsymbol{I}_{leg1} + \boldsymbol{I}_{leg2} + \boldsymbol{I}_{leg3+\boldsymbol{I}_{neutral-sprev}} \quad \textbf{Equation 3.30}$$

Note that, in Equation 3.27, the current for the third leg has been transferred to the phase 1 line. Similar adjustments are made to the calculations for each segment, based on its phase configuration. Once the line currents for all segments have been obtained, the losses can be calculated, as in Equation 3.31:

$$Losses\ (Wh) = T\left(\sum_{s=1}^{s=32} |\boldsymbol{I}_{line1}|^2 / G_{line1} + |\boldsymbol{I}_{line2}|^2 / G_{line2} + |\boldsymbol{I}_{line3}|^2 / G_{line3} + |\boldsymbol{I}_{neutral}|^2 / G_{neutral} + |\boldsymbol{I}_{service1}|^2 / G_{service1} + |\boldsymbol{I}_{service2}|^2 / G_{service2} + |\boldsymbol{I}_{service3}|^2 / G_{service3} \right)$$

Equation 3.31

Where *T* is the time period in hours over which the load flow applies, *s* is the segment number, I_{linen} is the current through the trunk cable's *n*th phase conductor's resistance (at the *t*th minute in the *s*th segment), $I_{neutral}$ is the current through the segment's trunk cable neutral conductor resistance, $I_{servicen}$ is the current through the service cable (connected to the *n*th phase) phase and neutral conductor(s) resistances. G_n represents the conductance of *n.* For $G_{servicen}$ the conductance includes the two (live and neutral) components.

The complete process for calculation of the voltages, currents and losses on the LV feeder, is shown in Figure 3.14.

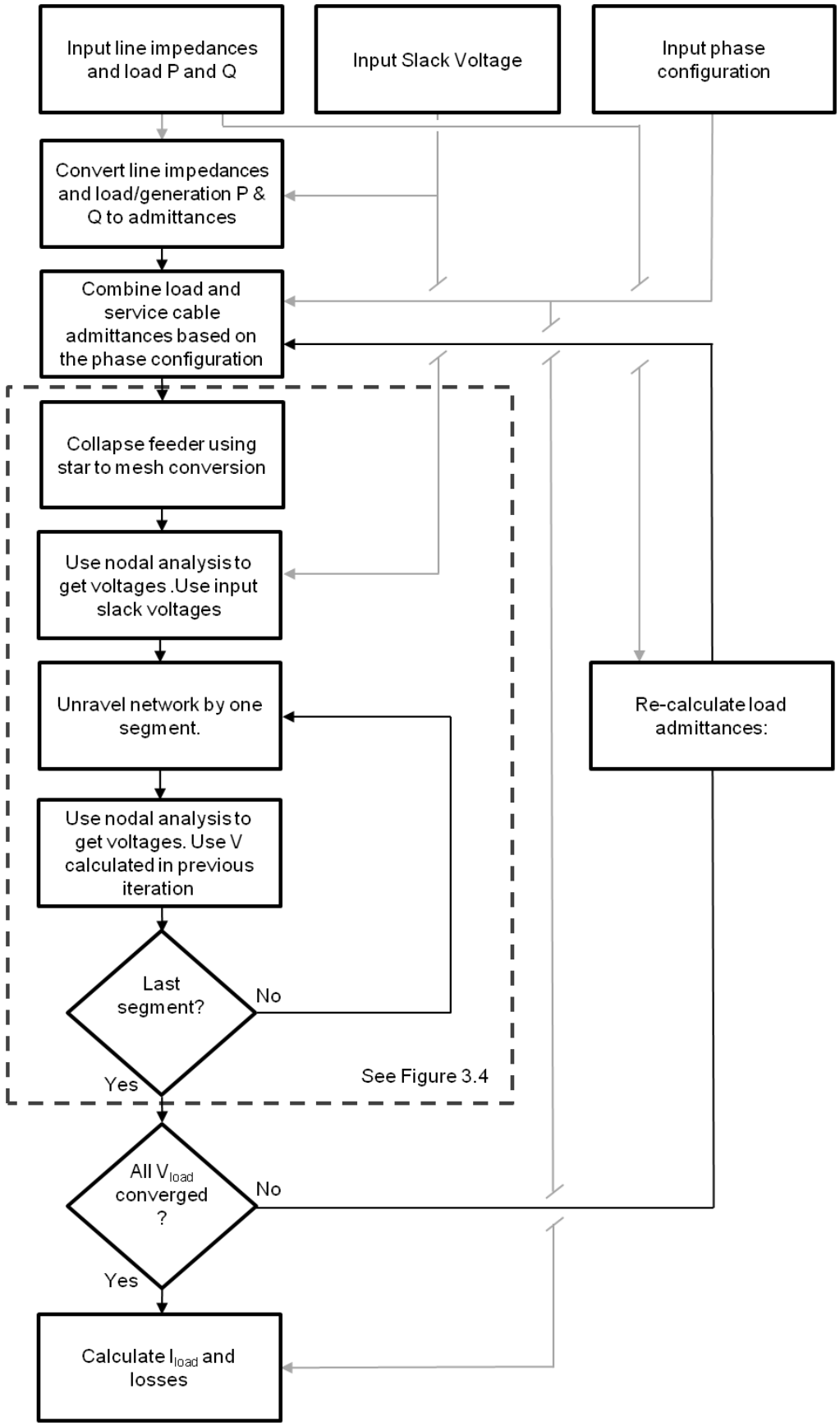
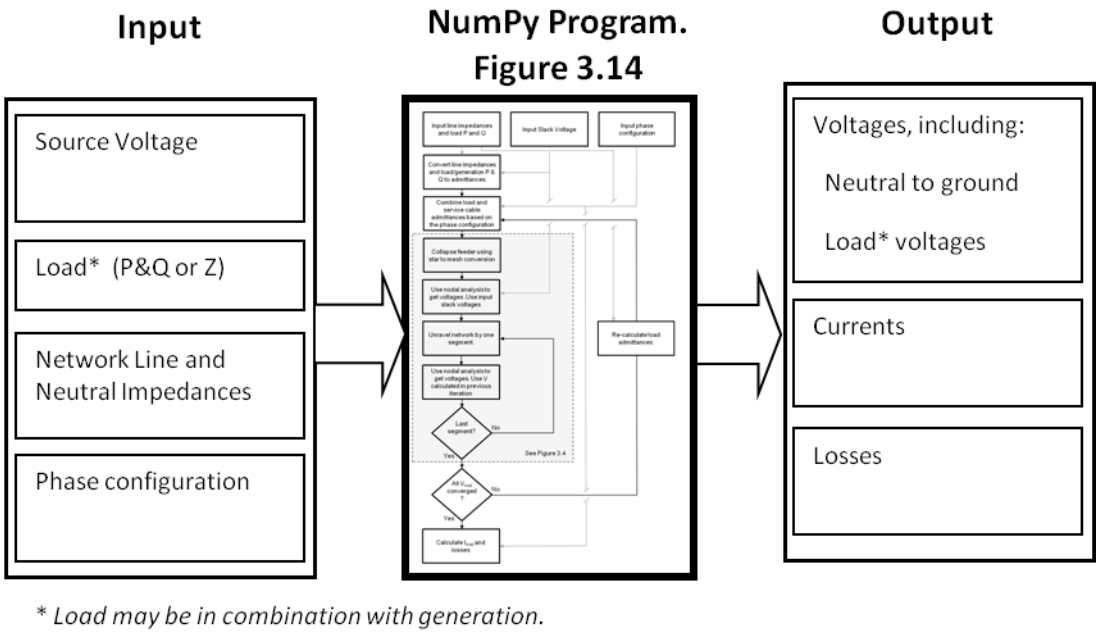


Figure 3.14 - The complete process used for calculation of voltages, currents and losses in an LV feeder with unbalanced phase connections.

Numerical Python (NumPy) [212] was written to model the LV feeder, as described by Figure 3.14. The NumPy program accepts single phase load (and/or generation) values (either P and Q, or impedance), the feeder impedances, the feeder configuration and a fixed source voltage at the substation feeder terminals. It returns the voltages, currents and losses. The script inputs and outputs are shown in Figure 3.15. The script code is shown in Appendix A2.



3.15- NumPy LV Feeder program - Input and outputs

3.5 Validation of the Model

The accuracy of the program output was checked by comparison with an equivalent model built and simulated with PSCAD™/EMTDC™. A number of scenarios were used; balanced and unbalanced phase configurations, consistent and varied load magnitudes (sampled from a gamma distribution) modelled with fixed impedance or fixed power (fixed P and Q) loads. The maximum percentage difference in load voltages, for fixed PQ loads in both the test model and the PSCAD model, was 0.0181% for the unbalanced phase configuration with varied load magnitudes. The same case had a maximum absolute percentage error in the resultant P and Q values of 0.0462%. The outputs are accurate enough to allow investigation of varied connection imbalance.

A feeder with 60 single phase loads, 20 three phase segments, was used for the validation. The impedance values used were based on the cable impedances given by Ingram, Probert and Jackson , see Figure 3.16 and Table 3.1.

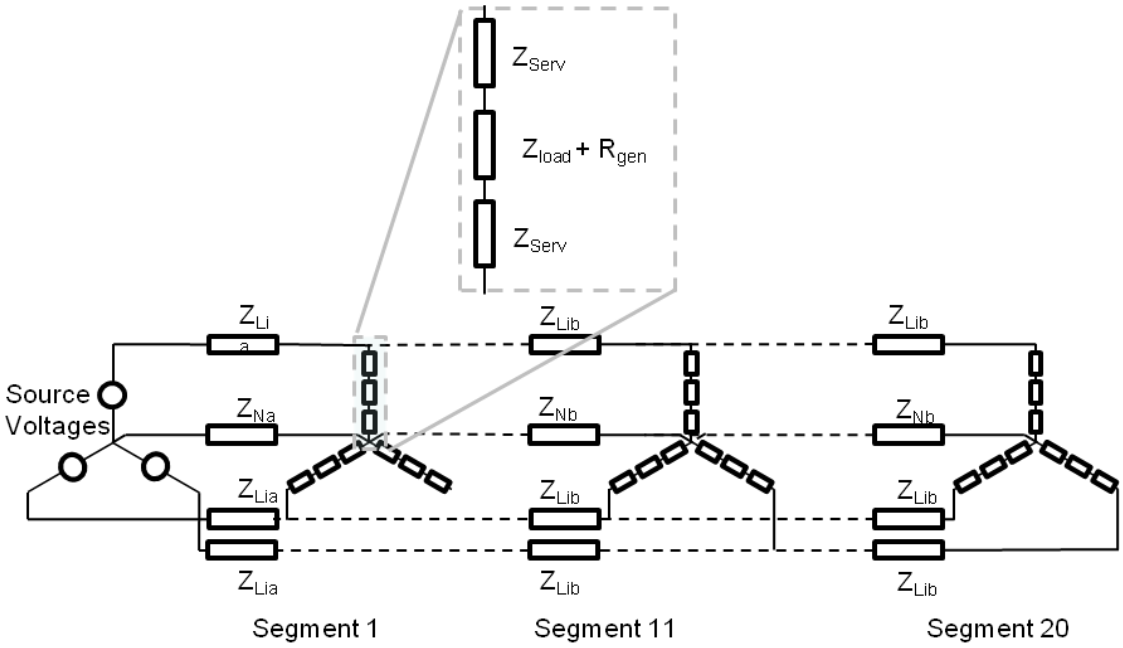


Figure 3.16 - The model used for validation of the NumPy LV feeder model .

Symbol	Description	Value
Z_{linei}	Segments 1-10 line impedances for 185mm^2 CNE cabling (all phase conductors) (15m)	0.00246 + j0.000111Ω
Z_{lineii}	Segments 11-20 line impedance for 95mm^2 CNE cabling (all phase conductors) (15m)	0.0048 + j0.001125Ω
Z_{Ni}	Segments 1-10 impedance of neutral for 185mm^2 CNE cabling (15m)	0.00246 + j0.000111Ω
Z_{Nii}	Segments 11-20 impedance of neutral for 185mm^2 CNE cabling (15m)	0.0048 + j0.00024Ω
Z_{serv}	Impedance of service cable feeding each lumped load (applies to all phases)(30m)	0.0255 + j0.00123 Ω

Table 3.1 - Impedance values used for the validation of the NumPy LV feeder model.

The meter voltages taken from the Python model and PSCAD model were compared. In the PSCAD model a 0.37s “settling time” was, based on a visual assessment of the graphs, chosen to ensure that the load voltages had reached a settled value (the settling time was chosen after inspection of the PSCAD output graphs and the arbitrary picking of an x-axis point in the settled part of the graph - the same settling time was then used for all cases). Table 3.2 shows the resultant maximum absolute percentage difference in meter voltages between PSCAD and the Python model, assuming that the PSCAD values are 100% accurate. This was done for different load types, values and arrangements. The maximum error between any of the calculated meter voltages was 0.0181%. On a base of 230V, this corresponds to approximately 40mV. It is not a significant error.

When fixed PQ loads are used, a mismatch occurs between the input P and Q values and the output values (i.e. the P and Q values calculated from the output voltages and currents). The mean and maximum absolute percentage errors in apparent power magnitude for cases 5 and 6 are shown in Table 3.2 below. The highest error is 0.0462%. For a 3kVA load this corresponds to an error of 1.4VA, again this is not a significant error.

A source of difference between the Python model and the PSCAD/EMTDC is floating point error [213]. The number of arithmetic operations using floating point numbers will differ between the two models – therefore errors can be expected. The results in Tables 3.4 and 3.5 show that the maximum error increases as the system gets less balanced; more iterations are required before convergence, increasing the floating point error. Another source of error is due to the differing techniques used in calculating the voltages. Whilst the source code of PSCAD/EMTDC is not available, the Electromagnetic Transients Program (EMTP, originating in the work of Dommel [214] and Meyer [215]), on which it is based, includes some, relatively small, error due to truncation error in discretisation – the conversion of a continuous signal into a sampled one.

Case	Python Loads	PSCAD Loads	Phase configuration	Maximum absolute % difference in Meter Voltage Magnitude[4], (3 significant figures)
1	Fixed impedance loads, calculated from a power of 1kW at 0.9 power factor at 250V.	Resistors and Inductors, calculated from a power of 1kW at 0.9 power factor at 250V.	Balanced	0.00954
2	Fixed apparent power loads 1kW at 0.9 power factor.	Resistors and Inductors – using values output from the Python Script	Balanced	0.001927
3	Fixed apparent power loads, real power values sampled from a gamma distribution* at 0.9 power factor.	Resistors and Inductors – using values output from the Python Script	Balanced	0.002068
4	Fixed apparent power loads, real power values sampled from a gamma distribution* at 0.9 power factor.	Single phase PQ loads	Balanced	0.0152
5	Fixed apparent power loads, real power values sampled from a gamma distribution* at 0.9 power factor.	Single phase PQ loads	Unbalanced: Selected randomly with a bias towards phase 1 (40%,30%,30%). Totals on each phase were 25, 16 and 19.	0.0181

Table 3.2 - Comparison of Python and PSCAD models of a 60 load LV feeder for 6 cases. * A gamma distribution with shape = 1.5 and scale=1.5. The same sampled values were used for cases 2, 4, 5 and 6.

[4] This assumes that PSCAD values are 100% accurate.

Case	Mean Absolute Percentage Error Input Apparent power magnitude is 100%, this was compared to apparent power magnitude from product of calculated voltage and the complex conjugate of the calculated current.[%]	Maximum Absolute Percentage Error Input Apparent power magnitude is 100%, this was compared to apparent power magnitude from product of calculated voltage and the complex conjugate of the calculated current.[%]
4	0.0264	0.0448
5	0.0275	0.0462

Table 3.3 - The PQ error, the difference between the input P and Q values and the calculated P and Q values based on the NumPy model's output voltages and currents.

3.6 Stress testing of the model

The model was repeatedly run with increasing demand until it failed. The accuracy of the output meter voltage magnitudes remains within 0.2% of PSCAD model until the balanced input (applied to all loads) is above 8.3 kVA (0.9 power factor). This is well above the After Diversity Maximum Demand (ADMD) likely to be seen on a typical feeder (Ingram, Probert and Jackson assume an ADMD of 1.3kVA for example). With an unbalanced load, the input remains well above the typical ADMD when the error reaches 0.2%. For instance, with input loads of 7.8, 6.2 and 9.3kVA applied to all loads on phases L1, L2 and L3, the maximum meter voltage magnitude error is 0.19%. Tables 3.4 and 3.5 show the meter voltage errors and the apparent power errors for increasing balanced and unbalanced demand.

Real Power demand (at 0.9 lagging power factor) applied to all loads [W]	% error in Load Voltage Magnitude, PSCAD values are 100%, 3 significant figures (Mean,Maximum)	% error in Calculated Apparent Power, Input values are 100%, 3 significant figures (Mean,Maximum)	Number of iterations
7500	0.0581,0.0974	0.421,0.567	9
15000	0.668,1.31	2.08,2.97	13
22500	3.41,7.94	6.29,9.83	20
30000	21.8,47.3	19.6,36.8	43
31600	Did not converge		

Table 3.4 - Comparison between PSCAD (fixed power loads) and NumPy (fixed power loads) for increasing balanced load

Real Power demand (at 0.9 lagging power factor) [W] applied to all loads on a phase (L1,L2,L3)	% error in Load Voltage Magnitude, PSCAD values are 100%, 3 significant figures (Mean,Maximum)	% error in Calculated Apparent Power Magnitude, Input values are 100%, 3 significant figures (Mean,Maximum)	Number of iterations
3500, 2800, 4200	0.0116,0.0277	0.0831,0.124	9
7000, 5600, 8400	0.0522,0.180	0.365,0.585	14
10500, 8400, 12600	0.19,0.695	0.925,1.67	24
14000, 11200, 16800	5.01,19.0	2.07,4.87	94
14300, 11440, 17160	Did not converge		

Table 3.5 - Comparison between PSCAD (fixed power loads) and NumPy (fixed power loads) for increasing unbalanced load. Loads on the 2nd phase (L2) were set to 80% of the loads on the first (L1) and loads on the third phase (L3) were set to 120% of the loads on L1

3.7 The equivalence of TNS and TN-C-S earthing arrangements

The TN-C-S (or Protective Multiple Earth – PME) and TNS earthing arrangements (see Figure 3.17) are equivalent in normal operation. This is because the impedance of the neutral is small compared to the earth path(s) – earth connections of the neutral along the main trunk conductor can therefore be ignored. Hence, TNS is used in the Python model. Proving the equivalence of TN-C-S and TNS is important because it is the policy of many Distribution Network Operators to use TN-C-S where practical [83][84][81]. Note that Ingram et al [185] do not state the earthing configuration of the LV feeder.

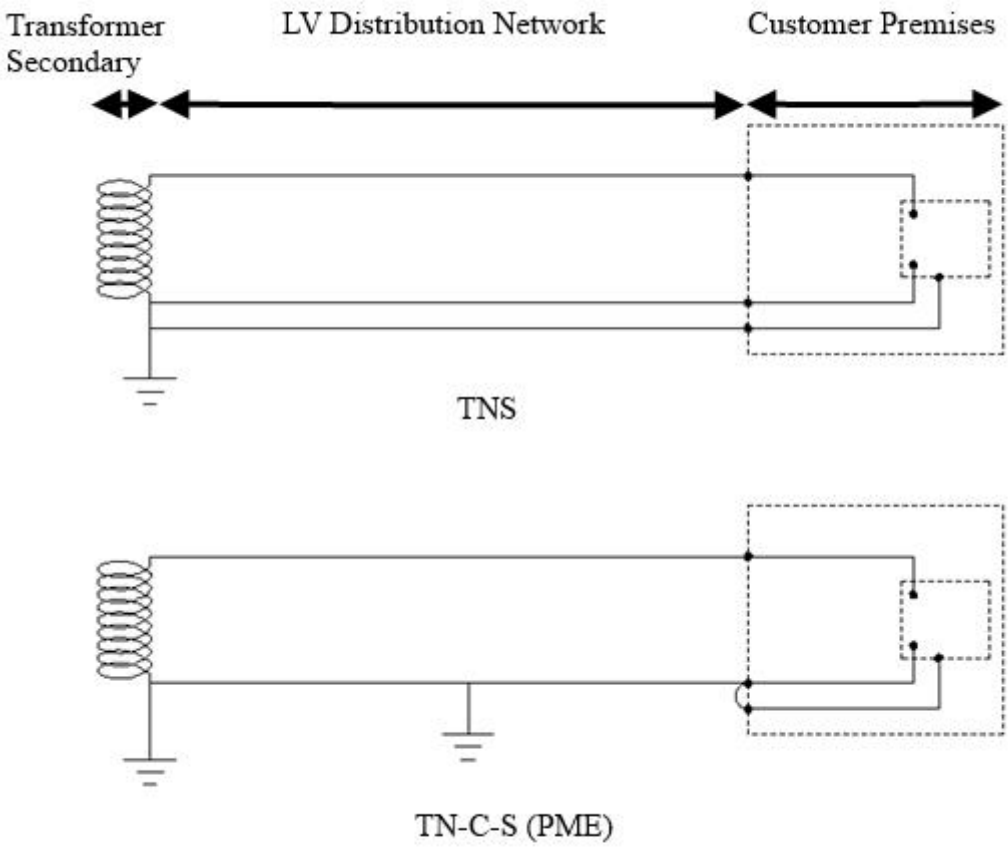


Figure 3.17 - TNS and TN-C-S (PME) earthing arrangements.

To demonstrate the equivalence of TNS and TN-C-S in normal operation, 3Ω earth electrodes were added along the neutral conductor in the PSCAD/EMTDC model. The results from Grcev and Popov [216] imply that, at 50Hz, with a 3m long copper earth rod and a soil resistivity of 10Ωm (a relatively low value), an earth resistance of approximately 3Ω can be expected. The resulting errors are recorded in Table 3.6 and the voltage profiles are compared in Figure 3.18. A maximum error of approximately 0.1% occurs when earth electrodes are added at every star point (star point here means the junction point where three neutral service cables meet the main trunk service cable. An earth rod at every star point represents far more earthing than would be expected in reality – it is modelled here to illustrate the equivalence of TN-C-S and TNS). Practical systems have fewer earth electrodes and actual earth resistances are likely to be higher than 3Ω, owing to soil type and electrode length; the UK Power Networks Earthing guide [81] implies that a single earth electrode at the end (the end of the cable furthest from the substation) of the neutral cable, with a resistance less than 100 Ω, would be the minimum requirement for the LV feeder used by Ingram et al. This means that the maximum

difference in load voltage magnitude between TNS and TN-C-S will be lower than that shown in case 1, 0.02%. This error is not significant; therefore TNS and TN-C-S are equivalent in normal operation.

Case	TN-C-S Earthing configuration	Maximum % difference in voltage magnitude from TNS configuration
1	3 Ohm resistor at load 20 (the end load)	0.0198
2	3 Ohm resistors at loads 10 and 20	0.0212
3	3 Ohm resistors at loads 2,4,6,8,10,12,14,16,18,20	0.0615
4	3 Ohm resistors at all loads	0.109

Table 3.6 - Comparison of TNS and TN-C-S (PME) earthing arrangements for an unbalanced, 60 load, LV feeder model.

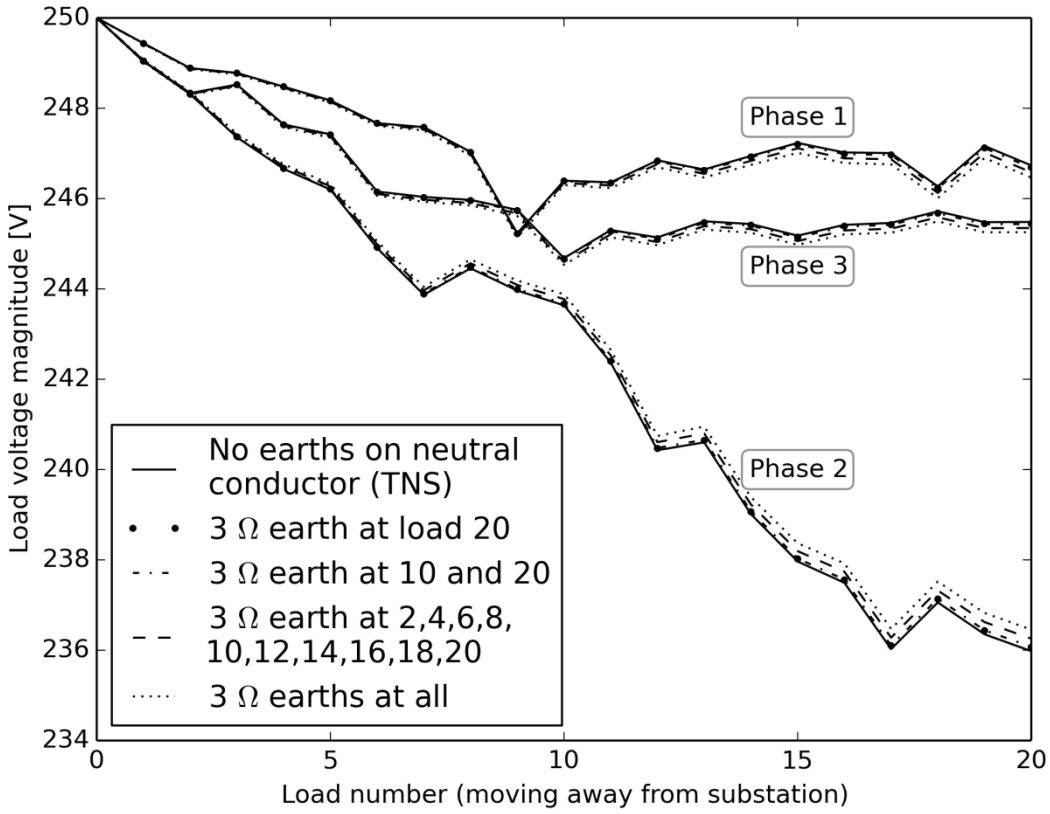


Figure 3.18 - Comparison of TNS and TN-C-S (PME) earthing arrangements for an unbalanced, 60 load, LV feeder model..

3.8 A lumped feeder model

The validated NumPy program was run for all 100 possible phase configurations for a 2 segment LV feeder. Note that each segment has 10 possible arrangements (see Figure 3.3). The number of possible phase configurations is the number of possible combinations of the possible segment arrangements. This is 10 raised to the power of the number of segments, in this case 2, giving 10^2 (10 possible arrangements of each segment, Figure 3.3, with 2 segments, Figure 3.17, leads to 10^2 possible configurations). The input values for the 2 segment feeder are lumped values based on Ingram et al's model. As such, each single phase lumped load represents 16 households. The input values used are shown in Figure 3.19 and Table 3.7

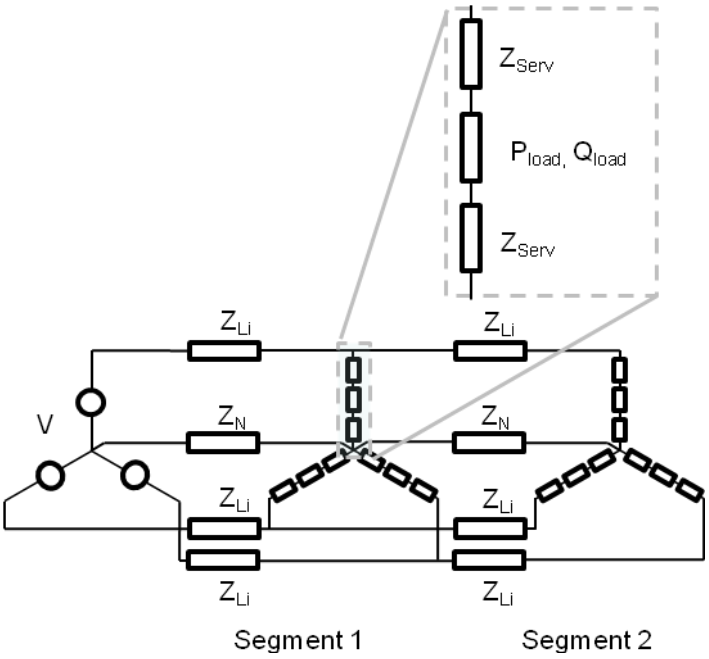


Figure 3.19 - Two segment lumped feeder

Symbol	Description	Value
Z_{Li1}	Segment 1 Line Impedances	0.0246 + j0.0111 Ω
Z_{N1}	Segment 1 Neutral Impedance	0.0246 + j0.0111 Ω
Z_{Li2}	Segment 2 Line Impedances	0.048 + j0.01125 Ω
Z_{N2}	Segment 2 Neutral Impedance	0.048 + j0.0024 Ω
Z_{Serv}	Service Impedances	0.0255 + j0.00123 Ω
P_{meter}	Lumped Metered Real Power, per segment, per phase	16kW
Q_{meter}	Metered Apparent Power, per segment, per phase	7.75 kVAr (3 significant figures)
V	Source voltages, line-ground	250, -125-j216.5,-125+j216.5 V

Table 3.7 - Input data for the 2 segment lumped feeder model

The highest proportion of loads connected to a phase, abbreviated to *HiProp*, was selected as an indicator of connection imbalance. If both segments have a phase configuration of [2,1,0], that is two loads connected to phase 1, one load connected to phase 2 and no loads connected to phase 3, the highest proportion of loads connected to a phase is 66.7 %. This is found by counting the number of loads connected to each phase and then dividing by the total number of loads on the feeder. In the previous example this is 4/6.

Generally, as might be expected, the minimum (i.e. the lowest of all the metered voltages) load voltage seen on the feeder decreases as the proportion of load connected to a phase increases. The highest voltage drop is seen when all loads are connected to a single phase. This can be seen in Figure 3.20. Notice that, whilst there are 100 possible phase configurations, there are only 21 distinct points on the graph. This is because many of the points lie on top of one another. Therefore each visible point represents one, or more, distinct phase configurations. By way of example, some of these have been annotated on Figure 3.20 (segment 1 top, segment 2 below - see Figure 3.3).

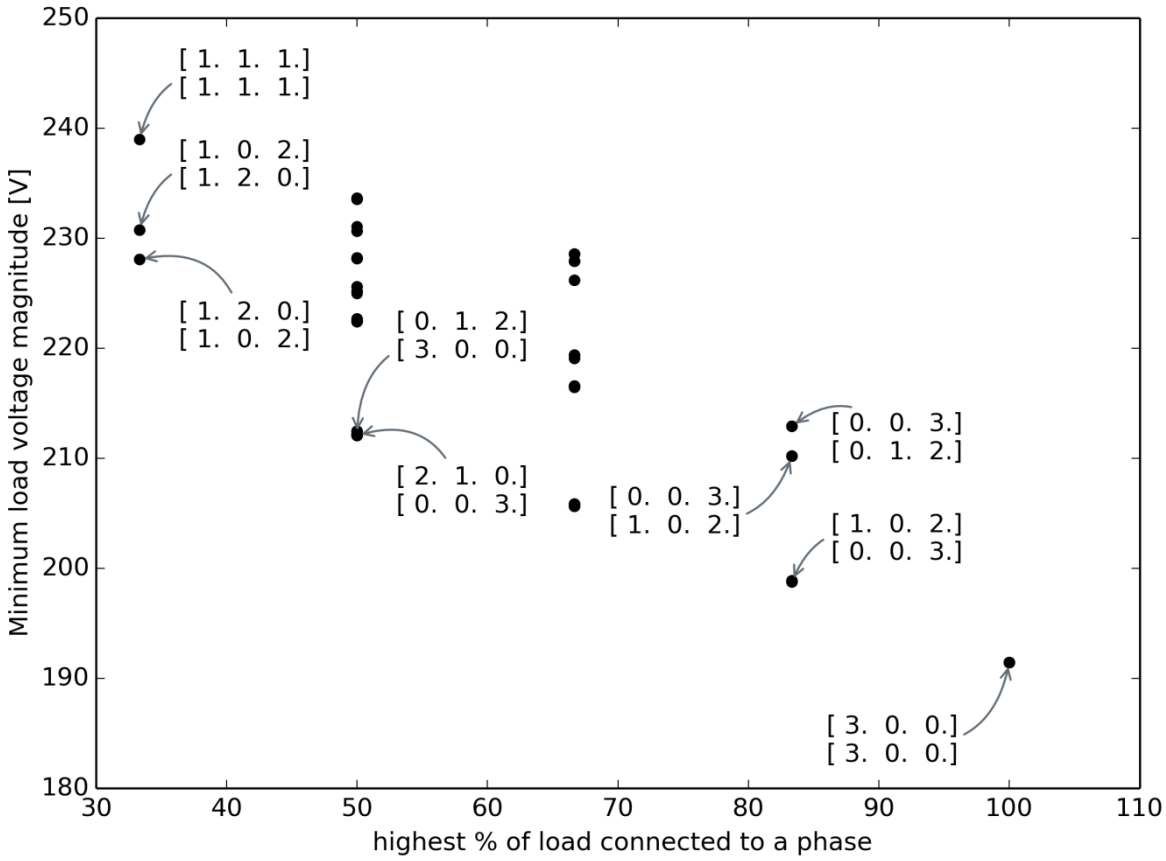


Figure 3.20 - Minimum load voltages vs connection imbalance, *HiProp*, for all phase configurations of the 2 segment LV feeder model.

The line and neutral I^2R losses increase with connection imbalance. Figure 3.21 shows the calculated line losses for all 100 possible phase configurations in the two segment feeder. The maximum line and neutral loss of approximately 50kW, ~34% of the input real power, seen under single phase configuration, is artificially high because, in this simplistic lumped case, half of the total load on the feeder is at the end of the line - resulting in an artificially large voltage drop.

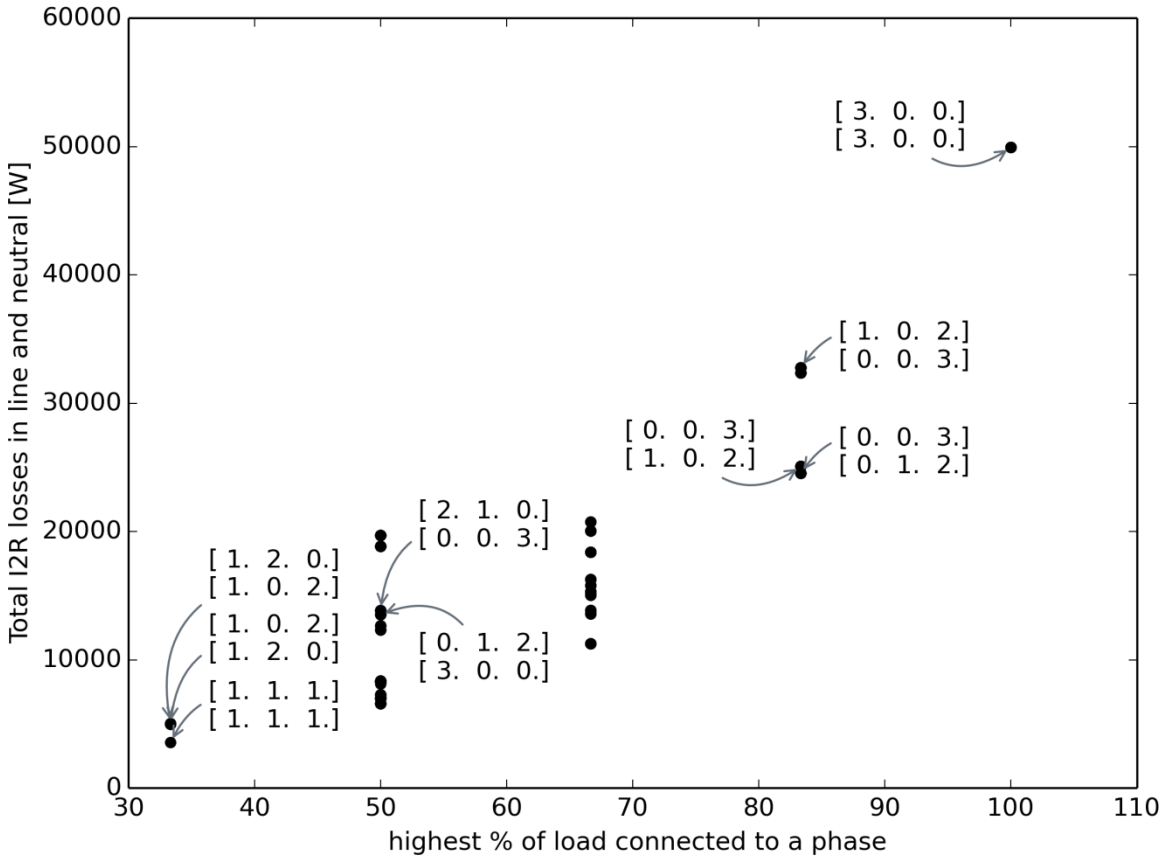


Figure 3.21 - Total losses vs connection imbalance, *HiProp*, for all phase configurations of the 2 segment LV feeder model.

3.9 Increasing the number of segments (the number of loads)

For the two segment feeder there are only five possible *HiProp* values; 2/6, 3/6, 4/6, 5/6 and 6/6. Each of 100 possible phase configurations falls into one of these five bins. If the number of segments is increased, the number of possible phase configurations and the number of bins will increase. For instance, with 32 segments, there are 10^{32} possible phase configurations and 65 bins. The previous two graphs show that, whilst there is significant variation of meter voltage and losses within each bin, observable correlation exists between *HiProp* and both meter voltage and losses. As the number of segments is increased, the same overall patterns in meter voltages and losses, are observed. To illustrate this, Figure 3.22 shows the minimum voltages for all of the possible phase configurations for 2, 3, 4 and 5 segment feeders.

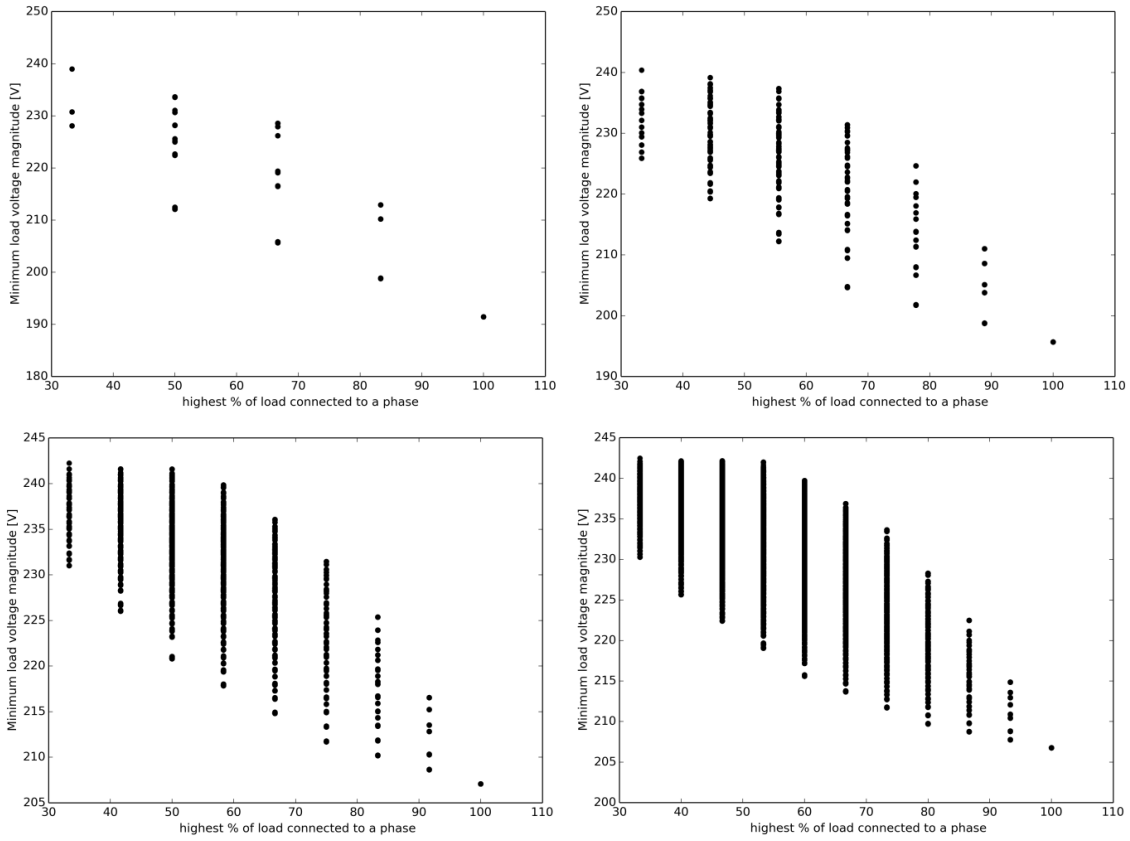


Figure 3.22 - Minimum voltages for the possible phase configurations for 2 (upper left), 3 (upper right), 4 (lower left) and 5 (lower right) segment feeders. Loads spread evenly along the feeder. Load values selected as lumped equivalent of 1kW per load for 32 segments, at 0.9 lagging power factor.

To match the feeder set out by Ingram, Probert and Jackson, the number of loads was scaled up to 96 and the line impedances adjusted accordingly (see Table 3.8 and Figure 3.23). Note the service cable phase conductor impedance was used for $\mathbf{Z}_{serv}$ – this represents a slight change to the neutral conductor resistance shown in Figure 3.1. The resulting program can be run many thousands of times given adequate computing resources. It is therefore well suited for use with stochastic techniques.

Character	Description	Value
Z_{Lia}	Segments 1-16 line impedance for 185mm^2 CNE cabling (all phase conductors) (9.375m)	0.00154 + j0.000694 Ω
Z_{Lib}	Segments 17-32 line impedance for 95mm^2 CNE cabling (all phase conductors) (9.375m)	0.003 + j0.000703 Ω
Z_{Na}	Segments 1-16 impedance of neutral for 185mm^2 CNE cabling (9.375m)	0.00154 + j0.000694 Ω
Z_{Nb}	Segments 17-32 impedance of neutral for 185mm^2 CNE cabling (9.375m)	0.003+ j0.00015 Ω
Z_{serv}	Impedance of service cable feeding each load (35mm2 CNE) (applies to all phases)(30m)	0.0255 + j0.00123 Ω

Table 3.8 - Segment impedances used for the 32 segment LV feeder model.

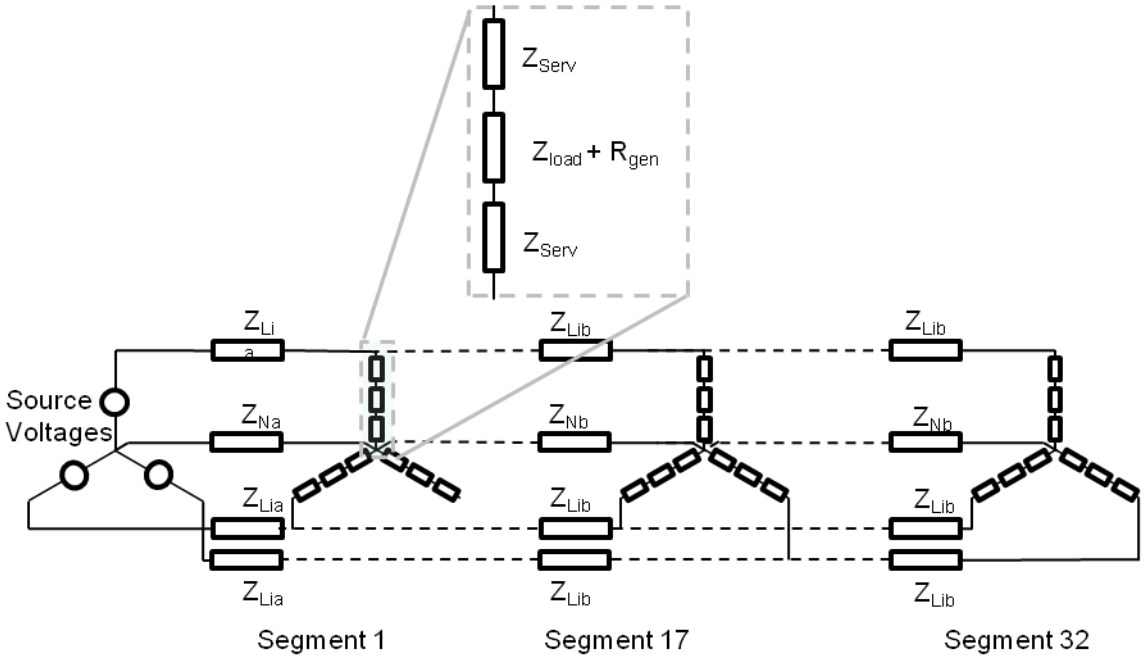


Figure 3.23 - The LV 32 segment feeder model (balanced configuration).

3.10 Discussion

A validated LV feeder model has been created. The model is suitable for use with stochastic demand models, allowing assessment of the influence of connection imbalance on the photovoltaic hosting capacity (Chapter 4) and the testing of Phase identification algorithms Chapter 5).

The work in this chapter has shown that modelling of a feeder with a T-NS earthing system is equivalent to a T-N-CS earthing system, in normal operation. This allows the model to be simpler than would otherwise be required.

This chapter has introduced a way of varying and quantifying imbalance in single phase connections. Connection imbalance is varied by reconfiguration of the arms within Y connected segments – represented by a three digit code. Connection imbalance is quantified using the maximum proportion of houses connected to a single phase on the feeder. Whilst this does not include information on the remaining two phases, it has been shown to be a measure of connection imbalance.

A consistent pattern in low voltage minima vs connection imbalance was observed, with an increasing number of houses connected to the feeder. This suggests that predictions can be made about the low voltage minima using only the single phase and balanced three phase cases.

4 The Influence of Connection Imbalance and Solar Photovoltaics on an LV Feeder

4.1 Introduction

The Low Voltage feeder model developed in Chapter 3 was combined with a stochastic demand model and a set of 1.3kWp clear-sky photovoltaic (PV) generation curves. Two hundred LV feeder cases were stochastically created with different configurations (different numbers of houses connected to each phase). Six thousand three hundred and thirty six combined demand (real and reactive power) and generation profiles were created, each with a 1-minute resolution. The profiles represent sets of 96 households, across six days in the year, with the amount of clear-sky PV generation increased in 11 steps. Based on this data, over 19 million load flow solutions were performed and the results analysed.

This work is timely because there is uncertainty around whether imbalance on British distribution networks is a significant problem. The aim of the work was to assess the severity of adverse effects (e.g. number of voltage excursions) related to imbalance in combination with trends in generation (e.g. the increase in photovoltaics).

4.2 Load and Generation Model

Stochastic demand modelling techniques were used to produce 576 demand profiles mimicking real variation in demand. These demand profiles were then combined with “clear-sky” photovoltaic (PV) generation profiles. The combined profiles were used as inputs to the unbalanced load flow technique developed in Chapter 3 to investigate the affect of connection imbalance and increasing penetration of PV on the LV feeder.

The demand and generation modelling technique was split into two parts. The first part was the creation of demand profiles, for weekdays within 6 different months, for 96 households. The second was the creation of “clear-sky” PV profiles and their subtraction from the demand profiles to produce combined demand and generation profiles. The modelling was based on the work of Richardson[217][180][218]. Their code was rewritten, in Numerical Python and adapted to be able to produce sets of 96 demand profiles at 1-minute resolution (See Appendix A4). Additional code as added to create a population of households with a distribution of household sizes (number of people resident) derived from UK census data [4], see Figure 4.1. Six sets of 96 demand (both real and reactive power) profiles were created, with a 1-minute resolution, representing weekdays in six months (January, March, May, July, September and November) across the year. The program structure, for creation of the demand profiles, is shown in Figure 4.2.

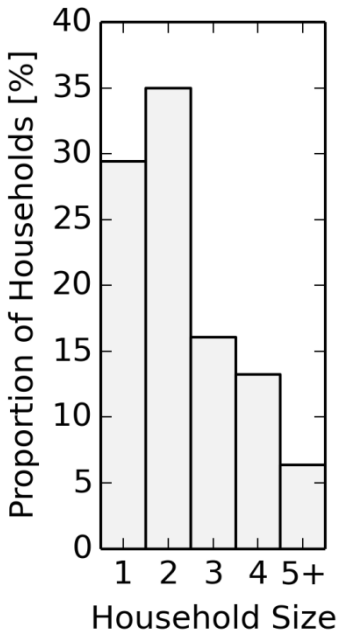


Figure 4.1 - Distribution of household sizes in the UK – based on data from Office National Statistics

In broad terms, Richardson’s technique works by generating random numbers for each minute of the day and using them to assess whether or not certain appliances turn on. However, if the simulation starts from minute 1 of 1440 minutes in a day, it cannot allow for, say, washing machines that may be continuing from the previous 24 hours. Therefore the program was adjusted to start from -198 minutes, as the longest running appliance in the simulation runs for 198 minutes. The code is shown in Appendix A4.

Six “clear sky” irradiance profiles (Wm^{-2}), representing alternate months (January, March, May, July, September and November), were produced, see Figure 4.3. These were used to create six PV generation profiles (real power only) by multiplying the calculated clear-sky irradiance on the panel (Wm^{-2}) with the panel size (m^{-2}) and system efficiency (output real electrical power/(irradiance x panel area)) at each time step. It is assumed that each panel is $10m^2$ with a system efficiency of 10% (based on assumptions of [219]). The chosen latitude was 51°48’N (just north of Merthyr Tydfil, Oxford and Chelmsford). The set of assumptions used are shown in Figure 4.4.

The clear-sky PV profiles were used to test the feeder under increasing penetration of PV. These represent a maximum value of PV export for the chosen PV installation. However, in reality, cloud and atmospheric changes mean that these smooth profiles are unlikely to occur. To illustrate this, a more realistic profile is shown in Figure 4.5. These spiky profiles are less repeatable and therefore less useful for use in comparison of PV hosting capacity. Hence the “clear-sky” profiles were used. Where the term “PV penetration” is used, it refers to the proportion of households with $10m^2$, south facing photovoltaic panels at a 35° incline with a conversion efficiency of 10% on a cloudless day.

The next step was to determine which households have a PV generator. In each of the sets of 96 households, a proportion was selected to be assigned a PV generator. This proportion was increased, in 10% steps, from 0% to 100%. The process by which this was done is shown in Figure 4.6. For each of the six months, the houses selected to have PV had the associated clear-sky power profile subtracted from their real power demand profile obtained in Figure 4.2. The result was a set 6336 {96 (no. connected households on a feeder) x 6 (months) x 11 (0 to 100% of households having ‘clear-sky’ PV)} 24 hour combined generation and demand profiles, at a one minute resolution.

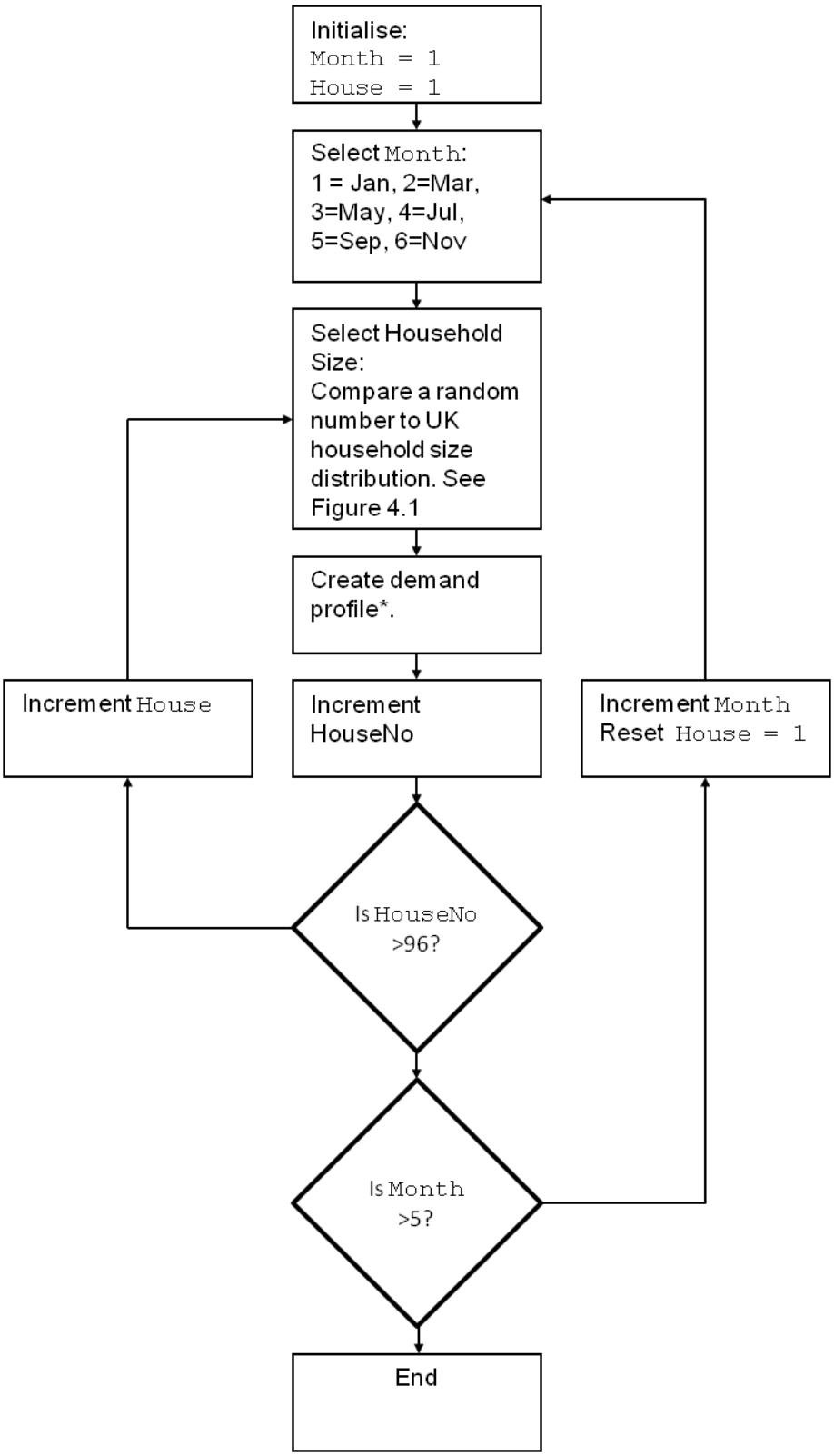


Figure 4.2 - Overview of demand profile creation method. *Using [180]**.**

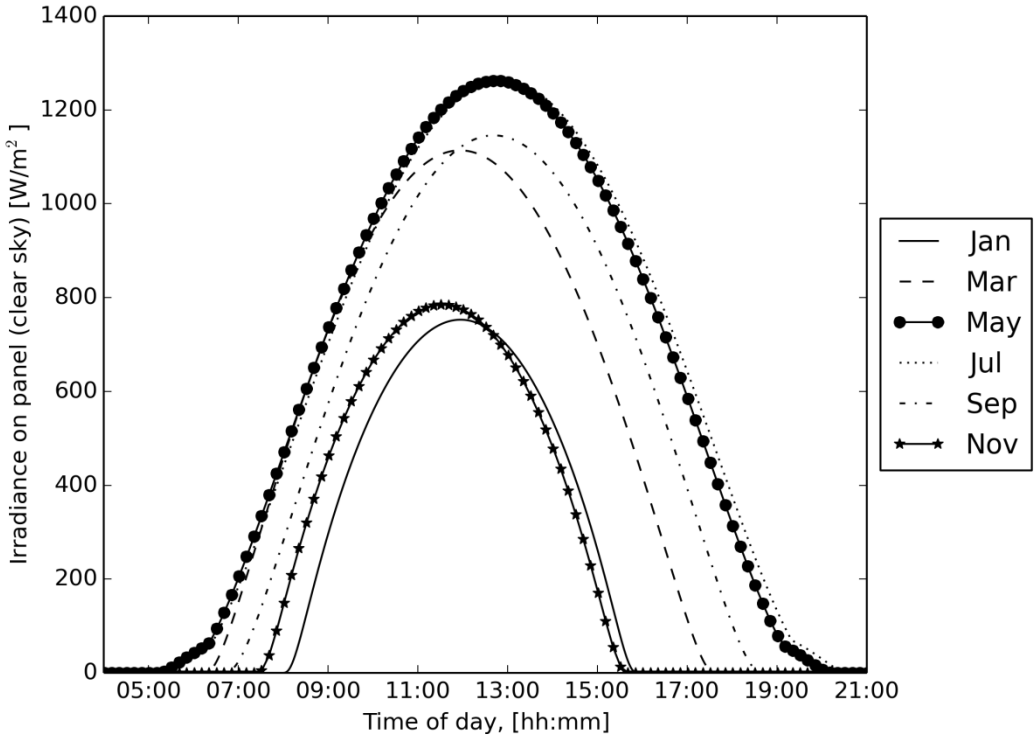


Figure 4.3 - Irradiance profiles with clear sky from Richardson's model [217] **for a selected day in each month (see Figure 4.2)**

```
System efficiency: 0.1
Panel Size: 10 m2

Day of the year   Jan   16
Day of the year   Mar   76
Day of the year   May   137
Day of the year   Jul   198
Day of the year   Sep   259
Day of the year   Nov   320

Latitude    51.48 degrees
Longitude   3.18 degrees

Day of the year that summer time starts   87
Day of the year that summer time ends     304

Slope of panel    35
Azimuth of panel  0

Ground reflectance      0.20
```

Figure 4.4- Assumptions used in creation of simplified monthly PV generation profiles

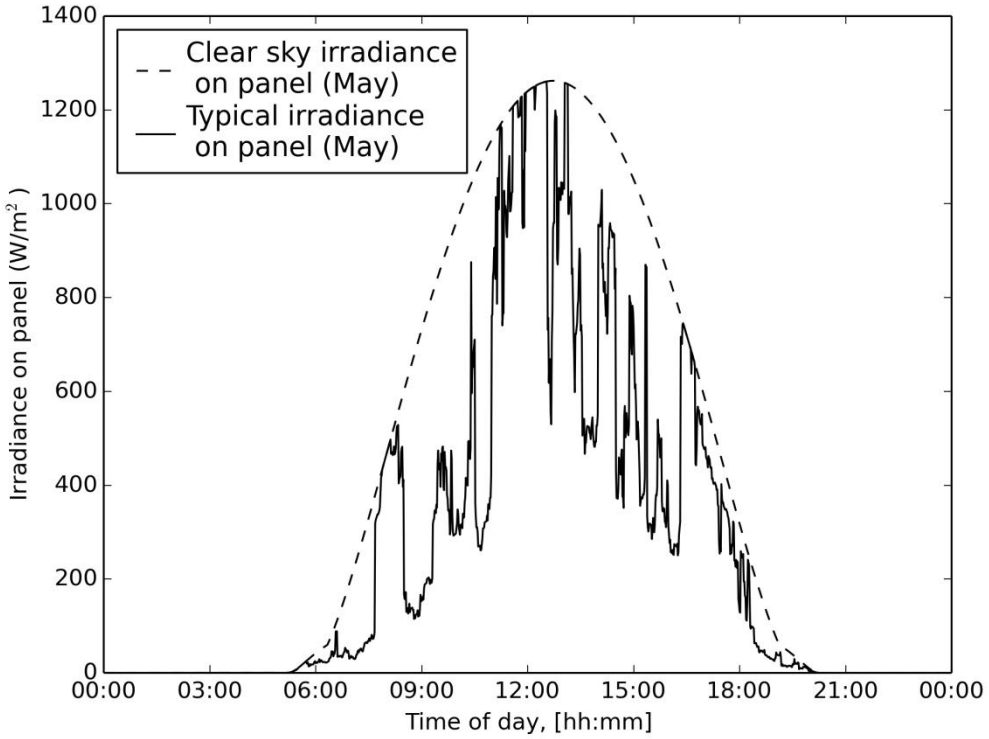


Figure 4.5 – Comparison of clear sky irradiance on panel and typical irradiance on panel (due to clouds). Outputted from Ian Richardson's Model

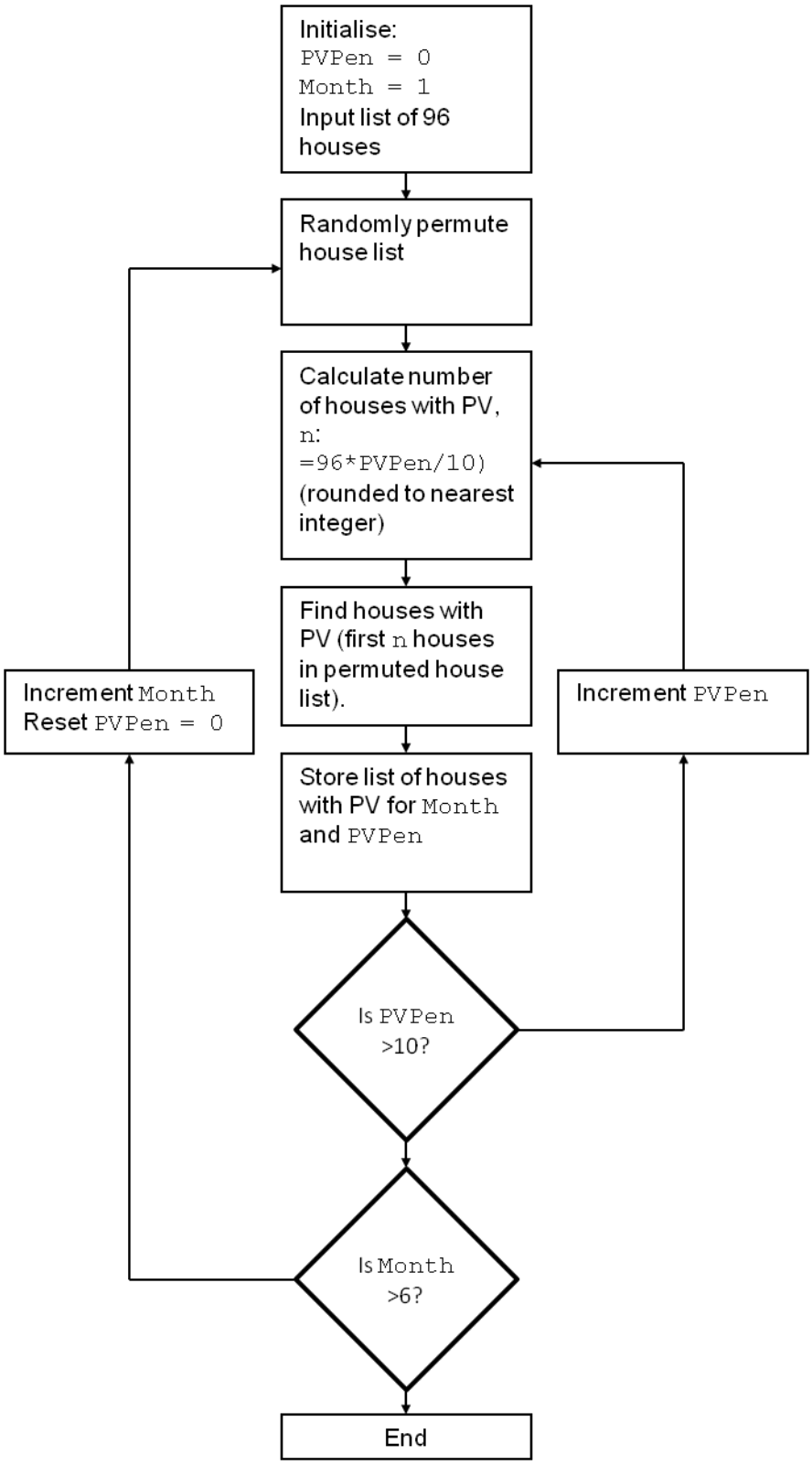


Figure 4.6 - Method for determining which houses have a PV generator. `PVPen` **is short hand for PV Penetration.**

4.3 Connection Imbalance

To examine the effect of increasing connection imbalance, 200 feeder configurations were created. The first two feeder configurations were selected as a balanced phase connection configuration (all segments having the [1,1,1] configuration) and a single phase configuration (all segments as [3,0,0]) . Figure 4.7 shows how the remaining 198 of feeder configurations were created using the Monte-Carlo technique. To do this, a “likelihood of connection to phase 1” number was used. For each household the question “to which phase is this household connected?” was asked. This was answered by comparison of a random number with the phase 1 connection likelihood. If a household was not on phase 1 it was given a 50% chance of connection to phase 2, else it was on phase 3. The process was repeated for 198 different feeders, with the random number threshold (representing the maximum number of houses connected to a phase) increased from 33.3% to 90% in 198 steps. The resulting distribution of the “proportion of households connected to phase 1”, for all 200 feeders, is shown in Figure 4.8.

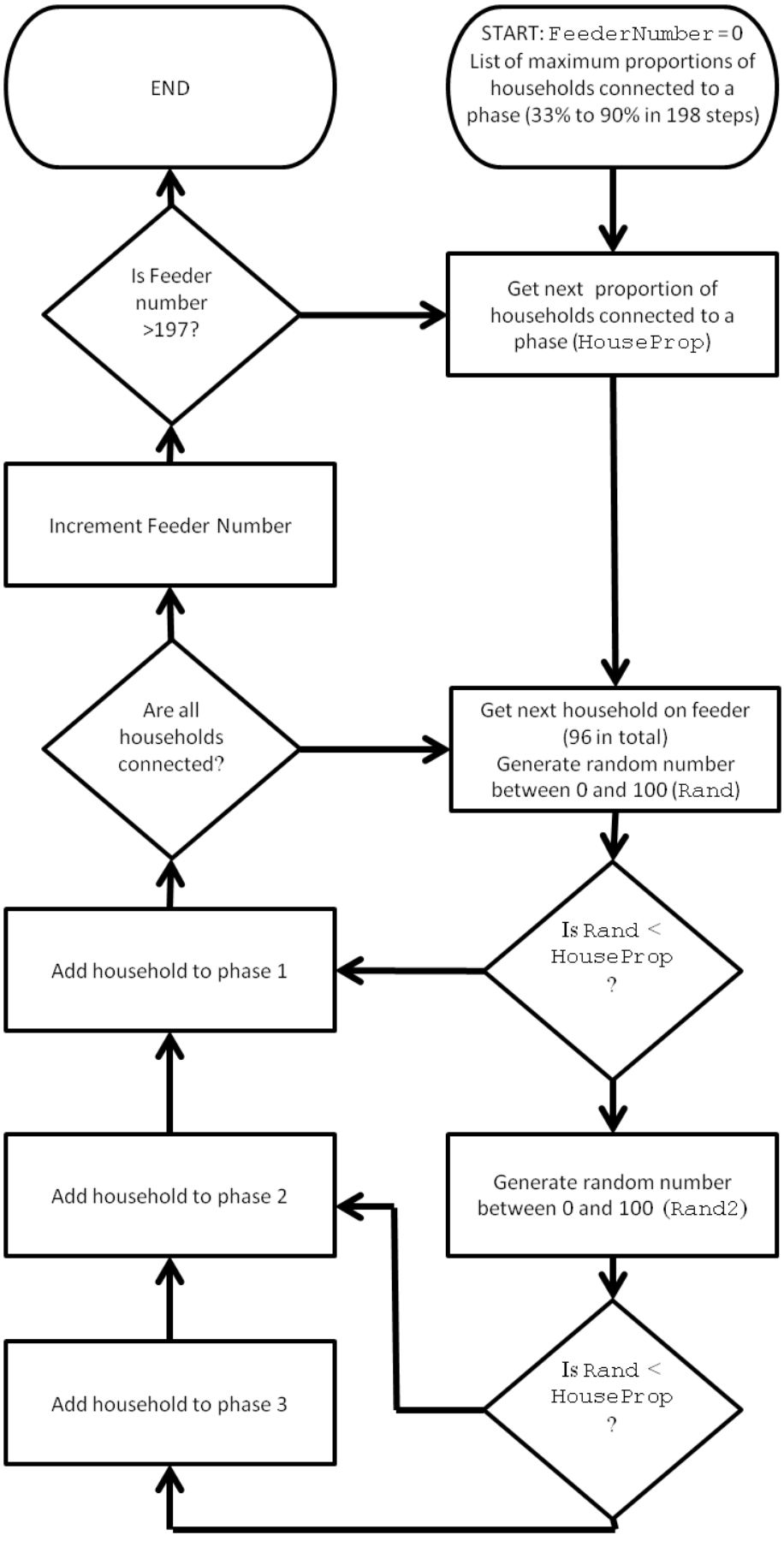


Figure 4.7 – Method for creation of 198 phase configurations with increasing proportion of houses connected to phase 1

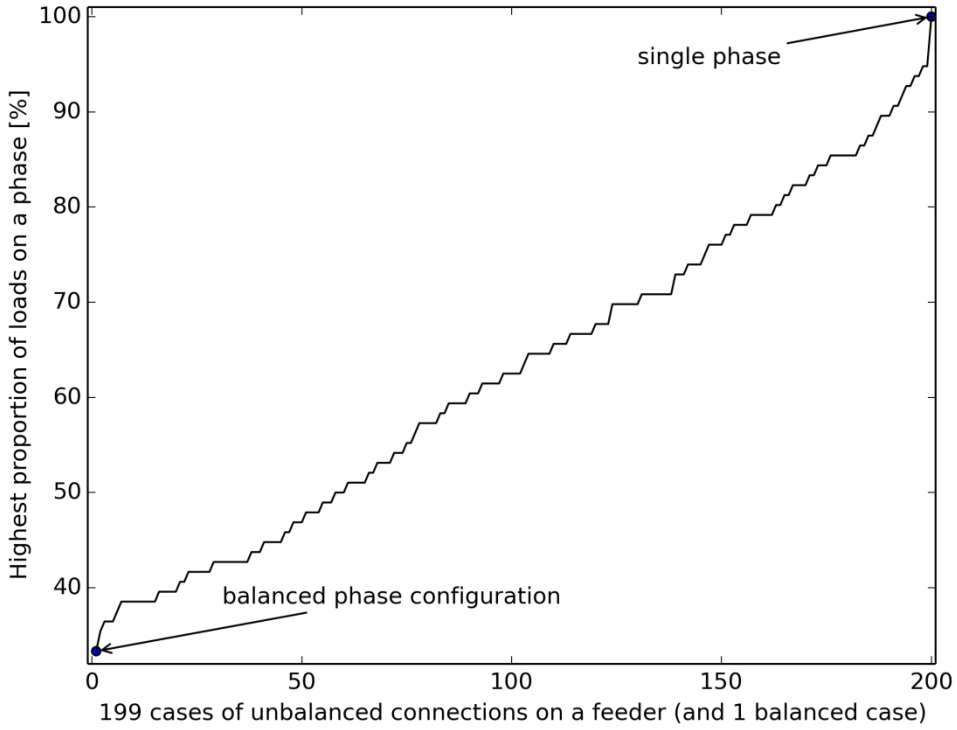


Figure 4.8 - Highest proportion of households connected to a phase for all 200 test feeders

4.4 Results - Meter voltage magnitudes

The 200 LV feeders were simulated 95040 times (6 months x 11 PV penetrations x 1440 minutes of the day), giving a total of 19.008 million unique sets of outputs. This resulted in a set of voltages at each distinct point on the LV feeder for every minute within the 6 selected days. From the results, maximum and minimum meter voltage magnitudes, maximum and minimum meter voltage times, supply currents and losses were calculated and examined. These maxima and minima can be thought of as annual values, where the annual range is approximated by the simulation of 6 days spread across the year. For the related code, see Appendix A4.

Figure 4.9 shows the maximum and minimum annual meter voltage magnitudes on every feeder for increasing levels of PV penetration. Each point on the graph represents either an annual maximum (those in the grouping towards the top of the graph) or an annual minimum (those in the grouping towards the bottom of the graph) meter voltage magnitude. Least squares polynomial lines of best fit are also shown on the graph. In Figure 4.10, for clarity, the samples are removed leaving only the lines of best fit.

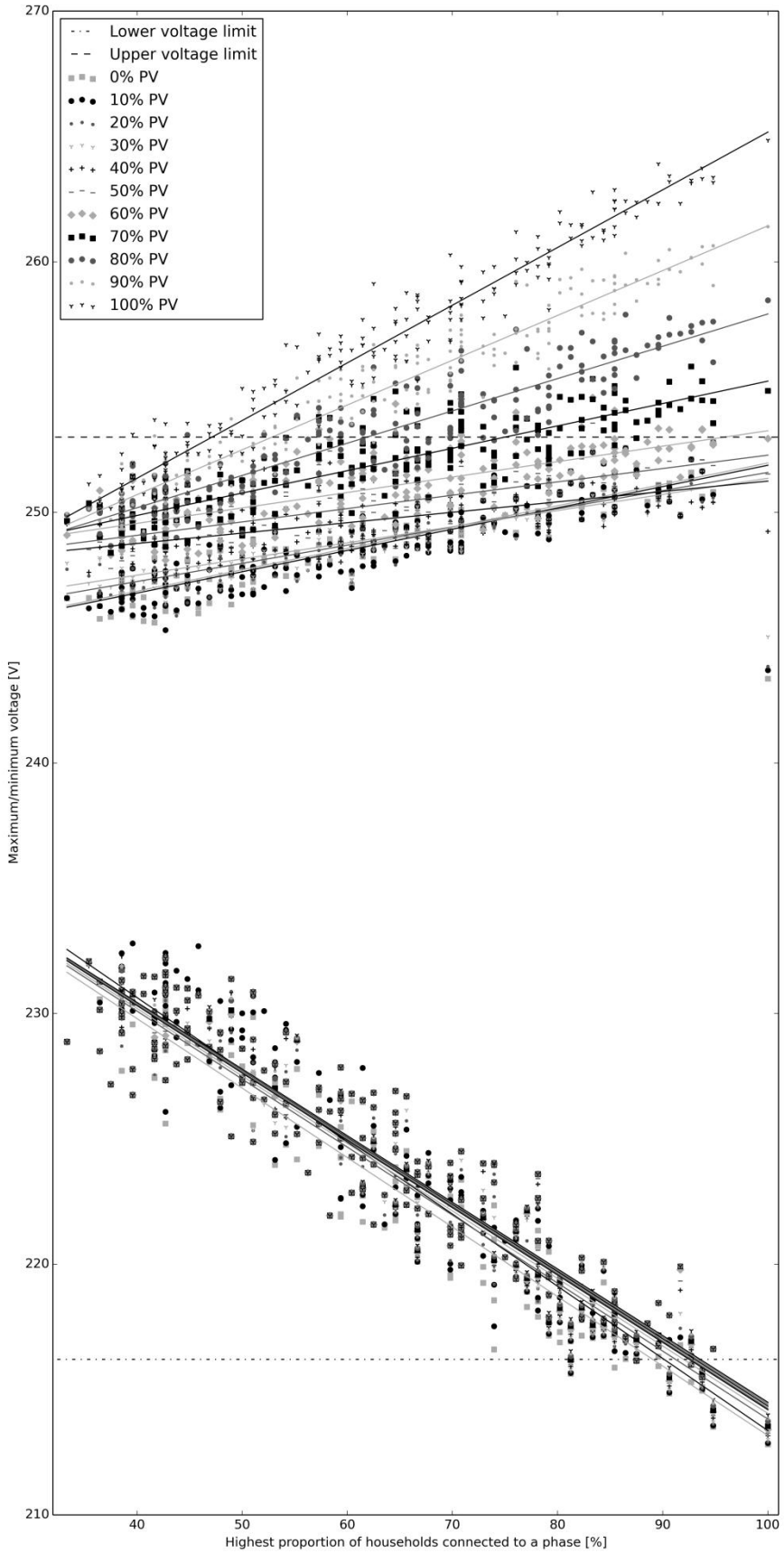


Figure 4.9 - Maximum and minimum voltage magnitudes on every feeder for different PV penetrations.

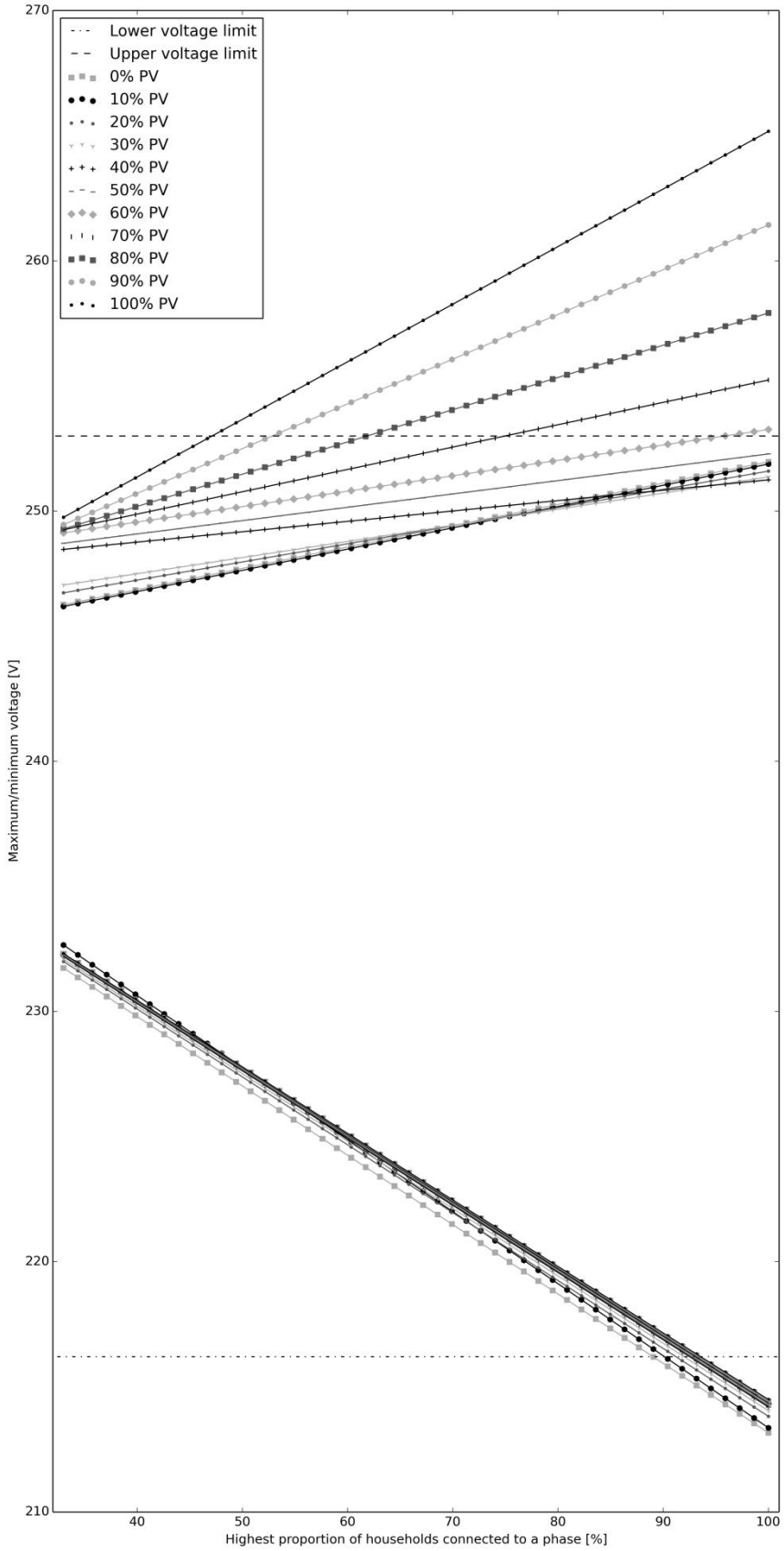


Figure 4.10 - Maximum and minimum voltage magnitudes on every feeder for different PV penetrations. Least squares polynomial lines of best fit.

In Figures 4.9 and 4.10, comparison of the upper (meter voltage maxima) and lower (meter voltage minima) groupings indicates that the minimum meter voltage is relatively independent of the amount of connected generation. In the lower group, most of the points fall on top of one another and the lines of best fit are tightly packed when compared to the upper group. Figures 4.11 and 4.12 show the lower grouping in more detail. They show that the connection of PV does have slight influence on the minimum meter voltage seen on a given feeder; in general, the connection of PV tends to lift the minimum meter voltage level. The mean separation between the 0% and 100% PV best fit lines is 0.95V. The impact of PV is influenced by imbalance; based on the lines of best fit, the difference between the 0% PV minimum annual meter voltage and the 100% PV minimum annual meter voltage grows from 0.57V at balanced configuration to 1.3V at single phase configuration. This indicates a general trend. It must, however, be kept in mind that there is significant variation in the minimum voltages around the trend lines, as can be seen in Figure 4.11.

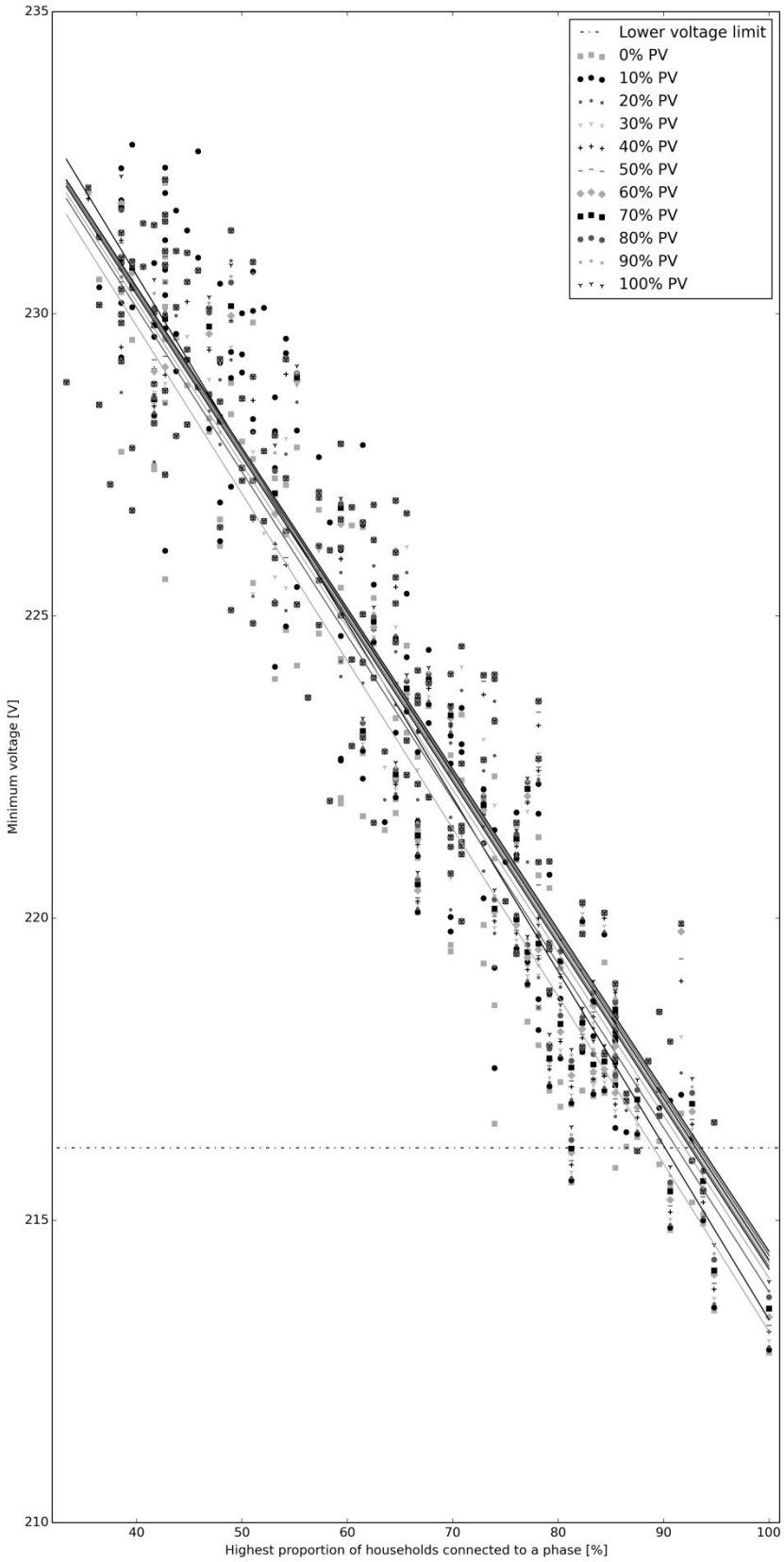


Figure 4.11 - Minimum voltage magnitudes on every feeder for different PV penetrations. With least squares polynomial lines of best fit.

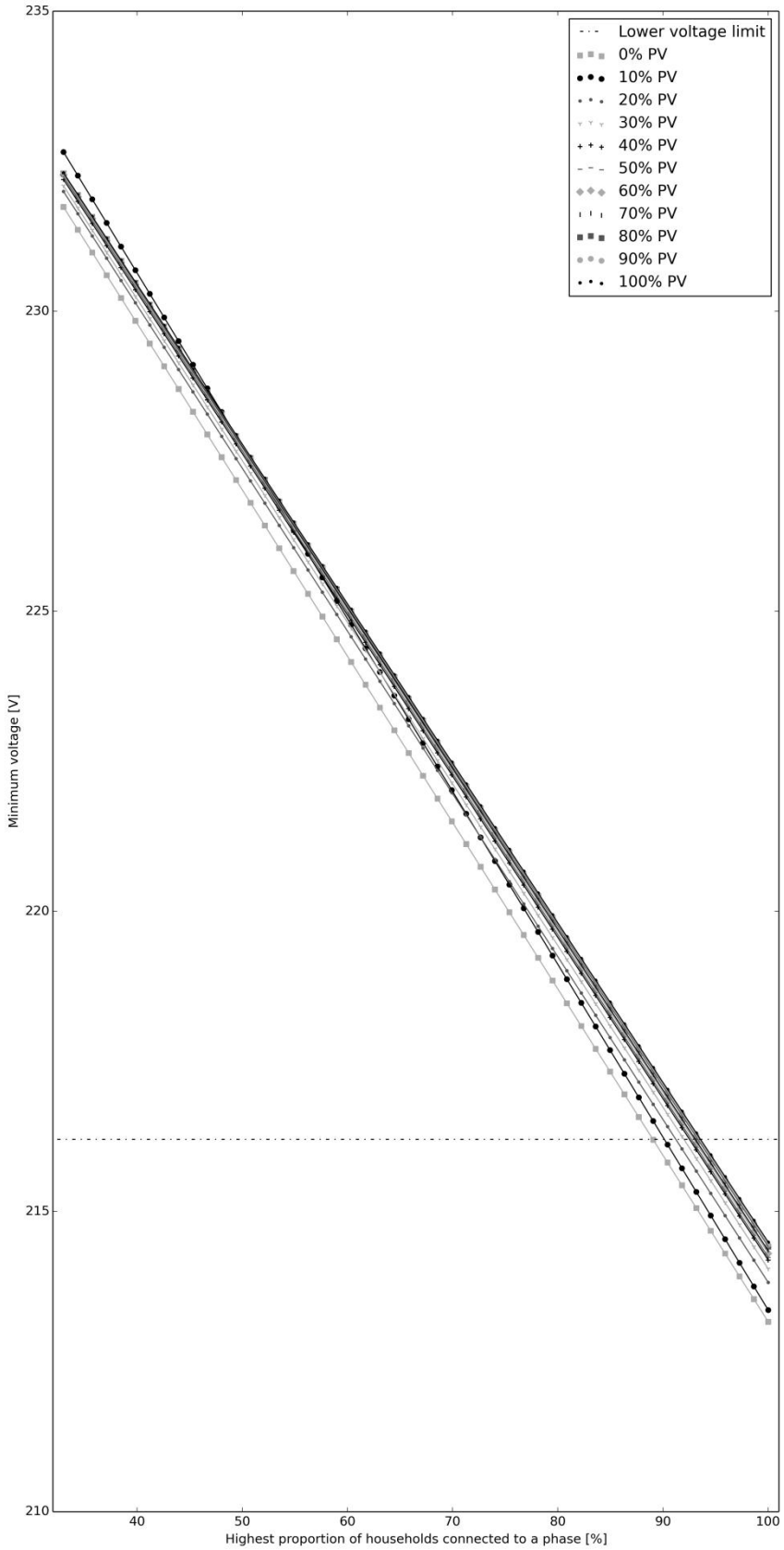


Figure 4.12 - Minimum voltage magnitudes on every feeder for different PV penetrations - least squares polynomial lines of best fit.

Inspection of the annual maximum meter voltages (Figures 4.13 and 4.14) reveals that increasing the penetration of PV tends to increase the annual maximum meter voltage. Furthermore, increasing the PV penetration increases the sensitivity of annual maximum meter voltage to increasing imbalance. As a higher proportion of households are connected to a phase, the maximum voltage tends to increase. Generally, the gradient of this increase gets steeper with increasing PV penetration, see Figure 4.14. There are two groups within the upper grouping. The first consists of the 0-30% lines – these lie on top of one another and no obvious effect occurs on increasing PV penetration within this range. At above 30% PV, however, the effect of increasing PV is clear. The lines from 40-100% PV fan out, with increasing gradients of the lines of best fit. Therefore, the increasing PV exacerbates the effect (increases annual maximum meter voltages) of increased imbalance.

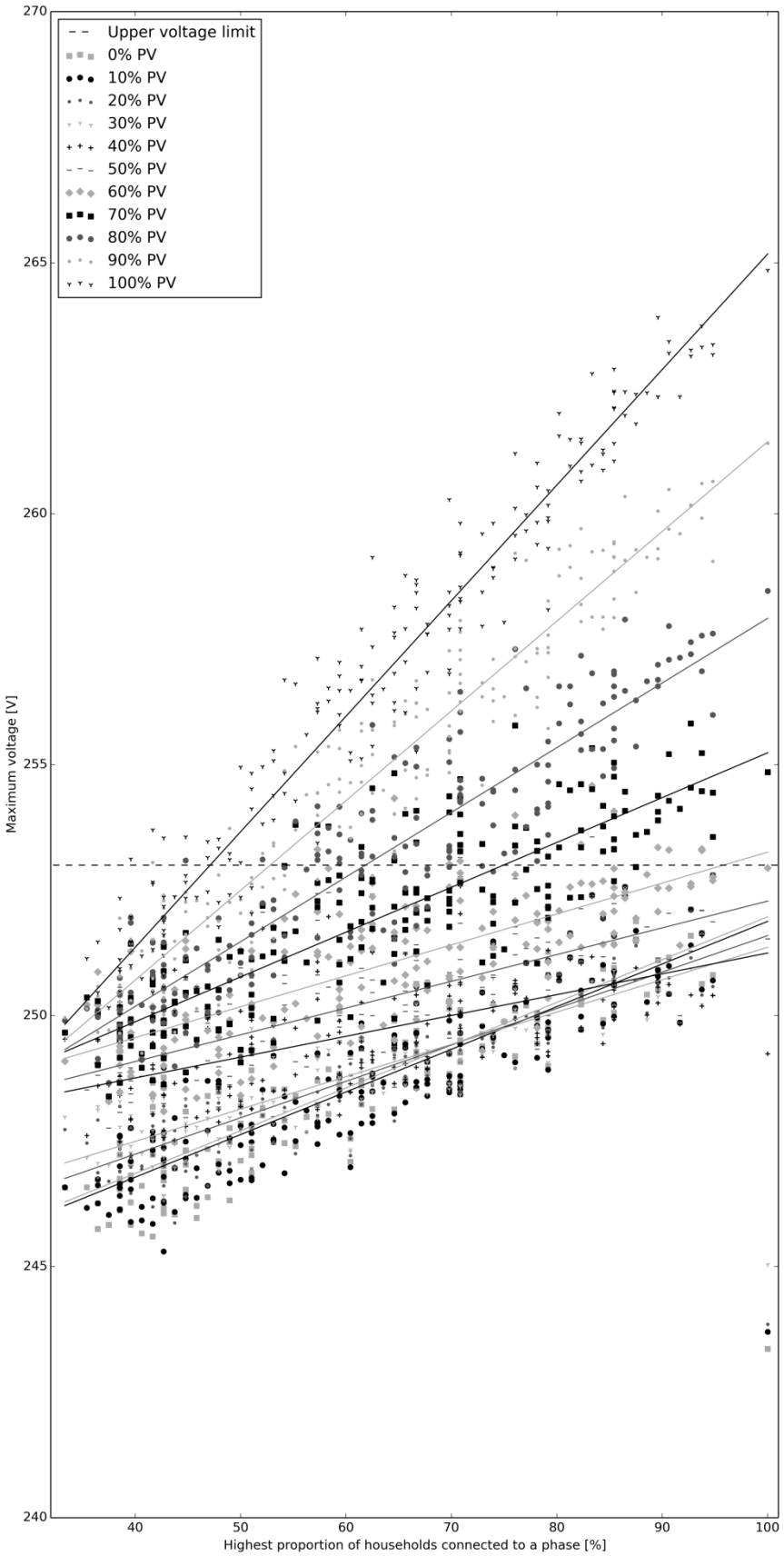


Figure 4.13 - Maximum voltage magnitudes on every feeder for different PV penetrations. With least squares polynomial lines of best fit.

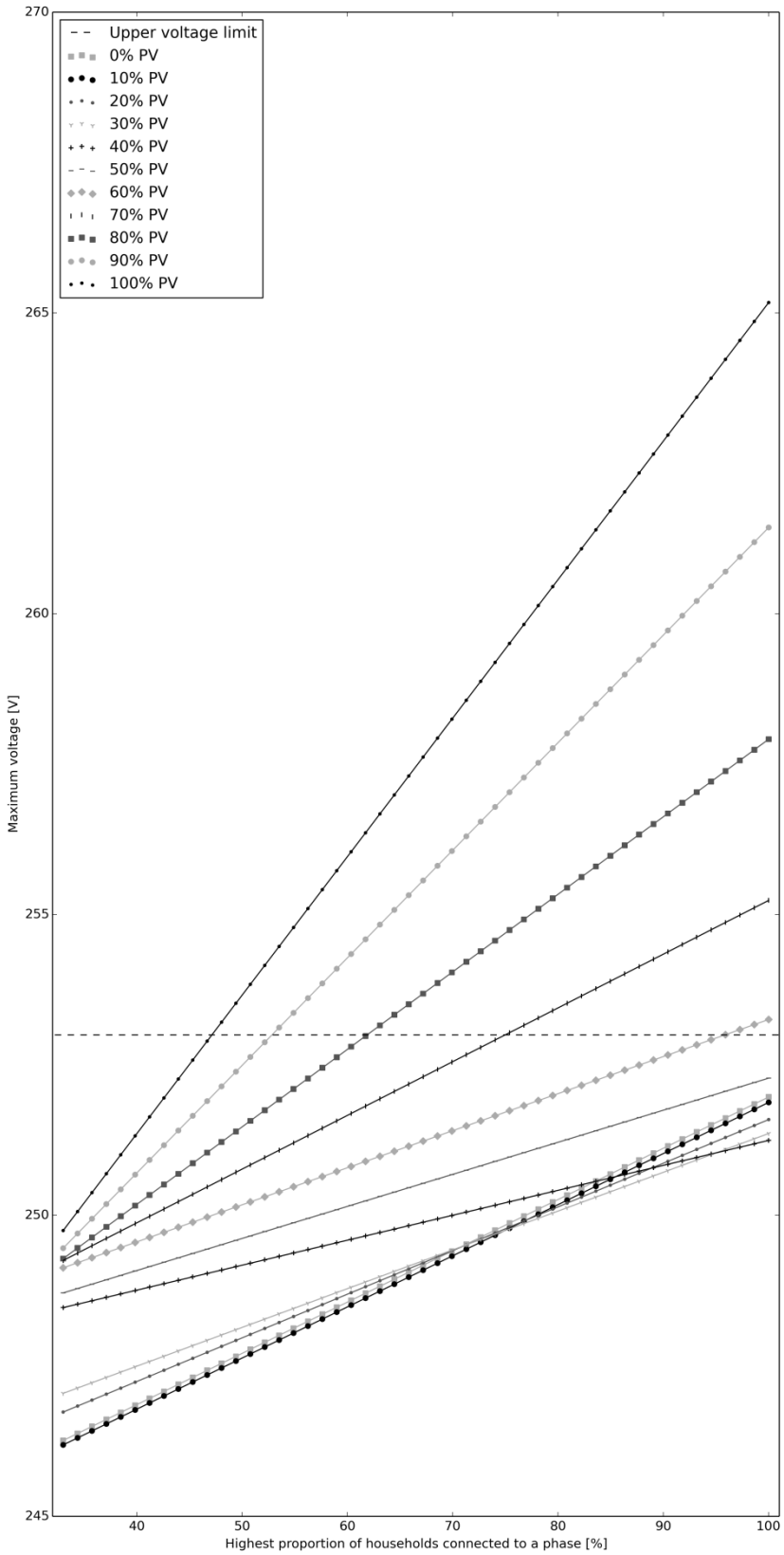


Figure 4.14 - Maximum voltage magnitudes on every feeder for different PV penetrations - least squares polynomial lines of best fit.

The range of voltages seen over a year, the annual minimum meter voltage subtracted from the annual maximum meter voltage, is a useful indicator of hosting capacity. If all of the meter voltages magnitudes seen on a feeder over a year fall within a 33.6375V range then no control measures need be taken – the entire voltage range can be accommodated by setting the transformer's off-load tap appropriately. In other words, there is at least one transformer tap setting that will ensure that all meter voltages remain within limits (therefore, considering only one feeder at a time, no additional active control would be required to prevent voltage excursions). The 33.6375V range is the 253V (230+10%) to 216.2V (230-6%) allowable range minus a 3.1625V allowance for the closest 1.25% (of 253V) transformer tap adjustment - see Appendix A3. This also assumes, however, that the transformer primary voltage is stable at 11kV.

Figures 4.15 and 4.16 show the annual meter voltage range seen on the 200 test feeders. It is the maximum meter voltages that have the greatest influence on the range. The similarities between Figures 4.13 and 4.14 and Figures 4.15 and 4.16 are clear. The maximum allowable proportion of households connected to a phase, for different PV penetrations, can be read from the graphs of Figures 4.15 and 4.16. The values based on both the lines of best fit and "the first sample outside the allowable range" are tabulated in Table 4.1 and graphed in Figure 4.17. Both lines follow the same general pattern; from 0 to 40% of households with PV, the allowable imbalance rises slightly. Above 40% however, the allowable imbalance drops. Based on the lines of best fit, all voltage changes are containable without control actions, providing the highest proportion of houses connected to any one phase remains lower than 66%.

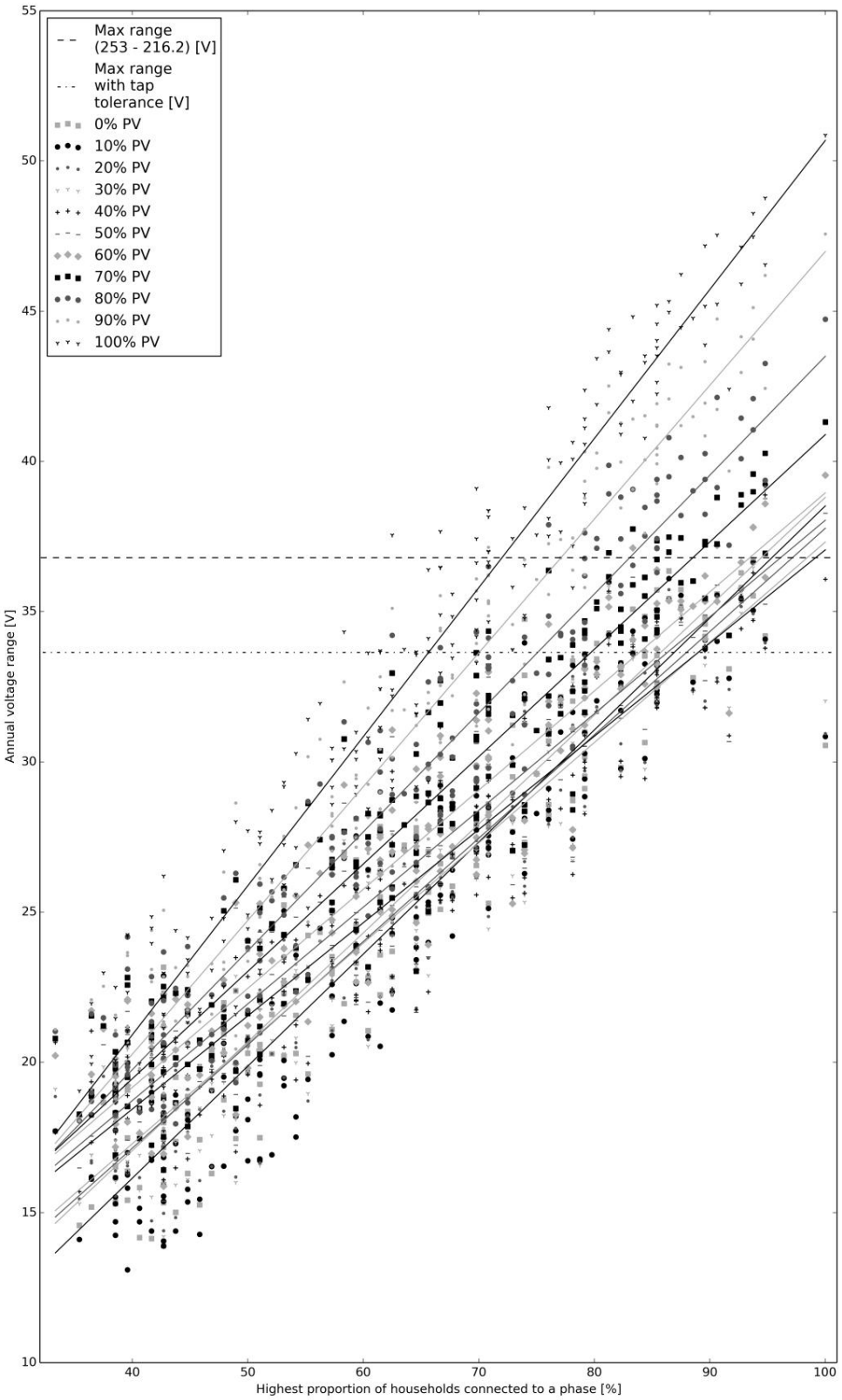


Figure 4.15 – Annual voltage range on every feeder for increasing PV penetrations with least squares polynomial lines of best fit.

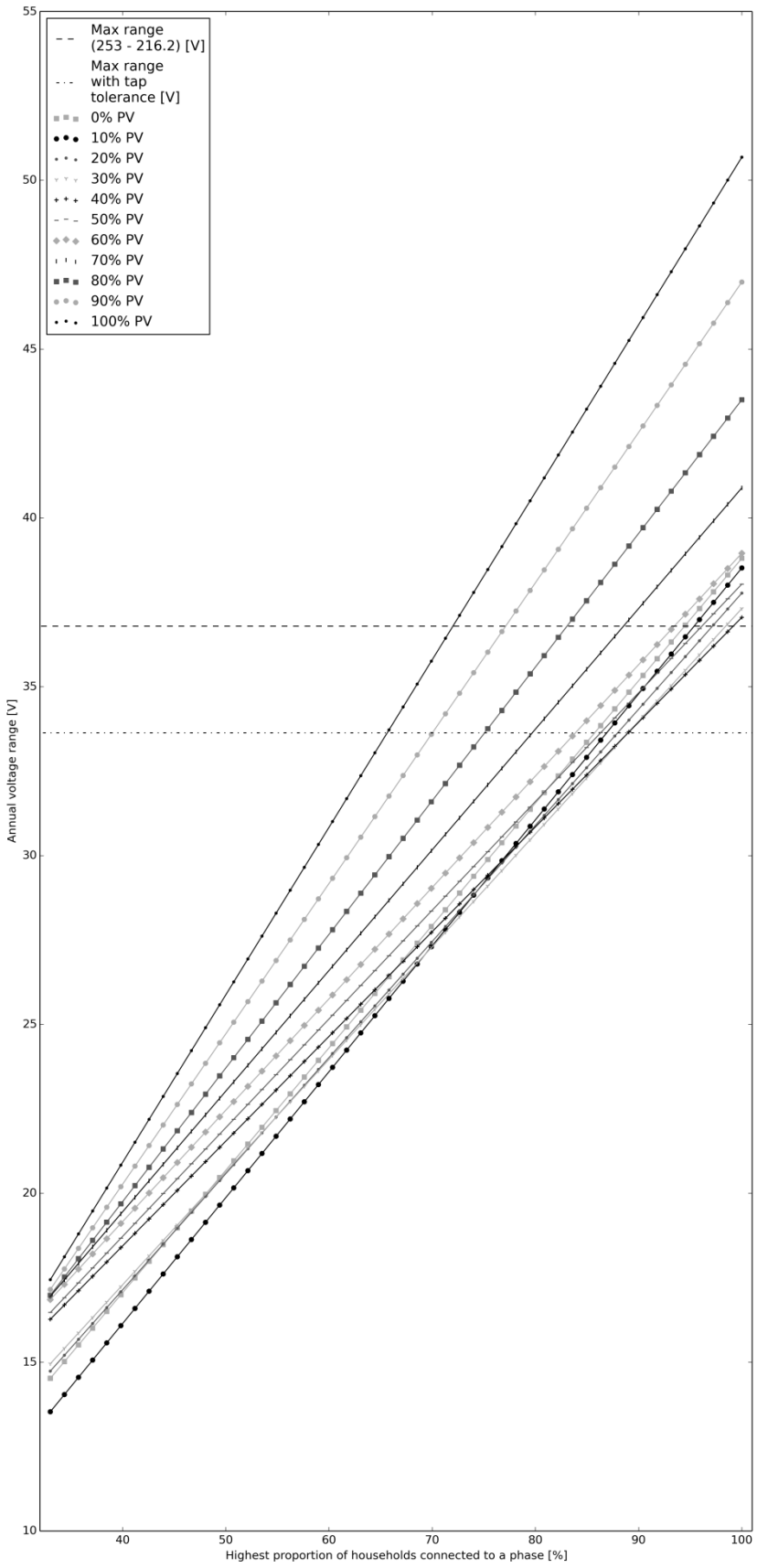


Figure 4.16 - Annual voltage range on every feeder for increasing PV penetrations – Least squares polynomial lines of best fit.

Proportion of households with ‘clear sky’ PV	Maximum proportion of households on a phase before voltage excursions will occur. (based on least squared polynomial line of best fit) [%]	Maximum proportion of households on a phase before voltage excursions will occur (based on first sample outside allowable voltage range).
100	66	58
90	70	63
80	75	63
70	80	71
60	84	76
50	86	81
40	89	81
30	89	81
20	88.0	81
10	86.9	74
0	85.7	74

Table 4.1- Maximum proportion of households connected to one phase, allowable without causing a voltage excursion, for varied 'clear-sky' PV penetrations.

Connection imbalance increases the likelihood of overvoltages caused by solar PV, especially where more than 40% of households have PV. However, for PV penetrations of less than 20%, imbalance is not a significant problem, in terms of LV voltage excursions, unless the proportion of households connected to any one phase exceeds 74%. With high penetrations of PV this limit drops (for example, to 58% if all households have PV).

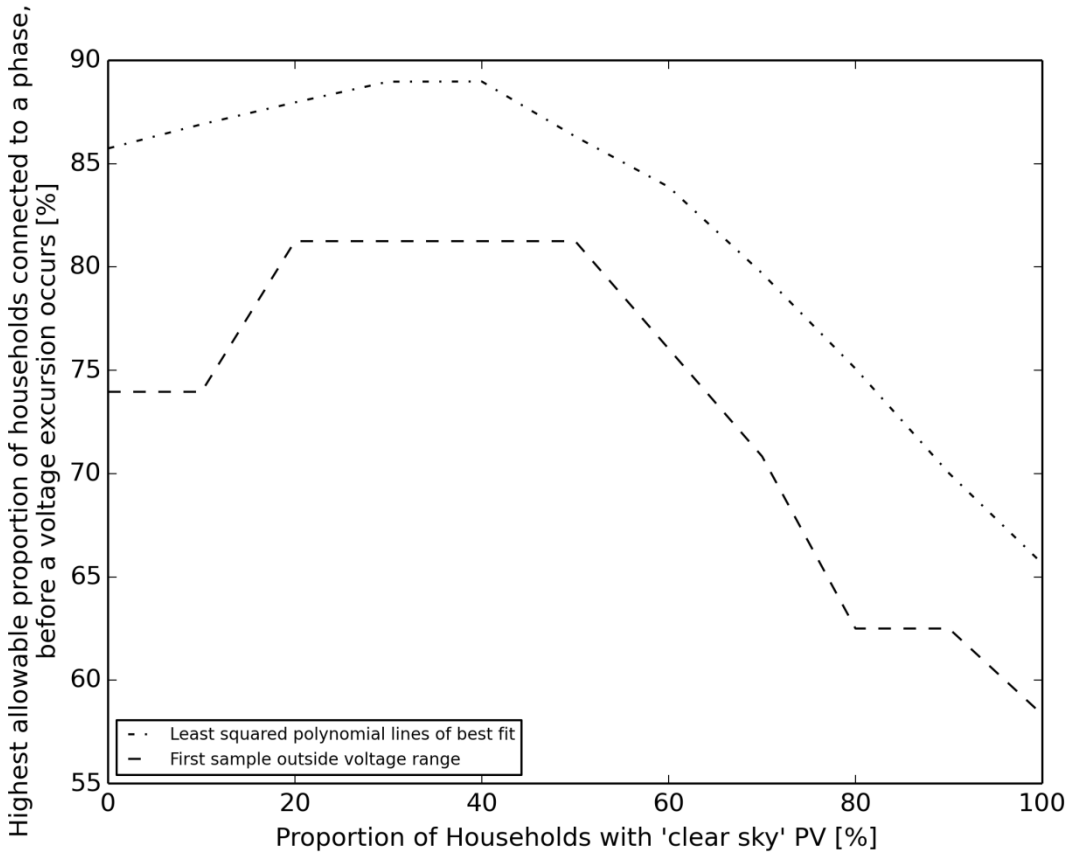


Figure 4.17 - Maximum proportion of households connected to one phase, allowable without causing a voltage excursion, for varied 'clear-sky' PV penetrations.

4.5 Results - The time of day that meter voltage maxima and minima occur

The time at which maximum and minimum voltage magnitudes occur for every feeder, for every month (of the 6 simulated) was determined. As one might expect, the maximum voltages generally occur during the middle of the day (e.g. at maximum PV output) and the minima generally occur during the evening (e.g. at maximum load). For each of the 6 simulated days and each of the 11 simulated PV penetrations, histograms were produced to visualise the time of day at which meter voltage maxima and minima are likely to occur. For both the maxima and the minima at each PV penetration, the 6 histograms were added together to give an annual overview, see Figure 4.18. Meter voltage minima are less likely to occur, from 0730hrs to 1600hrs, as the clear-sky PV penetration increases.

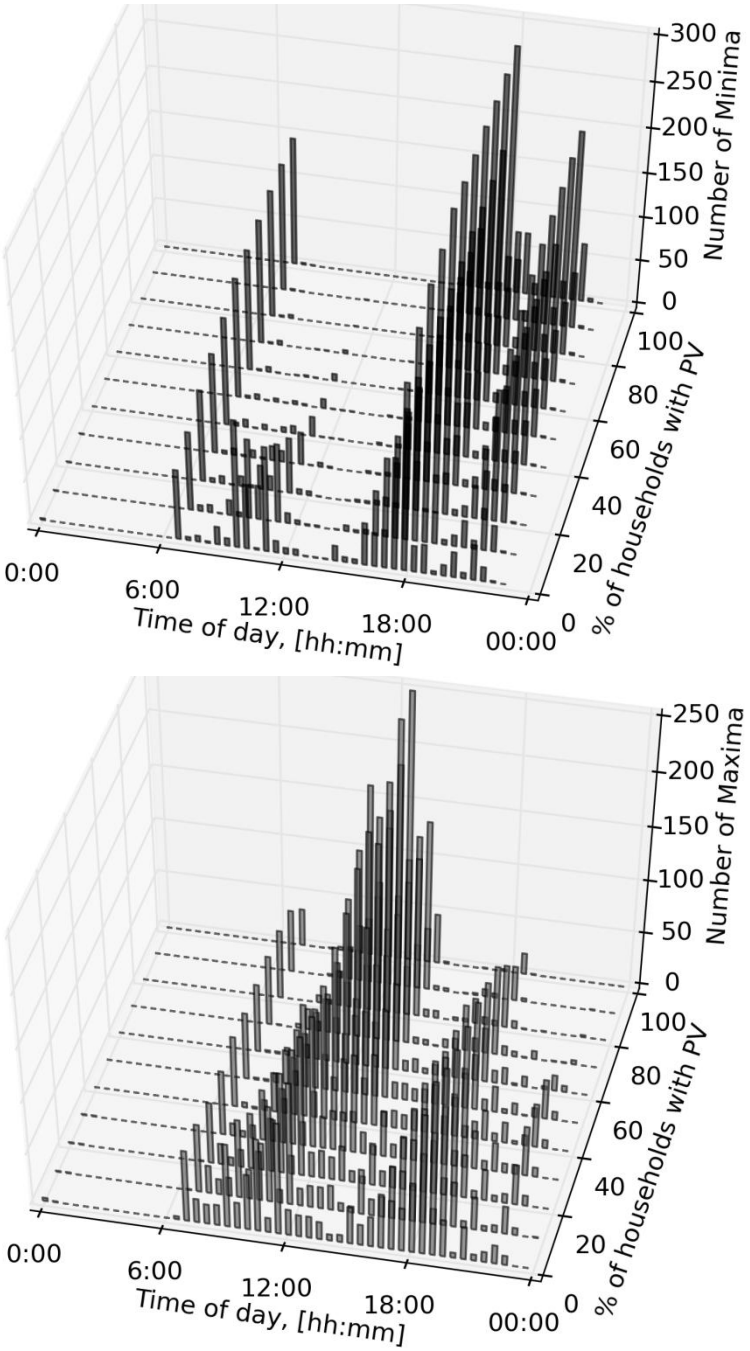


Figure 4.18 - The time of day that meter voltage maxima and minima occur for increasing PV penetrations, in 30 minute bins.

Figure 4.19 combines the histograms of Figure 4.18 into a single, 2 dimensional, histogram. From this it can be inferred that there is not complete temporal separation between the meter voltage maxima and minima. This means that there is possibility that adjacent feeders, fed from the same distribution transformer, may experience a meter voltage maximum and a minimum at the same time. If this occurs, and the voltage range is outside the 33.6375V allowable range, the scope for active tap change control is limited as it may be required to shift the tap setting

both up and down to compensate for simultaneous over and under voltages on adjacent feeders.

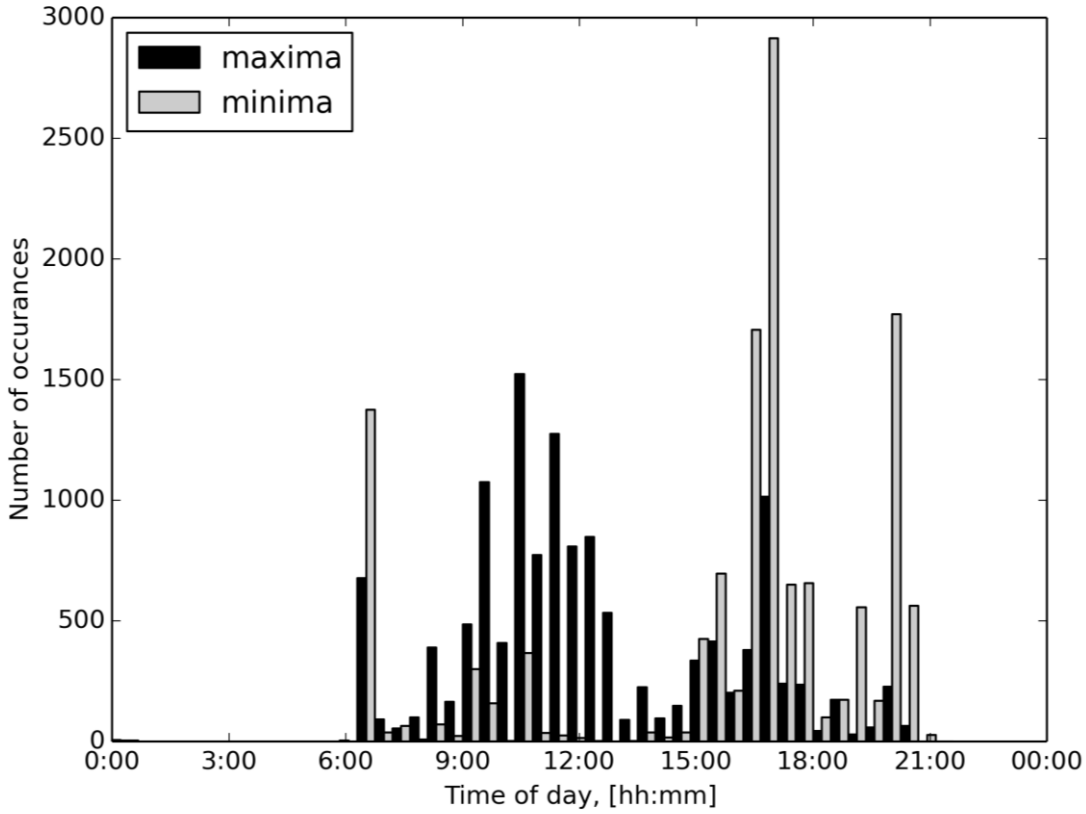


Figure 4.19 - Histogram of annual maximum and minimum meter voltages

4.6 Results - Phase Current Magnitude

The phase current magnitudes at the substation (the supply end of the feeder) were calculated. Figure 4.20 shows the maximum annual current magnitudes seen at each minute of the day, when all households have PV, when 60% of households have PV and when none do. The shown values represent a 'single phase' case as this is the state in which the phase current magnitudes are highest. The values can be divided by three to get an approximation of the currents in a balanced case. The rating, 458A [220], of the cable (185mm^2 XLPE/SWA buried direct) is not exceeded at any point. The introduction of PV lessens the maximum current seen during the middle of the day. At above 60% PV, the midday maximum current magnitude increases again – in the reverse direction. At 100% PV this reverse current exceeds the maximum load current that would have occurred with 0% PV – this is shown in Figure 4.20 (by comparison of the 100% and 0% lines). The addition of PV therefore increases the allowable load connection during the middle of the day. However, owing to the nature of the 'clear-sky' PV simulation, this effect may be

less evident in reality than indicated here. Also, any demand response taking advantage of the additional headroom would need to react to sudden changes in cloud cover.

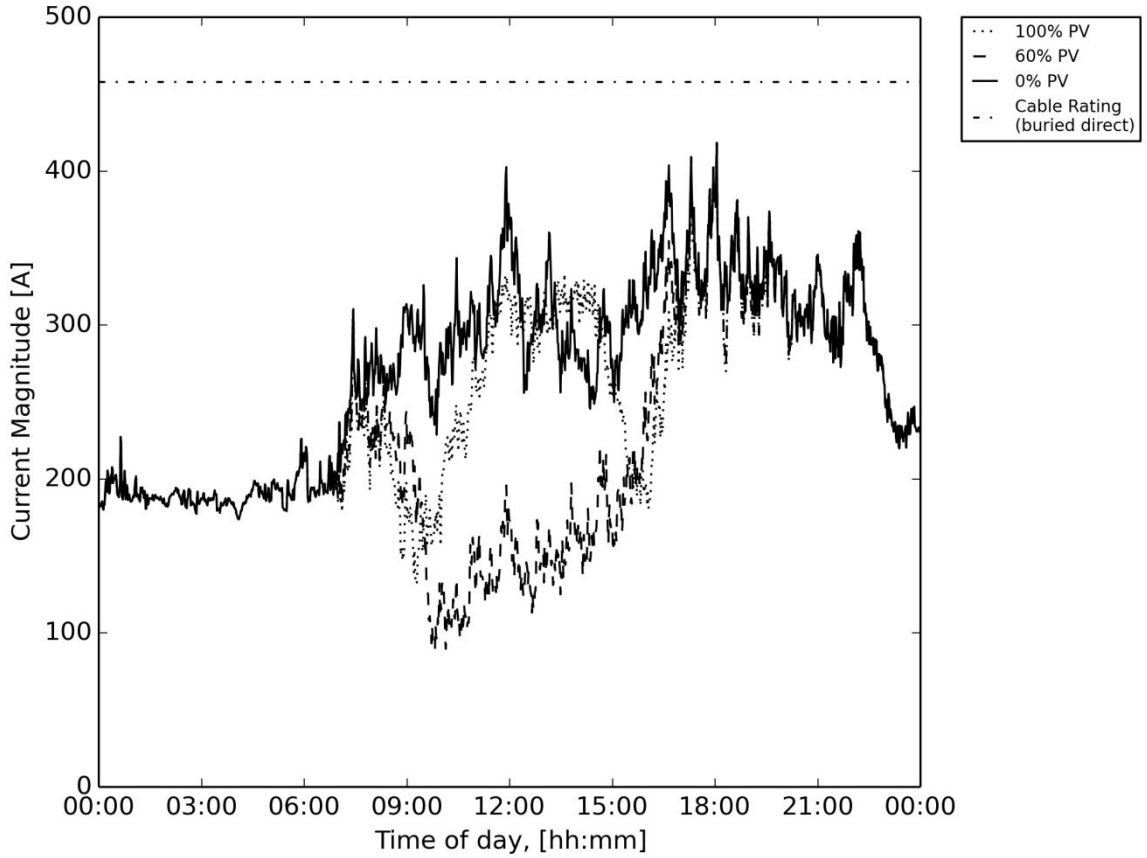


Figure 4.20 - Maximum phase current magnitudes at each minute of day for the 6 simulated months (Jan, Mar, May, Jul, Sep and Nov)

4.7 Results - Neutral Current Magnitude

As might be expected, the maximum neutral current increases with increased connection imbalance, see Figure 4.21. The increase in PV penetration has little influence on the maximum current. There are some configurations for which adding PV reduces the maximum current seen – this is particularly evident from 60 to 70% of households on a phase. This is due to the load reducing effect (from perspective of customer's meter point) that PV has during the middle of the day. Generally, however, these effects are not significant.

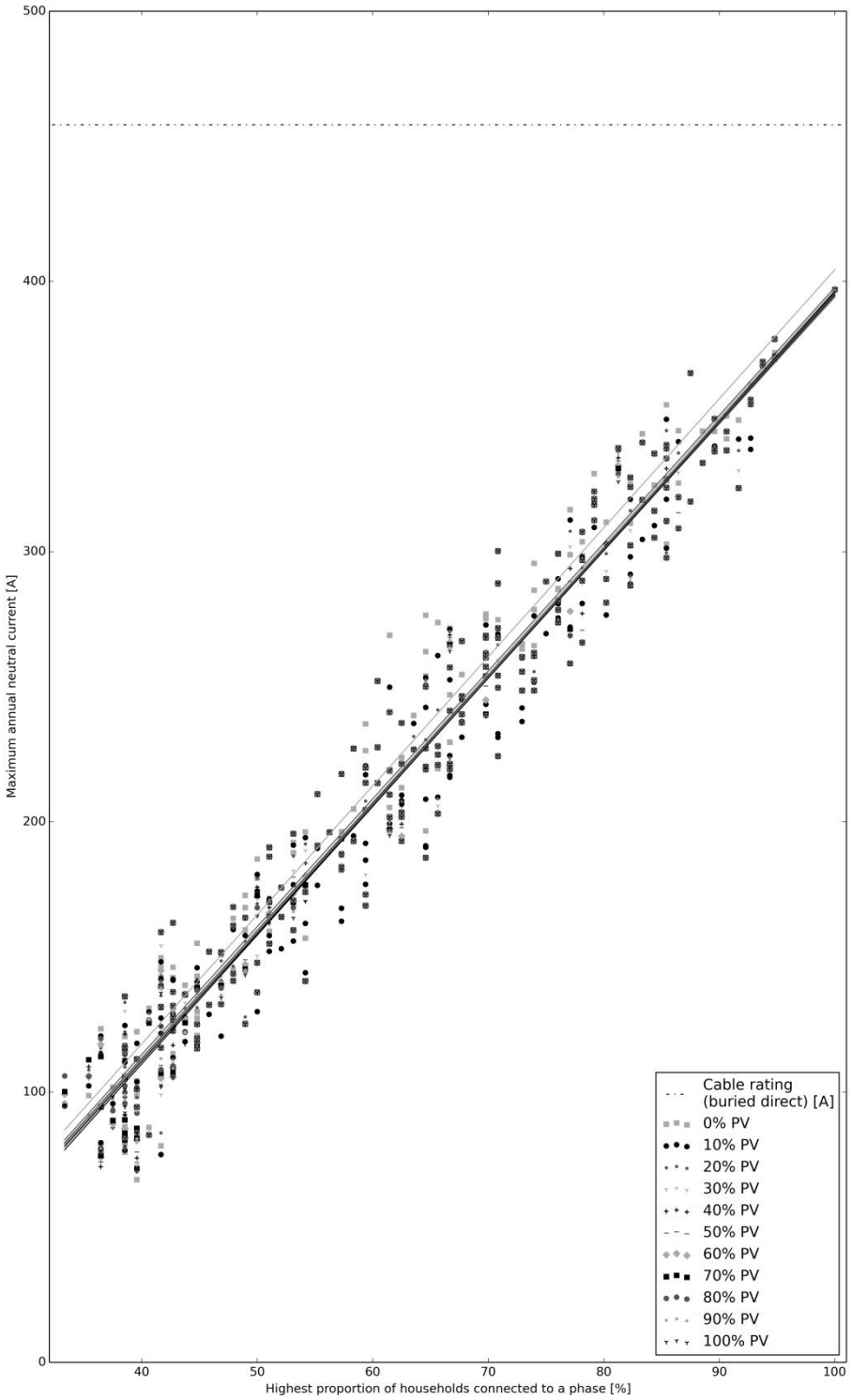


Figure 4.21 - Maximum annual neutral current for varying connection imbalance and PV penetration

4.8 Results - Losses

The losses incurred in the feeders were calculated over a 24 hours for the 11 PV penetrations for each of the six modelled months. With an increasing PV penetration, the losses first decrease and then (at above 70% PV) start to increase again as the feeder starts to export power. Figure 4.22 shows the mean values of the preceding 6 graphs for each PV penetration. The values for 0% PV range from approximately 7.8 kWh to 16kWh. A crude calculation, multiplying the mean daily values by 365, results in maximum annual losses of 5.8 MWh (occurs at maximum connection imbalance – single phase) and minimum annual losses of 2.8MWh (with a balanced connection of households). This represents a range of 0.7 to 1.4% of the feeder's projected annual demand. These values do not account for weekends or public holidays. Therefore, in practice, these loss values may vary significantly. Also, these approximated annual losses for feeders with PV are underestimated due to the clear-sky assumption– in reality periods of cloud cover would reduce PV output and so the cable losses would not be offset by as much. However, the general trends, increased losses with connection imbalance and first reducing then increasing losses with PV, will remain.

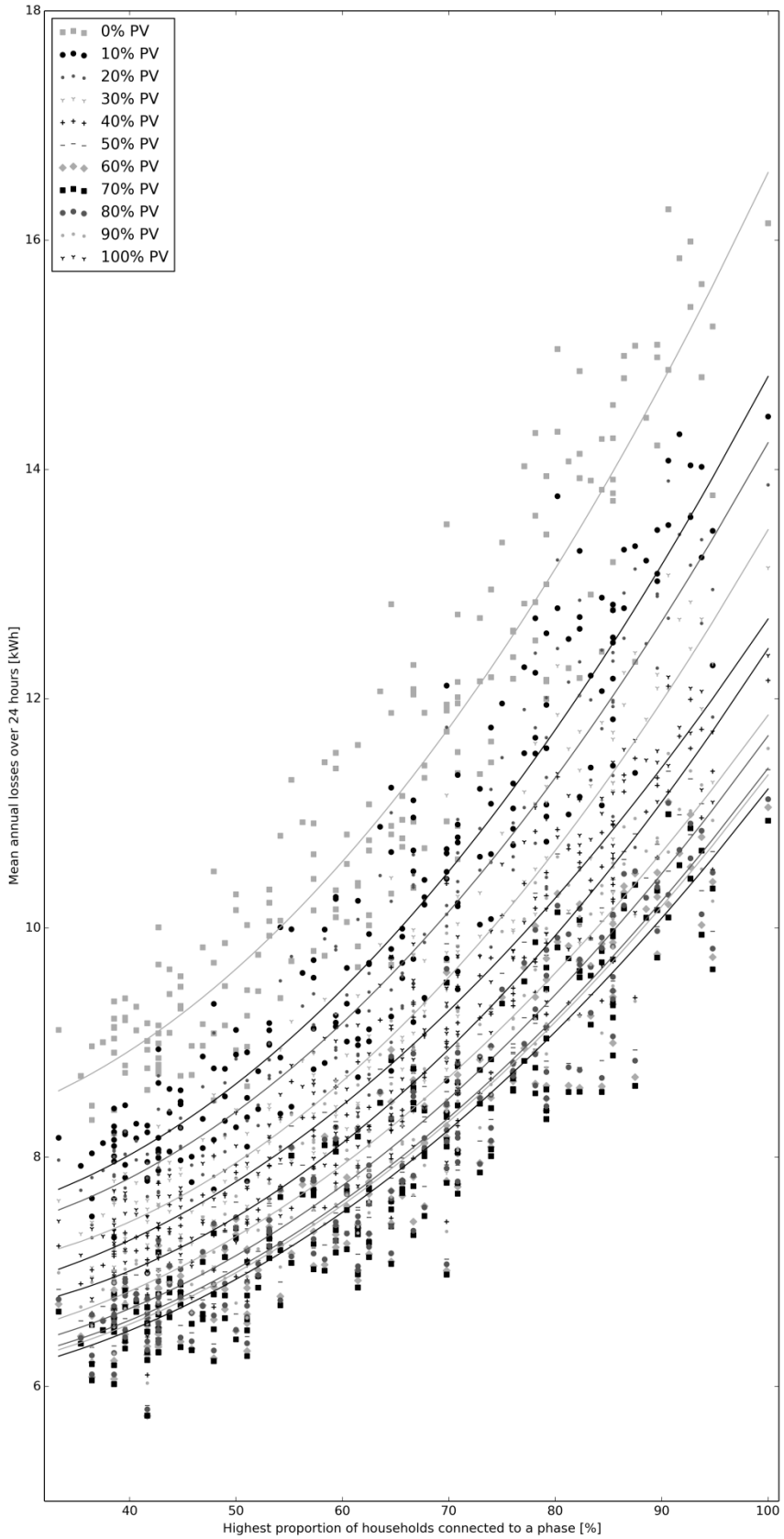


Figure 4.22 – Mean annual 24 hour losses for each of the feeders with varied proportion of households with PV.

4.9 Discussion

Based on the example feeder, the problem of connection imbalance can be treated as low priority by DNOs unless >~75% of houses are connected to a single phase, at which point voltage excursions will occur. This limit drops to ~60% if all households are given 1.3kWp PV. The limit would drop further if:

- either load or generation were significantly increased.
- the cable impedances were significantly increased.
- the 11kV voltage varies unsympathetically (i.e. low when meter voltages are low and high when meter voltages are high).

As connection imbalance increases, the annual voltage range on the feeder increases, the line and neutral current maxima increase and the losses increase. Therefore, to maximise asset utilisation and DG hosting capacity, DNOs should assess connection imbalance and take steps to compensate.

In this chapter, the allowable connection imbalance (highest proportion of meters connected to a phase) was found for a varied penetration of clear-sky PV.

The work in this chapter represents a method of expressing connection imbalance (highest proportion of meters connected to a phase) vs the annual voltage range and maximum line current magnitude. If the connection configuration and cable parameters are known, and the method described in this chapter is used, based on demand data retrieved from smart meters, then the PV hosting capacity, for a given configuration, can be estimated from the graphs produced.

As more PV is connected the losses in the feeder reduce. This is because the PV has a net load-reducing effect – much of the generated power is consumed at or near the household with PV. However, this effect is reversed, with additional PV causing increased losses, at clear–sky PV penetrations of greater than 70%. At this level the increased generated PV current not only supports the loads but causes a higher current to flow in the cabling than there would have been there if there were only demand. An estimation of maximum annual losses of 5.8MWh was made, reducing to 2.8MWh with 70% clear-sky PV penetration. The assumption of having cloudless days, whilst a useful tool for assessing the effects (on maximum meter voltage) of increasing PV generation, means that the minimum loss value is lower than expected. Despite this, for every feeder, there exists an optimum level of connected generation per household, in terms of the expected losses in the feeder cabling. This optimum is related to the demand near the generators. If the demand

could be made to respond to the available generation (or vice versa) then a reduction in losses would occur and the optimum (in terms of losses) amount of PV would increase. The potential for, and value of, such demand response schemes should be investigated.

Active control is likely to be required if adjacent LV feeders at the same substation are towards opposite extremes of PV penetration (i.e. high load or high PV). There was found to be overlap between the times that meter voltage maxima and minima occur. This overlap decreases as the clear-sky PV penetration increases – however the annual meter voltage range also increases. If two feeders, with simultaneous voltage minima and maxima, are connected to the same LV substation busbar, the potential for use of automatic transformer tap changers is limited

No voltage excursions occurred during the timeframe 0000-0600hrs. Therefore, this timeframe can be ignored when assessing voltage levels and PV, this would reduce the computing time required by ~25%.

The measure of connection balance used in this chapter, the maximum number of houses connected to a phase, was useful in that connection imbalance was reduced to a single value – allowing it to be easily used in graphs. However, the measure involved ignoring what was connected to the remaining phases. For example, a feeder with 60% of houses connected to one phase might have the remaining 40% shared equally between the remaining two phases or it might have it entirely on only one of the remaining phases. Both of the cases in this example would occupy the same spot (60%) on the x-axis of, for example, Figure 4.16. The example cases would, however be likely to occupy different locations on the y-axis. This, in part, accounts for the spread of points around the lines of best fit in the graphs in this Chapter.

The cable thermal rating was not exceeded in the simulations. Even in the single phase case, the current never came within 50A of the cable rating. If the cable cross sectional area were to drop to 120mm^2 (buried direct rating of 363A [220]) then some thermal overload would be likely in the single phase case. Considering the current alone, in the balanced case, assuming a maximum current of approximately 140A, the cable diameter could be dropped to 25mm^2 (buried direct rating of 152A)! However, realistically, this would be infeasible due to voltage drop and excessive losses. Therefore, meter voltage excursions are the primary factor limiting the demand and generation hosting capacity of the feeder.

5 Using Smart Meter Data for Phase Identification

5.1 Introduction

In Chapter 3, an LV feeder model (that allows for variation of single phase connection imbalance) was developed. In Chapter 4, the model was used, with a stochastic demand model, to investigate the influence of connection imbalance and PV penetration on meter voltages and line currents. This chapter used the output meter voltages, power profiles and line currents from Chapter 4 to develop and test a phase identification algorithm.

The developed phase identification algorithm uses line currents measured at the substation, as well as smart meter mean power and voltages, to identify the phase to which each meter is connected on a feeder. The algorithm has two steps. In the first step, meters are divided into ten groups, by comparison of voltages recorded by each smart meter (referred to as the "voltage clustering method") – this is done using hierarchical agglomerative clustering. The second step uses the ten groups and treats the problem as a subset sum problem (referred to as the "subset sum method") using currents measured at the supply busbar. The complete algorithm is referred to as "the phase identification algorithm"; an overview is shown in Figure 5.1. Also, to aid description, terms are defined:

> ***Measurement*** *– a single reading from a single smart meter*
>
> ***Profile*** *– a set of T consecutive measurements from a single smart meter*
>
> ***Cluster*** *– a weighted mean grouping of more than one profile or cluster*

The description has been divided into two parts. The first, covered in Sections 5.2 to 5.4, deals solely with the voltage clustering method. The voltage clustering method, on its own, was tested for correctness of the phase grouping and its limitations were examined. The second part, covered in Sections 5.5 to 5.7, combines the voltage clustering method with the subset sum method, forming the complete phase identification algorithm. The complete phase identification algorithm was tested and compared to the outcome from the voltage clustering method.

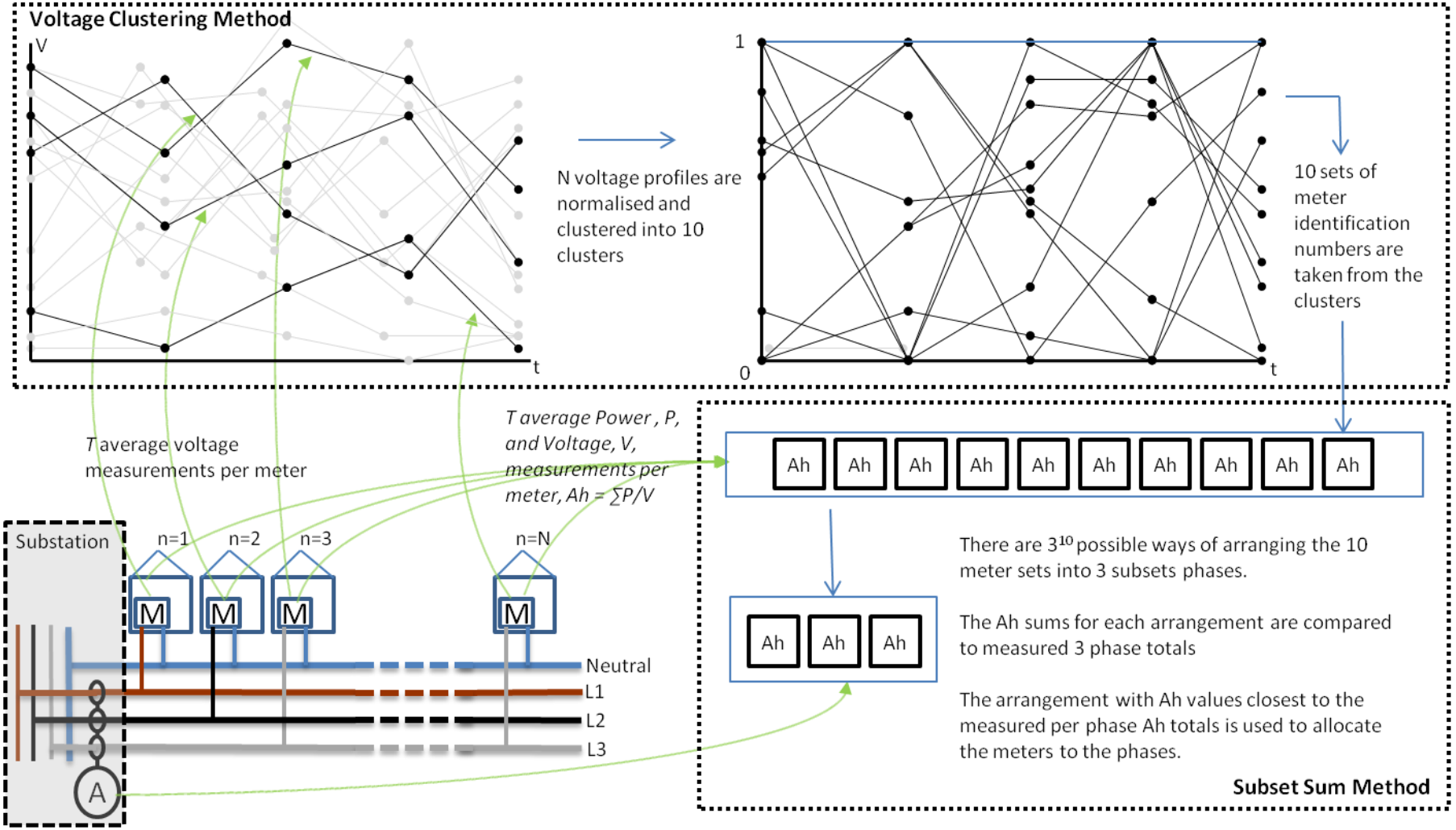


Figure 5.1 - Overview of the final phase identification algorithm

5.2 The voltage clustering method

The voltage clustering method uses hierarchical clustering, specifically agglomerative clustering with a squared Euclidean distance metric. The six steps comprising the voltage clustering method are shown in Figure 5.2. The first step is the acquisition of voltage profiles (consisting of *T* average voltage measurements) from *N* smart meters. An example meter's 24 hour, half-hourly (*T*=48) voltage profile is shown in Figure 5.3. The second step is to normalise each of the profiles (scale them so that the maximum value of that profile is 1 and the minimum value is 0), see Appendix A5.

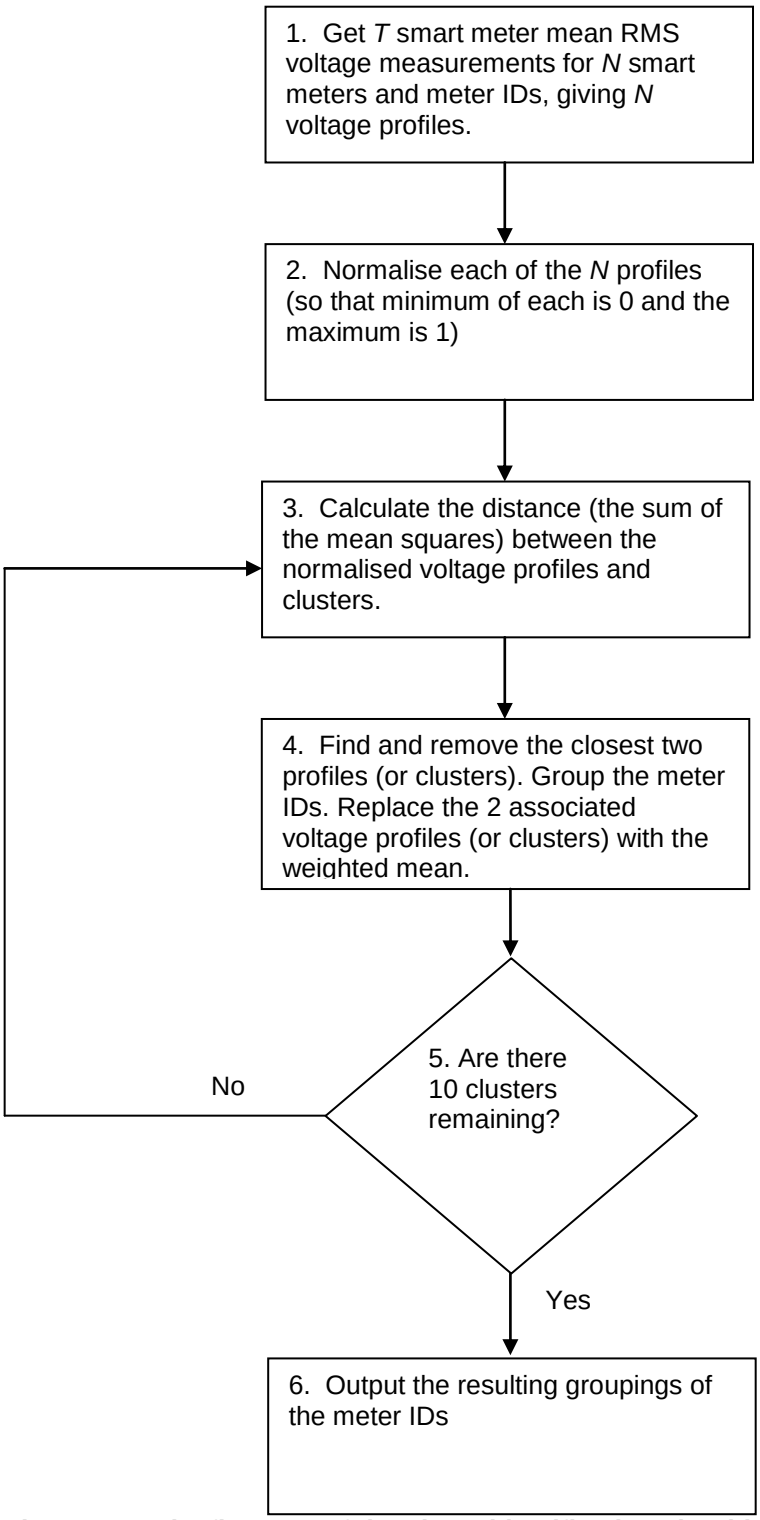


Figure 5.2 - The first part of the phase identification algorithm - referred to as the "voltage clustering method"

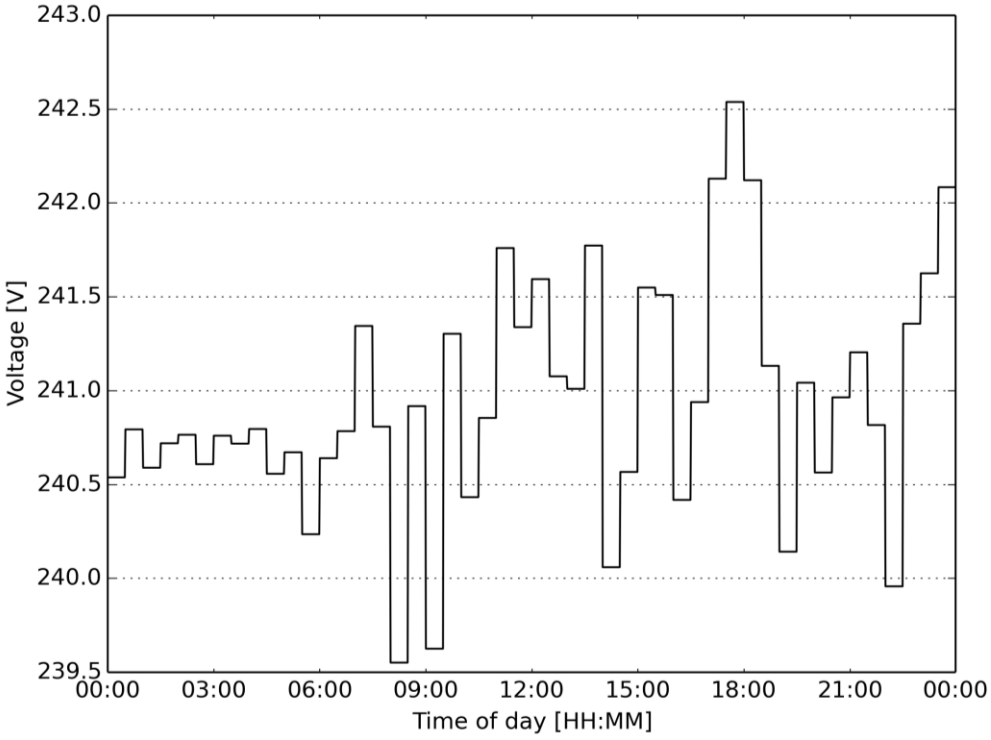


Figure 5.3 - An example mean half-hourly RMS voltage for a smart meter.

To cluster the meters, the distance between all of the normalised voltage profiles is calculated (step 2 in Figure 5.2). This is done using the sum of the differences squared. The distance between two profiles, *x* and *y*, is given in Equation 5.1.

$$Distance = \sum_{t=0}^{t=T} (x_t - y_t)^2$$

Equation 5.1

Where *x* and *y* are two normalised voltage profiles (step 3 in Figure 5.2), *t* is the measurement number, *T* is the total number of measurements (e.g. *T* is 48 if *x* and *y* are 24 hour profiles of half hourly mean measurements).

Once distance values have been calculated between all of the voltage profiles, the two closest profiles are found. This is done by first forming a distance matrix (see Figure 5.4) and then searching that matrix for the lowest non-zero value. If there are N profiles, the distance matrix has a size of N x N. It consists of the calculated distance values between each of the profiles, where the indices (row and column numbers) relate to the profile number. Its diagonal is all zeros (as this position represents a profile's distance to itself) and each distance value is repeated above and below the diagonal. The indices are used to identify the closest two profiles. This is indicated as indices 2 and 5 in the distance matrix of Figure 5.4.

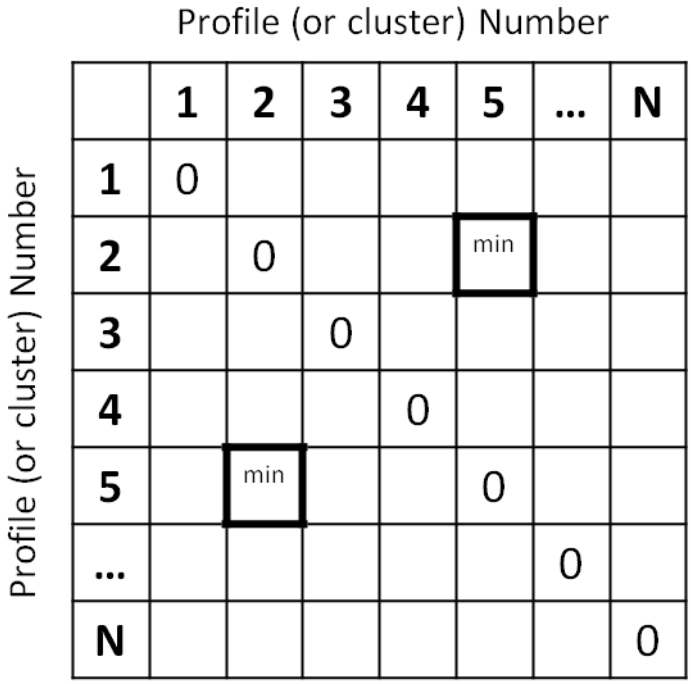


Figure 5.4 - The distance matrix for N profiles (or clusters) with the locations of an example minimum value indicated.

The two closest profiles (or clusters) are removed from the set of normalised voltage profiles (and clusters) and replaced with the weighted mean. The expression to calculate the weighted mean is shown in Equation 5.2. As an example, if, at step 4 in the flow diagram of Figure 5.2, two clusters are selected; cluster x representing 2 meters and cluster y representing 10 meters, the resulting cluster will be the weighted mean of 2 of the x clusters and 10 of the y clusters.

$$\boldsymbol{Normalised\ Voltage\ Weighted\ Mean} = \frac{\boldsymbol{Kx_t + My_t}}{\boldsymbol{K + M}}\ \forall \boldsymbol{t} \in \{\boldsymbol{0, 1, \ldots, T}\}$$

Equation 5.2

where *x* and *y* are clusters (the two closest of the remaining normalised voltage profiles and clusters - step 4 in Figure 5.2), *t* is the measurement number, *T* is the total number of measurements and *K* and *M* are the number of meters they represent.

The distance calculation and profile (or cluster) replacement steps (steps 3 and 4 in Figure 5.2) are repeated until 10 clusters remain.

To test the voltage clustering method (on its own), it was adjusted so that it outputs three sets of meter identification numbers, see Figure 5.5. The sets of meters relate

to the phase to which they are connected. The voltage clustering method does not discern which of the three phases each group is connected to, just the groupings.

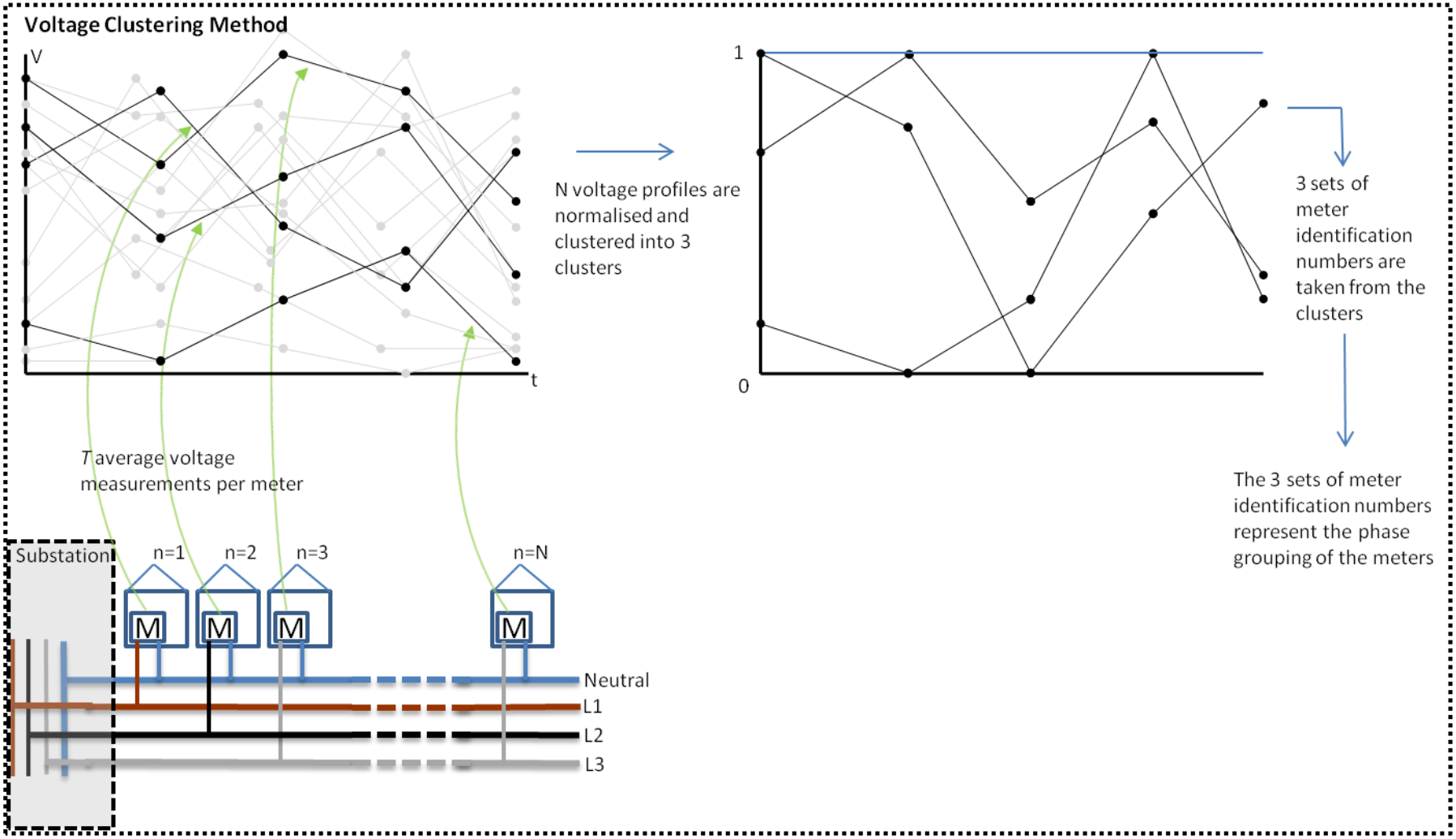


Figure 5.5- Overview of the voltage clustering method for testing

5.3 A simple example of the voltage clustering method

Consider an example feeder with seven metered households. The voltage profiles across 24 hours are those shown in Figure 5.6 – each measurement value represents a mean voltage across 30 minutes. The meters are each given an identification number: {1},{2},{3},{4},{5},{6} and {7}. In this example, as the voltage clustering method is used on its own, it is adjusted so that it outputs three sets of meter identification numbers, as per Figure 5.5.

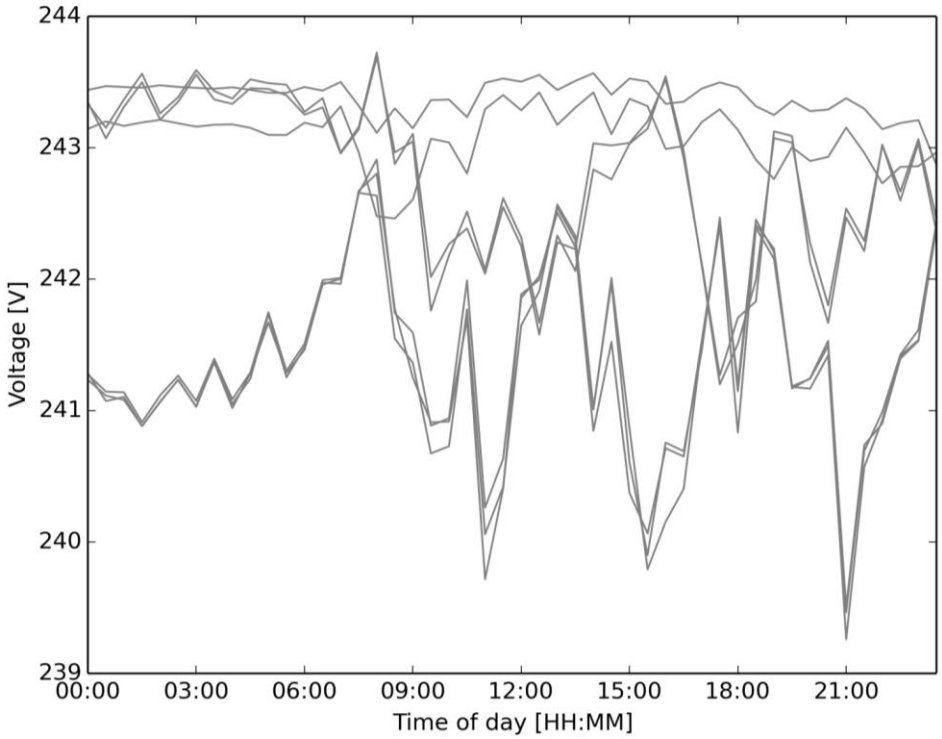


Figure 5.6 - Example of 7 smart meter voltage profiles. Each sample point represent a half hour mean voltage.

The first step is to normalise (scale between 0 and 1) the profiles. The normalised profiles are shown in Figure 5.7.

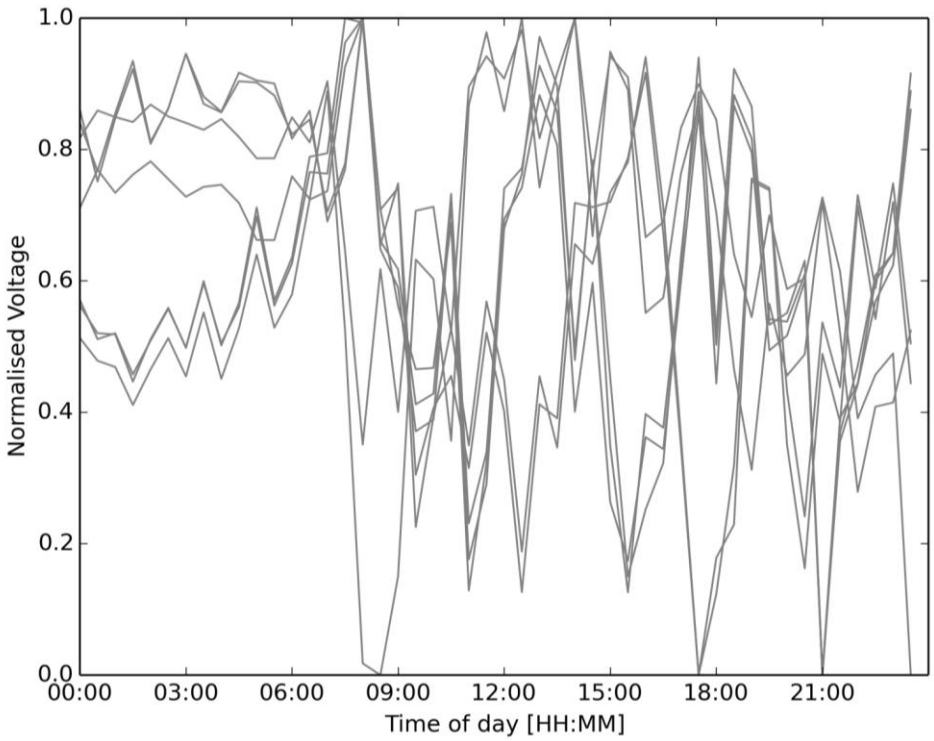


Figure 5.7 - The normalised voltage profiles

Next, the distance between all of the normalised voltage profiles is calculated (see Equation 5.1). The results are collected into a matrix, see Figure 5.8.

		Meter ID						
		1	**2**	**3**	**4**	**5**	**6**	**7**
Meter ID	**1**	0	38.08985	161.6782	165.602	187.6362	194.9637	182.0975
	2	38.08985	0	191.4658	193.0821	191.0142	191.8093	184.7624
	3	161.6782	191.4658	0	2.401794	174.9479	173.5315	170.3148
	4	165.602	193.0821	2.401794	0	182.4434	181.277	177.0893
	5	187.6362	191.0142	174.9479	182.4434	0	4.425732	2.57935
	6	194.9637	191.8093	173.5315	181.277	4.425732	0	4.925635
	7	182.0975	184.7624	170.3148	177.0893	2.57935	4.925635	0

Figure 5.8 - An example distance matrix - the "distance" between the normalised voltage profiles of each meter. The lowest non-zero values are highlighted.

This distance matrix is then searched for the minimum non-zero value. The matrix indices of the minimum non-zero value indicate the closest two profiles. Therefore, these meters are grouped, to give; {1},{2},{5},{6},{7} and {3,4}. Following this, the normalised voltage profiles associated with meters 3 and 4 are combined as the mean, shown in Figure 5.9

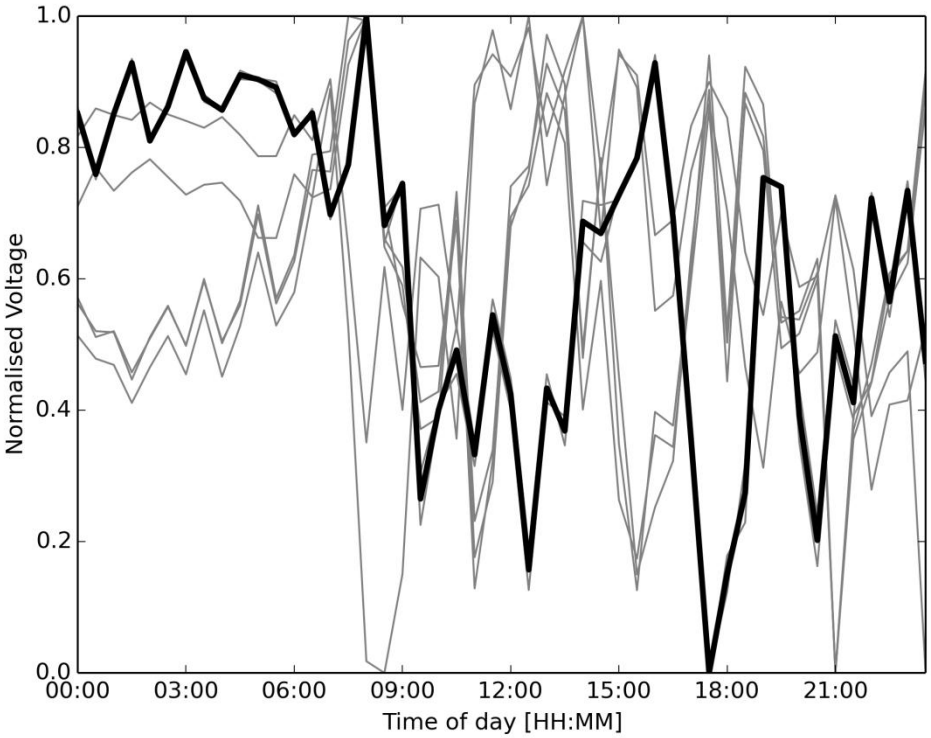


Figure 5.9 - The example 7 normalised meter profiles. The two closest will be replaced by a cluster (highlighted).

The process is repeated – finding the distance matrix, locating the non-zero minimum, grouping the associated meter IDs and combining the voltage profiles. The next step in this example is to group meters 5 and 7, giving; {1},{2},{6},{3,4} and {5,7}. Again the voltage profiles are combined as the mean (highlighted in Figure 5.10).

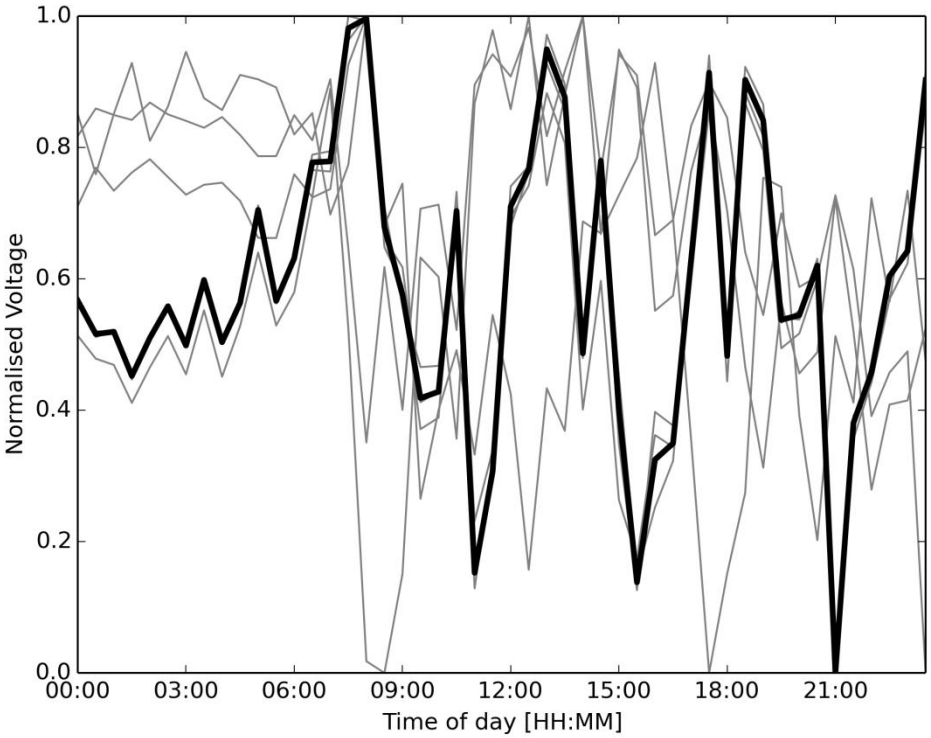


Figure 5.10 – The example set of 5 normalised profiles and 1 cluster. The two closest will be replaced by a cluster (highlighted).

In the next step, for this example, the closest are {6} and {5,7}, giving {1},{2},{3,4} and {5,6,7}. This time, however, as a cluster has been selected, the clustering is done using the weighted mean (see Equation 5.2), the resulting cluster is highlighted in Figure 5.11.

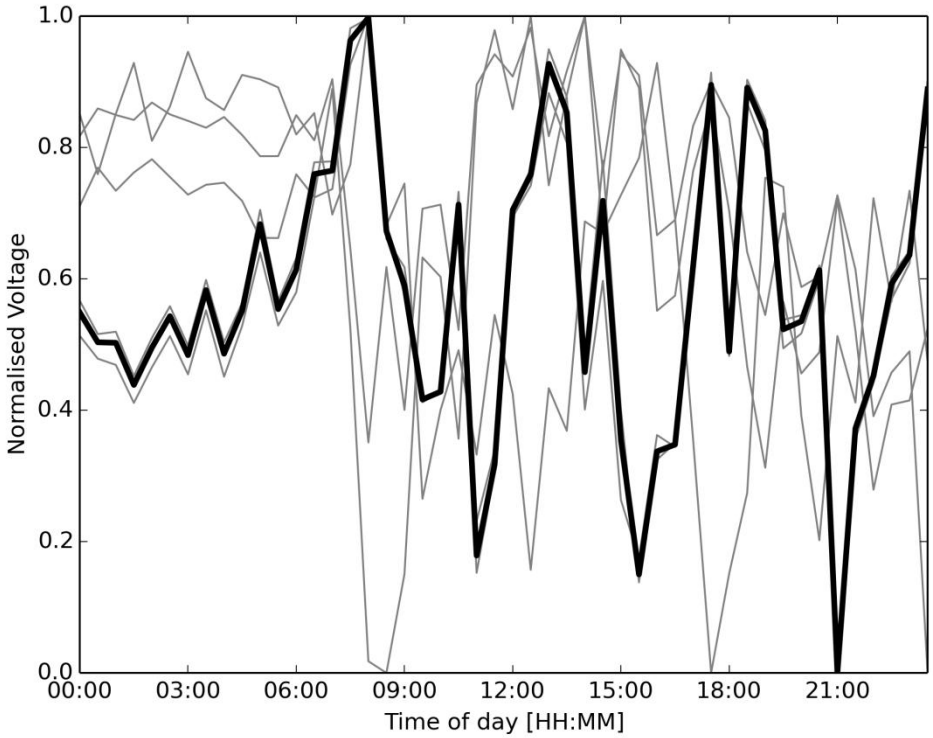


Figure 5.11 - The example set of 3 profiles and 2 clusters. The two closest will be replaced by a cluster (highlighted).

The final grouping is to combine meters 1 and 2 giving; {1,2},{3,4} and {5,6,7}. At this point there are three groups remaining, so the voltage clustering process, as shown in Figure 5.5, is complete.

5.4 Testing the voltage clustering method

To quantify the performance of the voltage clustering method, the proportion of correctly grouped meters was calculated, as defined in Equation 5.3. This compares the predicted meter grouping with the actual phase groupings. It is the ratio, expressed as percentage, of the correctly grouped meters to the total number of meters on the feeder.

$$Proportion\ of\ correctly\ grouped\ meters\ [\%] = \frac{100}{\sum_{h:3}|A_h|}\left[\max_{f:\boldsymbol{Perm}(3)}\sum_{h:3}\left|A_h \cap B_{f(h)}\right|\right]$$

Equation 5.3

where A is a set of the known phase groupings; a set of 3 sets containing the meter identification numbers. B is the predicted groupings output from the voltage clustering method (again, a set of 3 sets containing the meter identification numbers - representing the predicted phase groupings).

The voltage clustering method does not discern which of the three phases each group is connected to. Therefore the 3 predicted groups can be attributed to the actual 3 phases in 6 different ways. The ratio of the predicted to actual meters in each group is calculated for each of the 6 ways. The maximum (of the 6 ratios) is taken as the proportion of correctly grouped meters.

Consider the earlier example feeder (with 7 meter IDs; 1, 2, 3, 4, 5, 6 and 7) and phase groupings of {1,2},{3,4} and {5,6,7}. If the clustering groups the meters as {1,2},{3,4} and {5,6,7} then the proportion of correctly grouped meters is 100% . However, if the predicted phase groupings were {1,3,5}, {4} and {2,6,7}, the proportion of correctly grouped meters is 4/7,3/7 or 2/7 (x 100%), as per Equation 5.3. The maximum is chosen, so the proportion of correctly grouped meters is 57% (2 significant figures).

The voltage clustering method was tested using voltages produced in Chapter 4. A feeder with 96 dwellings was used. The dwellings were allocated to the phases in 100 different ways. The voltage data comes from unbalanced load flow performed using demand based on 6 days across the year. An averaging time, the time span over which the mean voltages were recorded, of 30 minutes was used. A 0%PV penetration was selected. The results are shown in Figure 5.12.

Each point in Figure 5.12 represents the proportion of correctly grouped meters from a single test of the voltage clustering method. The group of points along the top of the graph, at 100% on the y-axis, are where the voltage clustering correctly predicted the grouping of all of the meters. The group of points, spread, horizontally across the central part of the graph represent cases where approximately a third of the meters have not been grouped correctly. This occurs because, at a late stage in the process, two relatively large groups of meters are combined into a group of approximately two-thirds of the total number of meters on the feeder. It is likely that approximately half of the meters in that group (a third of the total number of meters) have been incorrectly combined, lowering the proportion of correctly grouped meters by ~33%. There are some points located towards the lower left of Figure 5.12. These represent cases where the majority of meters have been put into a single cluster, meaning that, on feeders with around same number of meters on each phase, around two thirds of the meters will have been incorrectly grouped, lowering the proportion of correctly grouped meters by ~66%. Finally, there are points located between the upper and central groupings, generally towards the right of the graph. These represent cases where relatively few meters have been incorrectly clustered.

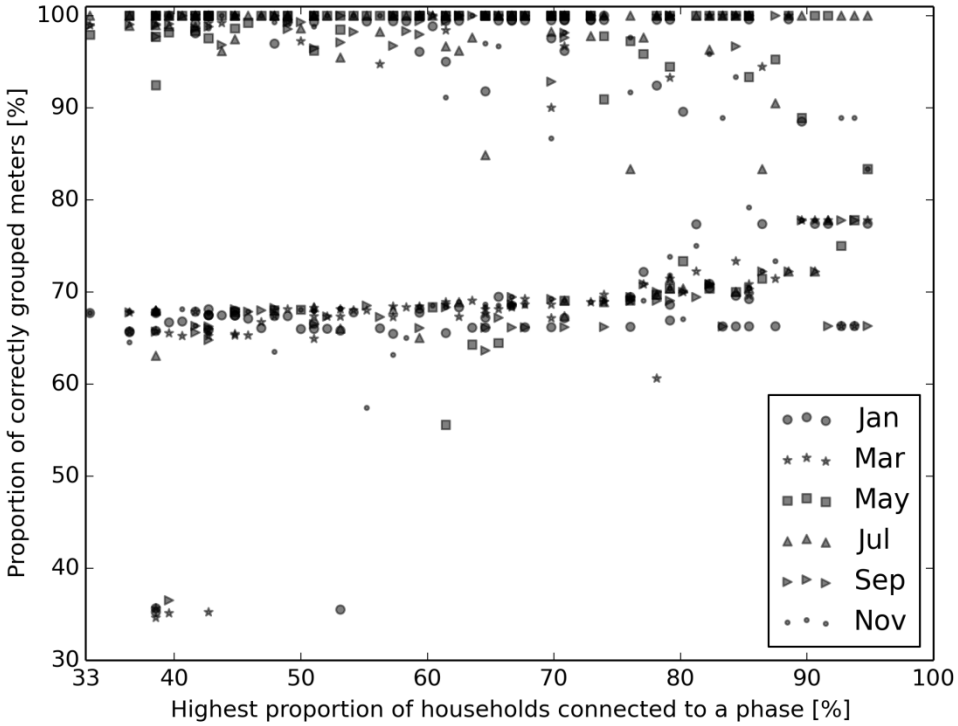


Figure 5.12 – Performance of the voltage clustering method with varying connection imbalance.

Next, the voltage clustering method was tested with varied averaging times, in six steps (160,144,120,60,30 and 15 minutes). In Figure 5.13, the results are presented as a probability distribution, repeated for the 6 averaging timeframes. Six sets of 600 tests were performed, with each test representing a run of the voltage clustering method with a different set of voltages from Chapter 4. The result was 6 sets of 600 values - the proportion of correctly grouped meters for each test. These values were ordered and plotted as a distribution, as in Figure 5.13.

This shows the probability of correctly grouping a meter across the 600 tests (6 months x 100 feeders). The area under the graph represents correctly grouped meters. The area above the graphs (up to a limit of 100%) represents incorrectly grouped meters. The largest area under the graph is for an averaging timeframe of 30 minutes. Therefore, the voltage clustering method is most successful when the averaging time is 30 minutes.

A distinctive feature of this graph is the drop in correctly grouped meters from 100 to ~70% that occurs approximately midway along the x-axis. This relates to the central grouping in Figure 5.12, resulting from the erroneous clustering of two large clusters.

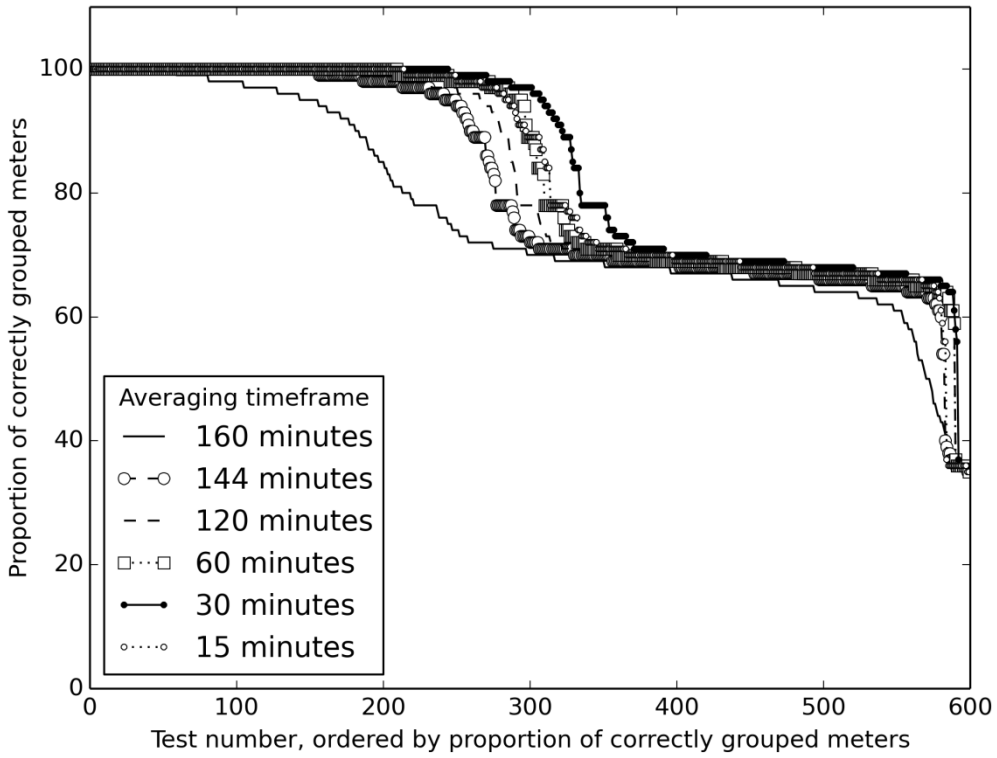


Figure 5.13 – Distribution of the proportion of correctly grouped meters for 6 averaging timeframes.

The voltage clustering method was tested with six PV penetrations (0 to 100% of houses with PV, in 20% steps). A 30 minute averaging timeframe was selected. Six sets of 600 tests were performed, with each test representing a run of the voltage clustering method with a different set of voltages from Chapter 4. Another distribution was plotted, for each of the PV penetrations, see Figure 5.14.

The increase of PV penetration had only a slight effect. In Figure 5.14, the proportion of correctly grouped meters begins to fall around the same area of the graph. The voltage clustering method appears to correctly group meters more often when 80% of houses had PV. This is likely to be due to a chance influence on the demand, in one of the month's data, causing a shift in voltage that changes the clustering decision. In general, the effect of PV on the voltage clustering method is not significant when a 30 minute averaging timeframe is used.

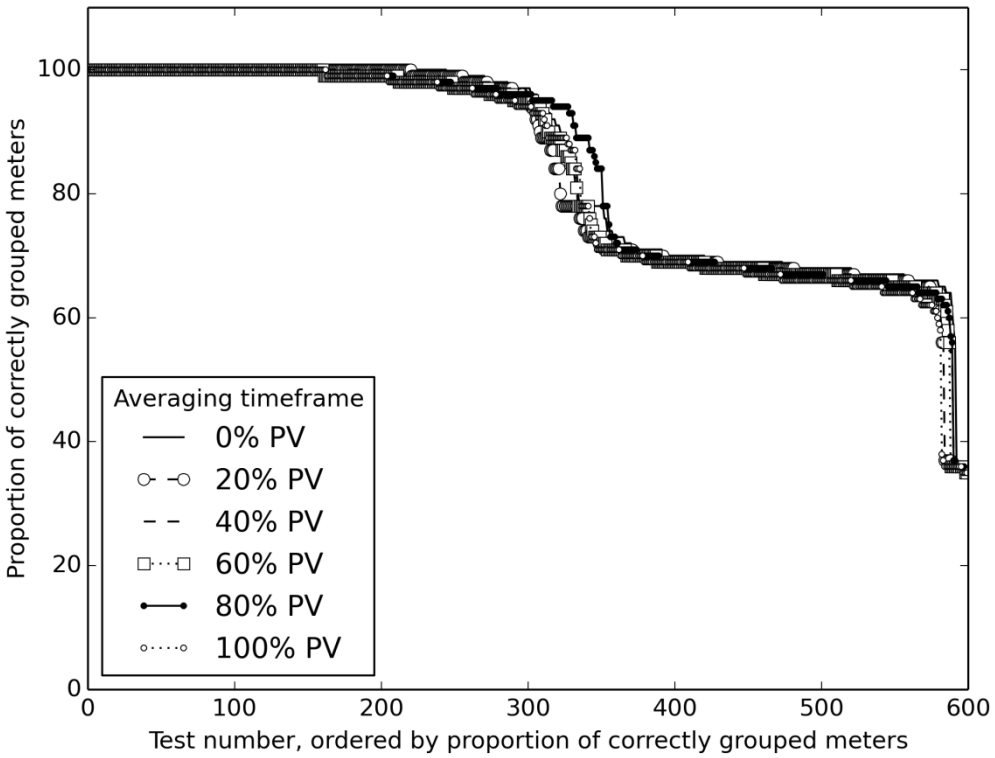


Figure 5.14 - Distribution of the proportion of correctly grouped meters for 6 PV penetrations.

To assess how far (the number of iterations) the voltage clustering method got before failure, the iteration at which it failed was recorded. As there are 96 meters on the tested feeders, the algorithm runs for a maximum of 94 iterations. At some point, if it fails, the algorithm will incorrectly group meters. As the actual phase configuration is known, it was possible to check the algorithm for failure at every iteration. If the algorithm incorrectly groups meters, it was deemed to have failed – at this point the number of iterations that the algorithm got to upon its first failure was recorded. The updated process, with the failure checking, is shown in Figure 5.15. A failure point of 1 indicates that the algorithm failed on its first iteration whereas a failure point of 94 indicates that the algorithm failed on the last iteration and almost completed successfully.

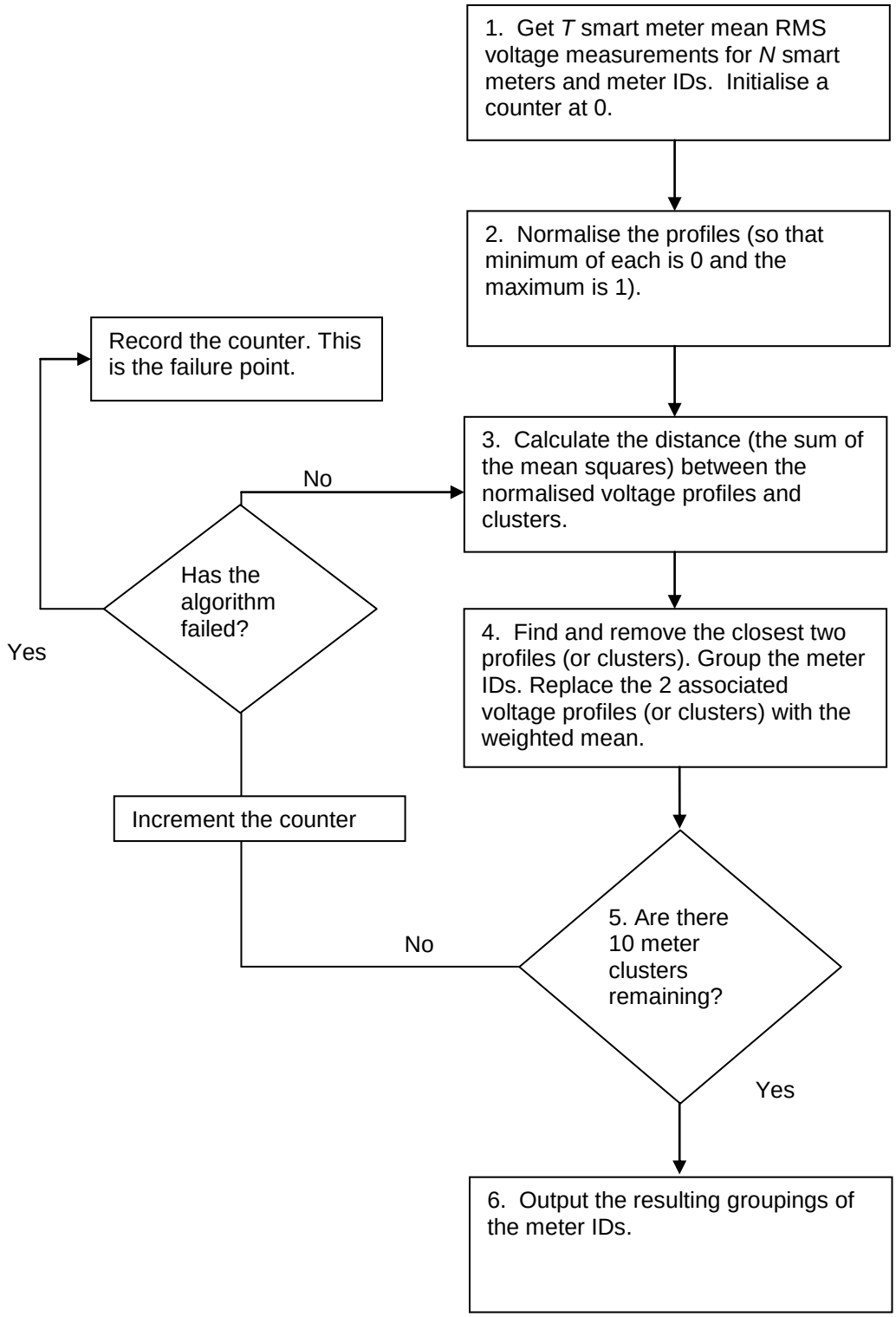


Figure 5.15 - The voltage clustering method with recording of the failure point

The distribution of the failure points, for 100 different feeders (for 6 sets of 96 load profiles and 6 PV penetrations) and 6 measurement timeframes, was recorded in Figure 5.16. The voltage clustering method tended not to fail until a high number of iterations was reached – this can be deduced from the grouping of the distributions towards the right of the graphs in Figure 5.16. For measurement timeframes of ≤30 minutes the voltage clustering method never failed before the 88th iteration was reached. For this reason, in the complete phase identification algorithm (Figure 5.1) the voltage clustering method was run until the 87th iteration, where10 groupings of meters remain. When 10 groups of meters remain, the subset sum method is used. If the number of houses on the feeder were to change, the distribution of the failure points may also change. Therefore, the choice of 10 groupings is specific to the Chapter 4 feeder with 96 dwellings.

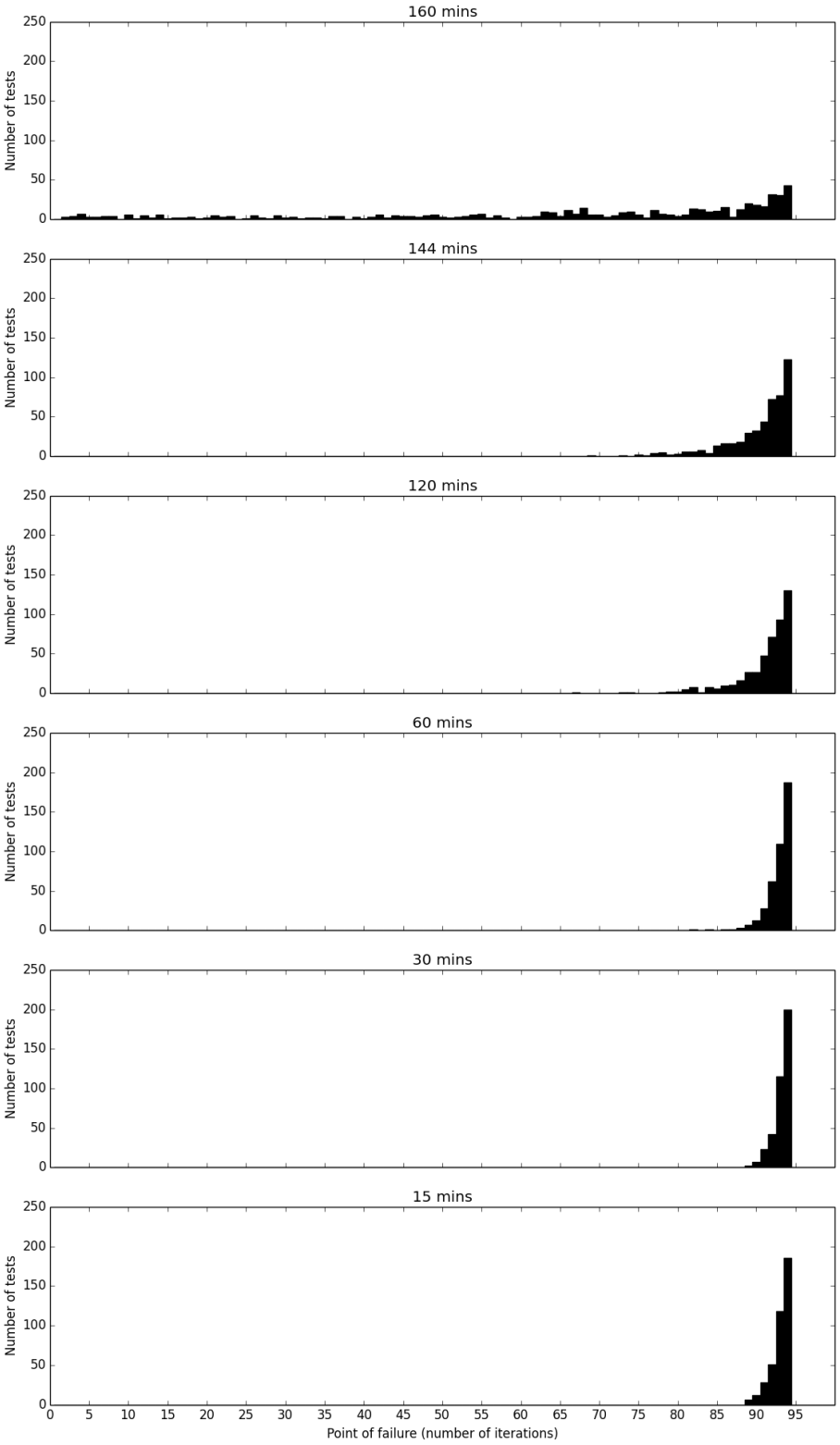


Figure 5.16 - Distribution of failure points for 6 averaging timeframes after running the voltage clustering method.

5.5 The subset sum method

The subset sum method finds the 3 mutually exclusive subsets, within a set of smart meters, whose currents sum to give the per-phase source currents at the substation. This is done by finding the subset sum currents for all possible combinations of the smart meters in three groups and comparing those sums to the three source totals. When the subset sum currents equal the source currents, the meters, within each of the three subsets, are the phase connections.

The subset sum method does not scale well. Complete solution of all 96 meters using the subset sum problem alone would require comparison of 3^{96} possible combinations. This is computationally unfeasible; A laptop with an Intel Core i5 CPU @2.53 Ghz performed a subtraction operation between two Numpy 32 bit floating point arrays of 96 elements in 1.37µs seconds (see Appendix A6). Even assuming (e.g. with faster processors and optimised code) that a subset sum iteration (comparison of a combination and the totals) and test could feasibly take place within this timeframe, 2.76×10^{30} centuries of computing time would be required to check all possible combinations! The scaling is illustrated in Figure 5.17. Arya's method [195] gets around the poor scalability by increasing the number of measurements used and formulating the problem so that it no longer a subset sum problem. Rather they formulate it as an Integer Linear Programming problem in such a way that it can be solved with existing solvers (e.g. CPLEX). However, despite its poor scalability, a subset sum method can be feasibly applied to small sets. In particular, using the subset sum method on the last 10 groupings from the voltage clustering method, where it tends to fail (see Figure 5.16), is feasible.

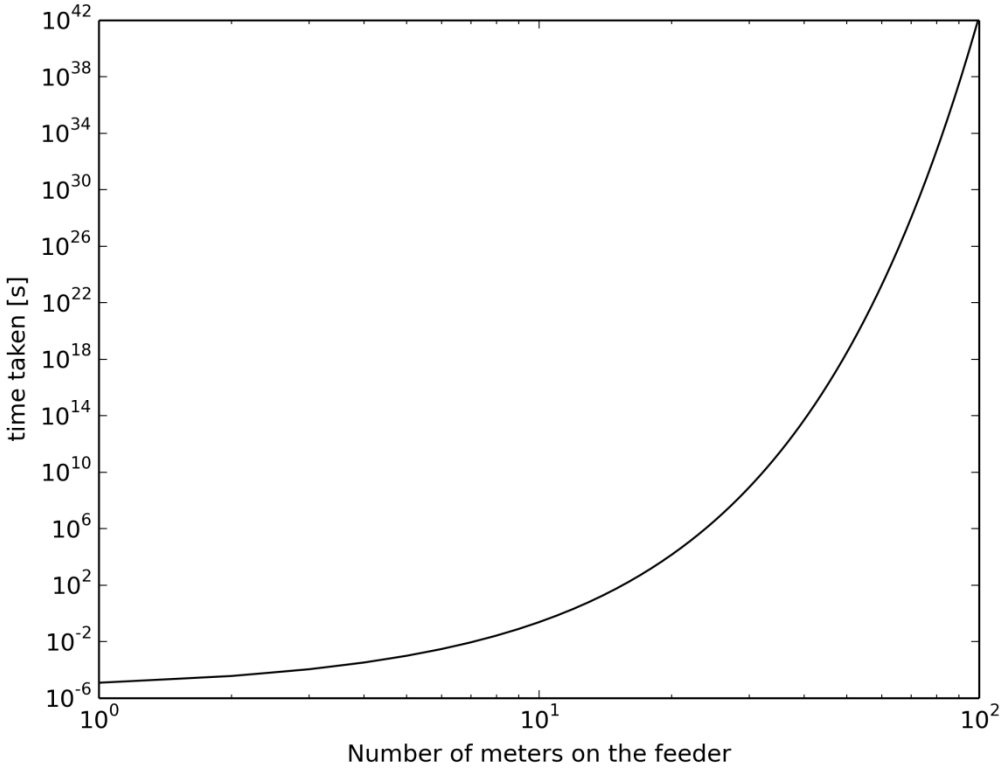


Figure 5.17 Scaling as 3^N (based on a 1.37µs time for a single operation).

The first step of the subset sum method is the input of meter groupings from the voltage clustering method (Step 1a in Figure 5.18). Next, the total net charge transferred through each meter is calculated. To do this, the mean currents for each meter were first calculated using the smart meter mean real power measurements with the smart meter voltage magnitude measurements. As the mean power and voltages will be available over a predefined averaging timeframe, and the data from Chapter 4 is in 1 minute intervals, the mean currents were approximated as shown in Equation 5.4.

$$I_{meanT} = \frac{\overline{P_1}}{\overline{V_1}} + \frac{\overline{P_2}}{\overline{V_2}} + \frac{\overline{P_3}}{\overline{V_3}} + \cdots + \frac{\overline{P_T}}{\overline{V_T}} \approx \left(\frac{\overline{(P_1 + P_2 + P_3 + \cdots + P_T)}}{\overline{(V_1 + V_2 + V_3 + \cdots + V_T)}} \right)$$

Equation 5.4

where I_{meanT} is the mean current over the averaging timeframe, P_t is the smart meter power for minute *t*, V_1 is the meter voltage for minute *t* and *T* is the averaging timeframe in minutes.

The mean currents were then summed in accordance with the 10 meter groupings output from the voltage clustering method (see Equation 5.5).

$$\boldsymbol{Ah_{meter-groups}} = \sum_{\boldsymbol{m=0}}^{\boldsymbol{m=|g|}} \sum_{\boldsymbol{t=0}}^{\boldsymbol{t=T}} \boldsymbol{I_{m,t}} \quad \forall \boldsymbol{g} \in \boldsymbol{G}$$

Equation 5.5

where G is the set of 10 meter identification groups output from the voltage clustering method, m is the meter number within a group, t represents the measurement number, *T* is the total number of measurements $I_{m,t}$ is the tth mean current for the $\boldsymbol{m}$th meter (see Equation 5.4).

Three current totals (per-phase) were taken from the data in Chapter 4. In reality, they would be measured at the substation (step 1b in Figure 5.18). The power factor is also required, so that the real component of the current can be calculated (see Equation 5.6).

$$\boldsymbol{Ah_{in-\emptyset}} = \sum_{\boldsymbol{t=0}}^{\boldsymbol{t=T}} \boldsymbol{Re\{I_{\emptyset-t}\}}$$

Equation 5.6

where $I_{\emptyset-t}$ is the tth mean real current on the phase ∅ recorded at the supply substation. T is the total number of samples.

The phrase “transferred charge” and the shorthand Ah is used to denote the values calculated with Equations 5.5 and 5.6. However, to actually get the transferred charge in units of ampere hours, the values would have to be divided by the number of measurements per hour (e.g. 2 for half hourly measurements). As this was not required for the algorithm to work, it was not done. But the *Ah* notation was retained here.

The Subset Sum Method finds the 3 subsets within the 10 meter charge transfer values that sum to give the 3 recorded phase conductor totals (the *sums*). All possible configurations of the 10 phase groupings, the powerset $\wp(Ah_{meter-groups})$, across the three phases were found and the total charge transfer values obtained for each phase. Each of these sets of three values were subtracted from the *sums* (the measured input values). The subset sums closest to the input values then represent the per phase meter grouping, see Equation 5.6. A flow diagram of the subset sum method is shown in Figure 5.18.

$$Subset\ sum\ function =$$

$$\underset{x,y,z}{\arg\min}\left((Ah_{in-1}, Ah_{in-2}, Ah_{in-3}) - \left(\sum_{n=0}^{|x|} x_n , \sum_{n=0}^{|y|} y_n , \sum_{n=0}^{|z|} z_n \right) \right)$$

$$\forall x \in \wp\left(Ah_{meter-groups}\right)$$

$$\forall y \in \wp\left(Ah_{meter-groups}\right)\ if |x \cap y| < 1$$

$$\forall z \in \wp\left(Ah_{meter-groups}\right)\ if |x \cap y \cap z| < 1$$

Equation 5.7

where Ah_{in-1}, $Ah_{in-2}$and $Ah_{in-3}$are the input charge transfer values (Equation 5.5), $Ah_{meter-groups}$ is the set of 10 charge transfer values calculated from the per meter charge transfer (Equation 5.4) and the meter groupings output from the voltage clustering method. When the minimum difference is found, x, y and z approximately represent the per phase charge transfer values. The location of these values in the powerset, $\wp\left(Ah_{meter-groups}\right)$, is mapped onto the meter IDs to find the phase identification of each meter. This formulation assumes that all 10 values in $Ah_{meter-groups}$ are unique – which, practically, is true.

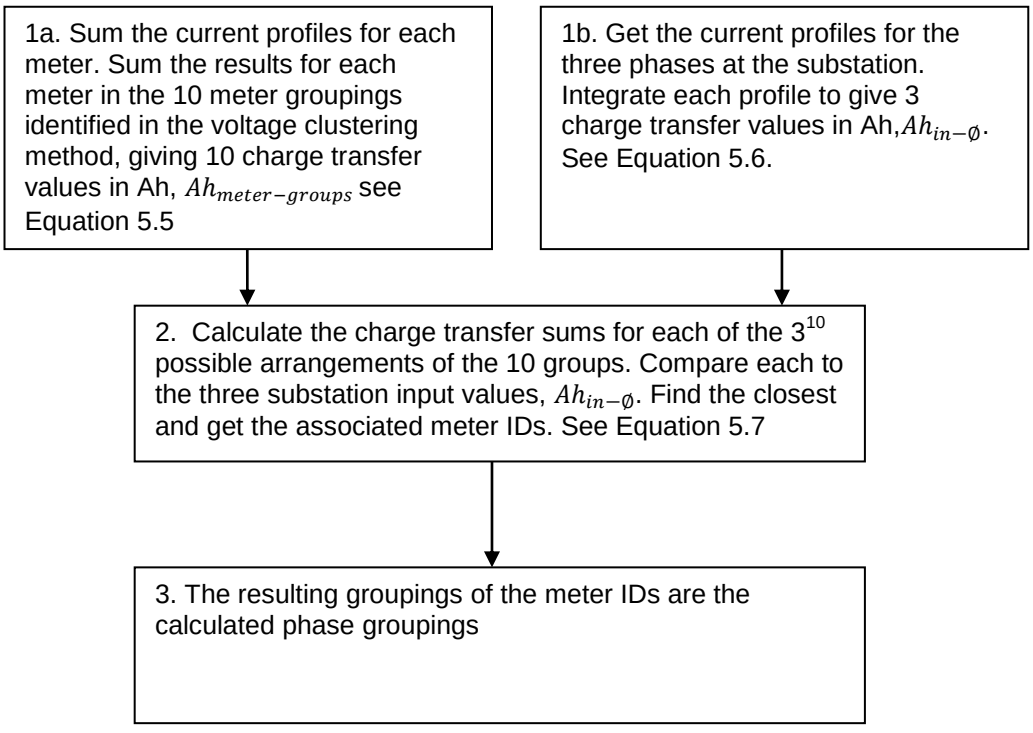


Figure 5.18 - The subset sum method used for the final 10 smart meter groupings.

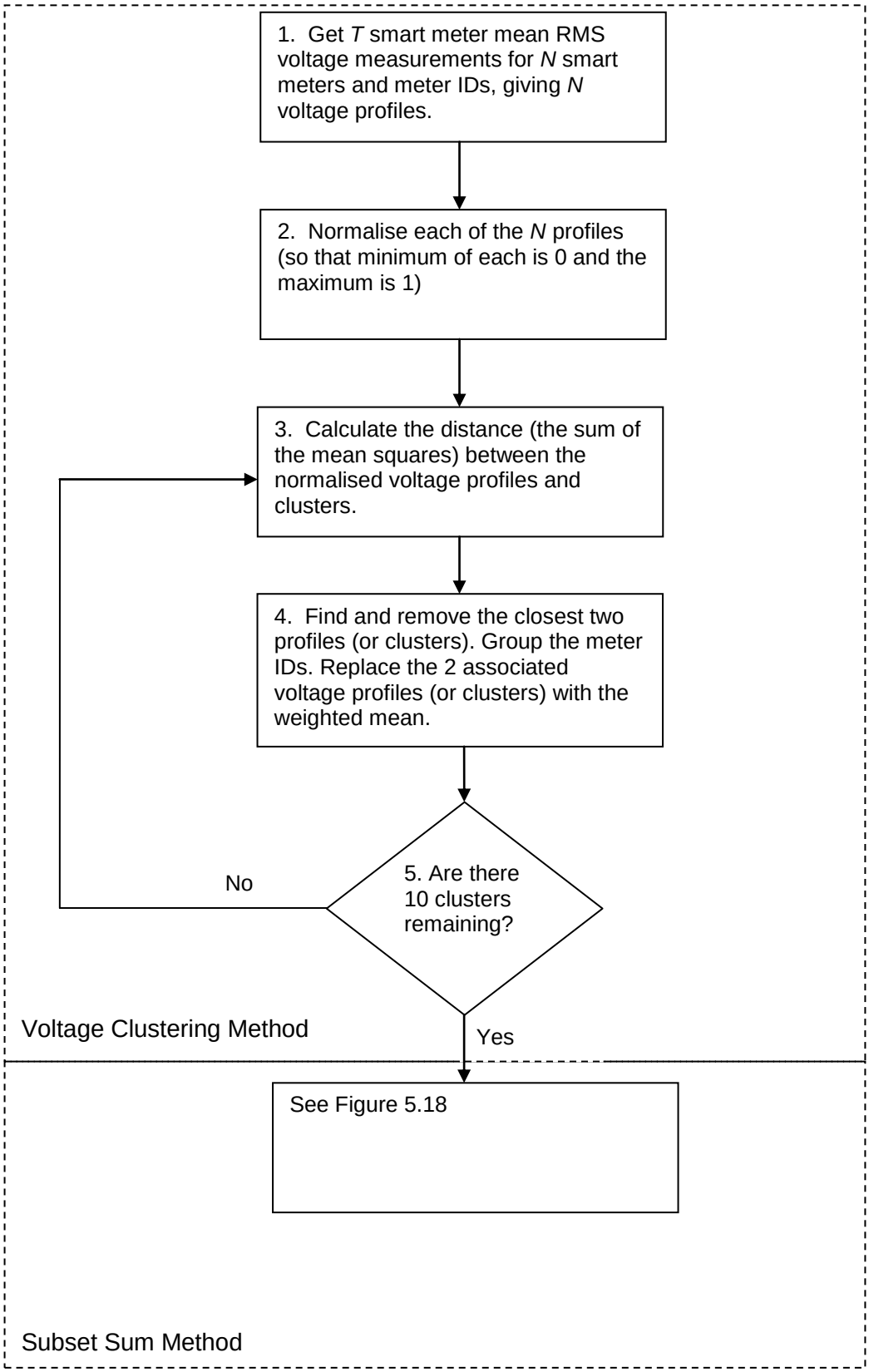


Figure 5.19 - The complete phase identification algorithm

5.6 A simple example of the subset sum method

Consider an example feeder with seven metered households is used. The meters are each given an identification number: {1},{2},{3},{4},{5},{6} and {7}. The calculated charge transfer for each is shown in Table 5.1. Assume that the subset sum method takes over when the meters have been reduced, using the voltage clustering method, to a set of 4 groups; {{1},{2},{3,4}{5,6,7}}, see Table 5.2. The source charge transfer values (Equation 5.6) are shown in Table 5.3.

Meter	Charge transfer [Ah] (3 significant figures)
1	38.2
2	32.5
3	109
4	42.3
5	96.6
6	78.4
7	134

Table 5.1 - Calculated charge transfer for each meter in the simple subset sum method example

Meter Groups	Charge transfer [Ah] (3 significant figures)
1	38.2
2	32.5
3,4	151
5,6,7	309

Table 5.2 - Calculated charge transfer for the known meter groupings in the simple subset sum method example

Phase	Measured charge transfer
1	70.7
2	151
3	309

Table 5.3 - Input (per phase) charge transfer for the simple subset sum method example.

The next step is to calculate per-phase charge transfer values for all possible ways of allocating the meter groups (the values in Table 5.2) across three phases. In this example, as there were 4 groupings initially, there are 81 ($=3^4$) possible configurations. For each the 81 possible configurations, these per-phase values are subtracted from the measured input charge transfer values. The absolute values of the subtractions were then summed. These steps are shown, for 8 of the possible phase configurations, in Table 5.4. The configuration with the lowest sum value, shown in bold in Table 5.4, is then used to select the phase configuration, in this example it is {1,2},{3,4} and {5,6,7}. Note that, in practice, due to the approximation of Equation 5.4, the lowest sum of absolute differences would be non-zero.

Configuration	Charge transfer from meter readings	Measured input charge transfer	Sum of absolute differences
{}{}{1,2,3,4,5,6,7}	{0}{0}{531}	{70.7,151,309}	70.7+151+309 =530.7
{}{1,2,3,4,5,6,7}{}	{0}{531}{0}	{70.7,151,309}	70.7+380+309 =759.7
{1,2,3,4,5,6,7}{}{}	{531}{0}{0}	{70.7,151,309}	460.3+151+309 =923
{2,3,4,5,6,7}{1}{}	{493}{38.2}{0}	{70.7,151,309}	422.3+112.8+309 =844.1
{2,3,4,5,6,7}{}{1}	{493}{0}{38.2}	{70.7,151,309}	422.3+151+270.8=844.1
…			
{1,2}{5,6,7}{3,4}	{70.7}{309}{151}	{70.7,151,309}	0+158+158 = 316
{1,2}{3,4}{5,6,7}	**{70.7}{151}{309}**	**{70.7,151,309}**	**0+0+0 = 0**
{3,4}{1,2}{5,6,7}	{151}{70.7}{309}	{70.7,151,309}	80.3+80.3+0 = 160.6
…			

Table 5.4 - Sum of absolute differences for 16 of the 81 possible ways of allocating the meter groups (Table 5.3) to three phases.

5.7 Testing of the complete phase identification algorithm

The complete phase identification algorithm, consisting of the voltage clustering method and the subset sum method (Figure 5.19, code in Appendix A6), was tested using the data (feeder input current and smart meter voltage and power profiles) from Chapter 4. The results, for a 30 minute averaging timeframe and 0% PV, are shown in Figure 5.20.

Compared with the results from the voltage clustering method alone, shown in Figure 5.12, the proportion of correctly identified meters has increased. Unlike Figure 5.12 there is no longer a group of points across the central region, indicating that the subset sum method succeeds where the voltage clustering method would otherwise have failed. The cases in which the algorithm failed to identify all meters were relatively few. The performance in those cases tended to worsen as the phase imbalance increased.

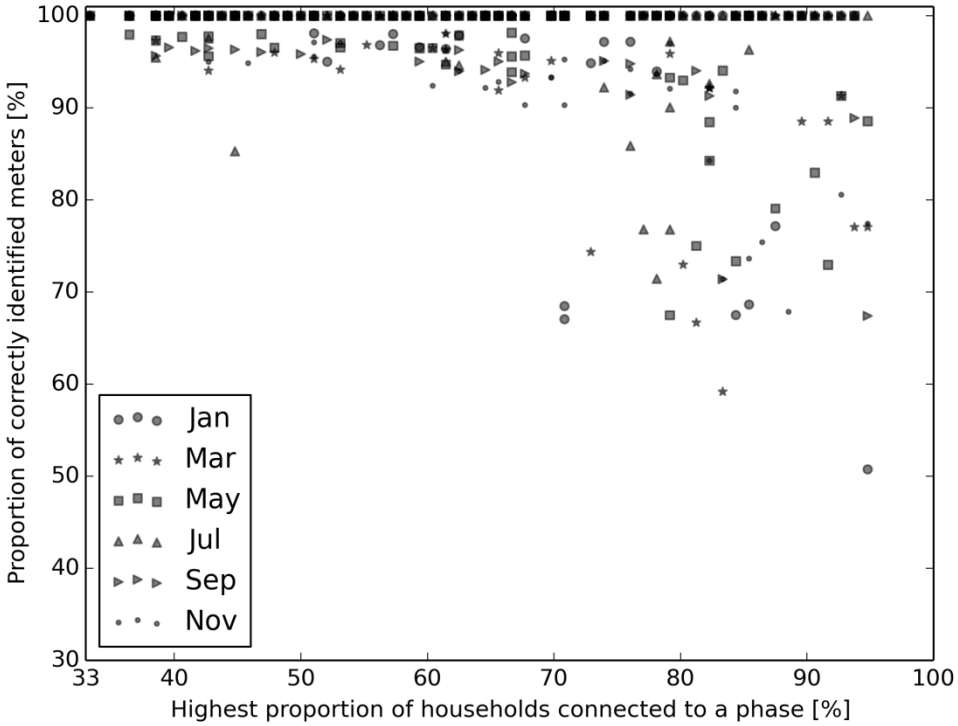


Figure 5.20 - Performance of the complete phase identification algorithm with varying connection imbalance.

Despite the overall improvement in performance, the subset sum method fails to correctly identify meters in some cases. These are represented by the points below 100 on the y-axis in Figure 5.20. The subset sum method fails due to error introduced in the approximation of Equation 5.4. This approximation gets less accurate as the variance of the voltage measurements, across the averaging timeframe, increases. This variance increases with imbalance, increasing the likelihood and severity of failure

as imbalance increases. If smart meters were able to directly measure and report accurate current profiles, the error would be removed. Alternatively, if only regions of the day where low variation of voltages are expected (e.g. overnight) this error will be reduced.

The improvement in performance is evident in comparison of the distribution of Figure 5.21 (the complete algorithm) with that of Figure 5.13 (the voltage clustering method alone). In Figure 5.20 shows that the algorithm tends to perform better as the averaging time frame is decreased.

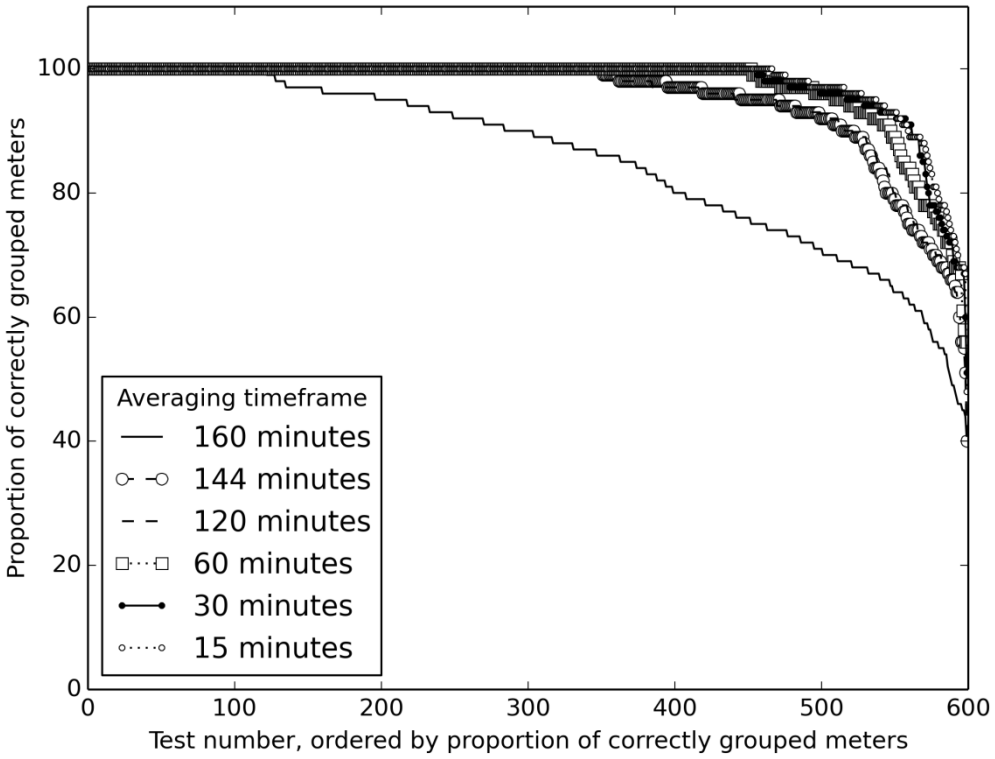


Figure 5.21 - Success factor distribution for the complete phase identification algorithm.

For the 30 minute averaging time, with all penetrations of PV, ~97% of meters were correctly identified. Table 5.5 compares the performance of the voltage clustering method alone with the complete algorithm, with all penetrations of PV included. There is a clear increase in the proportion of correctly identified meters.

Measurement Timeframe	% Successfully grouped meter phases – Voltage clustering method only	% Successfully identified meter phases – Combined voltage clustering and subset sum method
160 minutes	76.96	84.80
144 minutes	81.42	93.38
120 minutes	81.64	94.26
60 minutes	82.56	96.05
30 minutes	83.92	96.92
15 minutes	83.13	97.56

Table 5.5 - Comparison of voltage clustering and complete phase identification methods - with varying measurement timeframes .

The introduction of PV has a relatively slight effect on performance of the algorithm, see Figure 5.22. There is a small degradation in performance as the amount of PV increases. Table 5.6 shows the proportion of meters that were correctly identified across the PV penetrations and the measurement timeframes. It also shows a slight degradation in performance of the algorithm as the amount of PV increases, for all the averaging timeframes. However, overall, the effect is relatively insignificant.

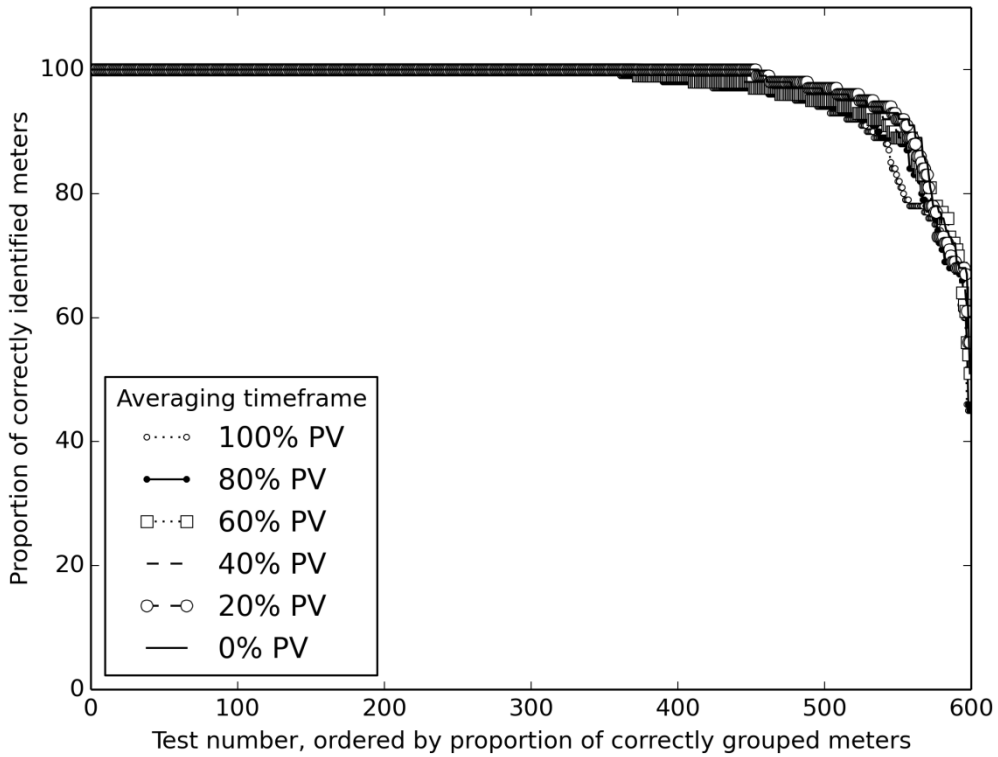


Figure 5.22 - Distribution of proportion of correctly grouped meters for 6 PV penetrations after testing the complete phase identification algorithm.

Measurement Timeframe	0% PV	20% PV	40% PV	60% PV	80% PV	100% PV
160 minutes	85.23	85.88	85.23	84.64	84.62	83.21
144 minutes	95.24	94.42	94.3	92.56	92.07	91.67
120 minutes	95.35	95.24	94.91	93.48	93.27	93.33
60 minutes	97.17	96.72	96.23	95.93	95.41	94.86
30 minutes	97.57	97.45	97.17	96.87	96.4	96.07
15 minutes	97.86	98.23	97.59	97.3	97.15	97.21

Table 5.6 - Percentage of correctly identified meter phase connections with varying measurement timeframe and PV .

Table 5.7 shows the performance of the algorithm across the 6 months (single days within the month). The intention was to give an impression of whether there is a month on which the phase algorithm performed better than others. The results are inconclusive. One might have expected the algorithm to perform better in the winter months as there would be lower PV output, however this cannot be clearly observed from the data. Therefore, no firm conclusion can be reached as to the best (i.e. where the algorithm is most likely to succeed) time of year to run the algorithm.

Measurement Timeframe	Jan	Mar	May	Jul	Sep	Nov
160 minutes	82.76	86.69	85.39	85.57	83.7	84.71
144 minutes	93.17	93.55	93.03	92.92	93.95	93.64
120 minutes	93.89	95.3	94.44	93.58	93.95	94.42
60 minutes	97.16	96.39	94.94	95.44	96.2	96.2
30 minutes	97.67	97.01	96.35	96.75	96.88	96.87
15 minutes	98.33	97.51	97.24	97.4	97.67	97.19

Table 5.7 - Percentage of correctly identified meter phase connections with varying measurement timeframe and month .

5.8 Discussion

The phase identification algorithm correctly identified the phase of ~97% of smart meters when half hourly measurements were used. The method compares well with Arya et al's [199] K-means voltage profile clustering method, which uses measurements at 5 minute intervals and reports that an accuracy of 93% could be achieved. The use of 30 minute interval is an advantage of the presented algorithm as fewer measurements are required. However, for direct comparison, the two methods (and others from the literature) should be tested on identical datasets.

The amount of PV on the feeder had only a slight effect on the performance of the algorithm. For the 30 minute averaging timeframe, the proportion of correctly identified meters dropped by 0.15% for every additional 10% of houses with 1.3kWp PV – a slight degradation in performance. This could be because PV output does not vary independently across the three phases. When the influence of PV is strong, it produces a tendency of the voltages across the three phases to vary in the same direction. As the algorithm relies on picking between differences in voltage variations, this is problematic. The problem could be resolved by pausing the algorithm during the middle part of the day and perhaps stitching together profiles from consecutive evenings, nights and mornings – thus avoiding the periods where the influence of PV is strongest. One other thing to note is that, in the data from Chapter 4, clear-sky PV was used. In reality, on many days, the PV profiles would be spiky – based on the cloud cover. Whilst the PV output connected to each phase is still likely to vary together, the degradation in algorithm performance, at high levels of PV, may lower than the results indicate.

The ~97% meter phase identification success rate implies that, if run on consecutive days, with the most commonly predicted outcome selected, the success rate could be increased further. The meters which cannot be confidently identified, with contradictions in their allocation on different days, could be identified for further analysis. This approach might also work for the voltage clustering method alone, which, when tested on its own, had a success rate of ~84%.

The optimal point (the one that maximises the success of the algorithm) at which the algorithm should transfer between the voltage variation method and the subset sum was selected at 87 iterations in the presented algorithm (see Figure 5.16). This was chosen based on the point of failure indicated by the chosen feeder and dataset. Where feeders of different length, with more houses, are used, this point of failure (the clustering iteration at which the algorithm first incorrectly identifies the phase of a meter or group of meters) might change. The algorithm would possibly need to be adapted to cope with variation in feeder impedance and number of connected households.

A tendency of the presented algorithm's performance to worsen as connection imbalance (the highest number of dwellings connected to a phase) was observed. This means that, if real feeders have low levels of connection imbalance (<50% dwellings on one phase) then the performance of the algorithm, when used on a population of feeders, would be greater than 97%.

When the subset sum part of the algorithm fails, it is due to error in approximation of the current. This error could be eliminated through recording the current directly, instead of relying on mean Voltage and Power measurements. The GB smart metering specifications do not require that current be recorded. However, the meters must use current measurements to accurately calculate the mean active power and reactive power. If the meters made to be capable of storing and reporting accurate half-hourly current profiles, the phase connection of smart meters could be correctly identified in approaching 100% of cases.

6 Conclusions and Further Work

6.1 Conclusions

6.1.1 Modelling of an Unbalanced Low Voltage Feeder

The LV feeder model uses Berg, Hawkins and Pleines' [93] unbalanced load flow technique which is based on network reduction and re-expansion with nodal analysis. The feeder model uses a TNS earthing arrangement; this was shown to be equivalent to TN-C-S in normal operation, allowing for simpler modelling. A metric for connection imbalance was introduced – the highest proportion of houses connected to any phase. The model is capable of varying connection imbalance by changing the phase to which each house is connected.

The implemented model was validated using PSCAD. The maximum percentage error in load voltages, for fixed PQ loads in a 60-load test model and an equivalent PSCAD model, was 0.0181% for an unbalanced phase configuration with load magnitudes sampled from a gamma distribution. The maximum absolute percentage error in the resulting calculated P and Q values was 0.0462%. The errors are small enough to allow the model's use in investigation of connection imbalance.

As part of the work it was shown that, for normal operation, T-N-CS earthing arrangement is equivalent to TNS. This is a useful conclusion as TNS is simpler to model but T-N-CS is the more common earthing arrangement on British Low Voltage distribution networks.

As the number of houses connected to the feeder was increased, a consistent pattern in voltage minima vs connection imbalance was observed. This suggests that, reasonable predictions about the voltage minima could have been made using only the single phase and balance three phase cases – assuming that houses are evenly spread and that the cable impedance is known.

The modelling method is well suited for use with stochastic demand modelling techniques. It can be run many thousands of times given adequate computing resources. When coupled with stochastic demand modelling techniques, the program made possible the creation of smart meter voltage profiles. This enabled examination of distributed generation hosting capacity, with varied connection imbalance, at low voltage level (Chapter 4) and study of phase identification techniques (Chapter 5).

6.1.2 The influence of solar photovoltaics and connection imbalance on an LV feeder

The connection imbalance was varied by randomly allocating houses to different phases. Demand profiles were created stochastically and PV generation was added to a varied proportion of houses (0 to 100% in 10% steps), based on Richardson et al [217]. More than 19 million unbalanced load flow calculations were performed using a supercomputer, see section 1.3.

Connection imbalance does not cause voltage excursions on a typical urban feeder until it becomes relatively extreme, >~75% houses on one phase. This limit drops to ~60% if all households are given 1.3kWp PV.

The addition of PV, above 40% of households, exacerbates the effect of increasing connection imbalance – so, for example, connection imbalance on a feeder with 100% PV has a bigger effect on the annual meter voltage range than a feeder with 40% PV. Adding non-concurrent load or generation and increasing the cable impedance would further lower the amount of allowable connection imbalance (before voltage excursions occur).

With an increasing PV penetration, the losses first decrease and then (at above 70% PV) start to increase again as the feeder starts to export power. The line and neutral currents did not exceed the cable rating at any point in the simulations.

The hours of 0000-0600hrs can be ignored when assessing voltage levels and PV, this would reduce the computing time required by ~25%.

6.1.3 Using Smart Meters for Phase Identification

An algorithm was developed that identifies the phase connection of smart meters on a set of test feeders. The algorithm combines two types of phase identification algorithm found in the literature - voltage measurement clustering and subset sum formulation techniques. It uses smart meter voltage profiles and active power profiles with current measured at the supply substation. It correctly predicted the phase connection for ~97% of smart meters, using simulated data representing a set 100 different connection configurations, across 6 different days (different sets of demand profiles) with a measurement averaging timeframe of 30 minutes.

On their own, the voltage clustering method and the subset sum method did not adequately solve the phase identification problem. The voltage clustering was shown to perform well until the final 10 clusters are formed – whereupon it had a significant probability of failure. The subset sum method is limited due to its poor scalability – the time taken scales as 3^N, where N is the number of meters. In combining the two methods, both of their shortcomings were overcome.

6.2 Further Work

6.2.1 Modelling of an LV feeder with connection imbalance

Improvements that could be made to the model include the integration with an unbalanced 11kV network solver. This would allow the input slack voltages to be more realistically modelled. Some form of iterative process would be required. For example, running an 11kV unbalanced load flow based on the per phase substation P and Q totals to get initial values for the LV slack voltages and then running the unbalanced LV load flow for each substation feeder and calculating the losses. The 11kV unbalanced load flow would then be re-run with the losses included and the process repeated until convergence is reached. Some amendments would have to be made to the algorithm to allow for modelling of delta connected loads, as would be common at the 11kV level. Preliminary scripts have been written and run to do this.

Improvements could also be made to the load modelling. At present the load must be an admittance, calculated from a given impedance or power and voltage. The model could be made more detailed by allowing for the inclusion of ZIP loads – that is loads that include some element of fixed impedance, fixed current and fixed power. Using the work of Tsagarakis, Collin and Kiprakis [221] would be a good starting point.

6.2.2 Variation of connection imbalance and PV on an LV feeder

A useful direction for this work in future would be to apply it to a larger population of feeder types (e.g. with differing impedances and number of houses) and number of profiles. Strbac et al [25] observed that, “reinforcements in urban areas are primarily driven by thermal overloads, while for semi-urban/rural and rural areas this is mostly due to excessive voltage drops”. However, the analysis in this chapter implies an urban feeder constrained by voltage drops (and rises). This emphasises the importance of analysing the effect of imbalance on other feeders, include some on which hosting capacity is limited by thermal constraints.

A direction of future development could be in the area of active control, to demonstrate an increase in the load and generation hosting capacity of a feeder. Given that the modelling method includes appliance-by-appliance modelling combined with unbalanced load flow, there is scope to include demand response and home energy storage schemes as well as substation based apparatus such as automated tap changers or batteries. A modelling mechanism to handle communication between smart meters and the DNO could be straightforwardly added. An active distribution network model of this type would be a useful for comparison of active control schemes at low voltage level.

An aim for expanding for this work could be in the estimation of cable parameters based on smart meter readings. If this could be achieved, an automated script gradually working out the impedances of the Low Voltage system would be a valuable outcome. Once this has been arrived at, the hosting capacity assessment of the type used in this thesis could be utilised using smart metering data.

6.2.3 Phase identification using smart meters

A priority for future work is the testing of the phase identification algorithm on a real feeder. Whilst the algorithm has been shown to work well, there are a number of further factors which must be taken into consideration. The first of these is the synchronisation between the voltage profiles and the power profiles. I.e. in the case of half hourly profiles, the 30 minutes over which the mean voltage is calculated must coincide with the 30 minutes over which the mean active and reactive power is calculated. How precisely this will be able to be done in practice is uncertain. Also, the possibility of meter error (i.e. measurement error) was not examined. A test using real recorded data would help in assessment of these factors.

There are likely to be missing measurements on any given feeder; at least until the end of 2020, when the GB smart metering roll out is expected to complete. Therefore a valuable direction of improvement for the algorithm is in allowing for missing or unavailable readings. The voltage clustering method can still be applied if some households do not have smart meters. The subset sum method would, however, need to be modified. This could perhaps be done by looking at the difference between the total input charge (across all three phases) and the sum of the available smart meter measurements. The comparison of totals in this way might also be useful in identifying cases of electricity theft.

The optimal point (the one that maximises the success of the algorithm) at which the algorithm should transfer between the voltage clustering method and the subset sum method should be investigated further. It may be that, if adequate computing resource were available, the subset sum method could be started earlier, perhaps improving the performance of the algorithm.

A way of improving the algorithm's performance would be to run the subset sum method multiple times. If it was run based on, say, half hourly data then 48 predicted phase configuration results would be obtained. The algorithm would then pick the most frequently predicated configuration. This would improve on the implemented algorithm – in which meter groupings with a relatively low transferred charge value risk being misattributed to the wrong phase.

A final consideration is the potential for using total timeframes of longer than 24 hours. The 24 hour period was chosen due to the limitation in the data available from Chapter 4. In reality no such limitation exists. The algorithm could use data over weeks or months, reducing the occurrence of no-load houses. It may also be that, when longer timeframes are considered, the voltage variation method alone would be successful – negating the need to perform measurements at the substation. This, therefore, is a potentially valuable area of future research.

6.2.4 Supercomputing

There is potential value to DNOs in making use of supercomputer/cluster-computing facilities similar to the one, Raven [10], utilised in this thesis. Such facilities could allow DNOs to improve the quality of their operational and predictive models. The increased computing capacity will allow the use of stochastic network models (e.g. for demand, generation, weather, traffic etc), influenced by historic and real time data. This will lead to a more intimate understanding of the networks, increased asset utilisation and improved investment decisions.

In making use of such facilities, the following research areas should be considered:

- Application, of knowledge and understanding of handling large data flows for system operation and planning, from other industries (e.g. aviation, web based social-media, financial computing systems etc).
- Improvement of modelling techniques; How to best optimise modelling across timescales (μs to decades).
- Use of new information streams; how to best accommodate information flows, that influence/predict network operation) from outside sources (e.g. weather, traffic, social media, events).
- Multi vector modelling; combination of electricity network models with other energy vectors (e.g. gas, heat).

References

[1] DECC, “Smart Metering Implementation Programme: Third annual report on the roll-out of smart meters,” 2014.

[2] National Grid, “Electricity Ten Year Statement,” 2014.

[3] The IET Power Network Joint Vision (PNJV) Expert Group, “Electricity Networks Handling a Shock to the System - IET position statement and technical report,” 2013. [Online]. Available: http://mycommunity.theiet.org/energy/pnjv. [Accessed: 17-Jan-2013].

[4] J. Ekanayake, N. Jenkins, K. Liyanage, J. Wu, and A. Yokoyama, *Smart Grid: Technology and Applications*. John Wiley & Sons, 2012.

[5] ENA, “Smarter Networks Portal.” [Online]. Available: http://www.smarternetworks.org/. [Accessed: 18-Oct-2013].

[6] L. de Alvaro Garcia, G. A. Taylor, D. C. H. Wallom, G. Gershinsky, A. Yunta Huete, and K. Diwold, “High Performance Computing and Communcations Technology Solutions for Future Smart Distribution Network Operation,” in *CIRED 22nd International Conference on Electricity Distribution Stockholm , 10-13 June*, 2013, pp. 10–13.

[7] R. C. Green, L. Wang, and M. Alam, “Applications and trends of high performance computing for electric power systems: Focusing on smart grid,” *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 922–931, 2013.

[8] L. J. Thomas, “Private conversation with a Distribution Network Operator representative at HubNet Symosium, Cardiff, September 2013.” .

[9] DECC, “Statistical data set Weekly solar PV installation & capacity based on registration date.” [Online]. Available: https://www.gov.uk/government/statistical-data-sets/weekly-solar-pv-installation-and-capacity-based-on-registration-date. [Accessed: 02-Jun-2015].

[10] ARCCA, “Advanced Research Computing @ Cardiff - Introduction to Raven.” [Online]. Available: http://www.cardiff.ac.uk/arcca/services/equipment/ravenintroduction.html. [Accessed: 27-Nov-2015].

[11] OFGEM, “Transmission Networks Overview.” [Online]. Available: http://www.ofgem.gov.uk/Networks/Trans/Pages/trans.aspx. [Accessed: 14-Nov-2015].

[12] ENA, “Issue 19 of the Distribution Code and the Guide to the Distribution Code of Licensed Distribution Network Operators of Great Britain,” Dec. 2012.

[13] J. Ward, M. Pooley, and (Sustainability First), “GB Electricity Demand – Context and 2010 Baseline Data,” 2011.

[14] ENA, “Distribution Charges Overview.” [Online]. Available: http://www.energynetworks.org/electricity/regulation/distribution-charging/distribution-charges-overview.html. [Accessed: 14-Nov-2015].

[15] OFGEM, “GB Electricity Transmission Network.” [Online]. Available: https://www.ofgem.gov.uk/electricity/transmission-networks/gb-electricity-transmission-network. [Accessed: 06-Nov-2015].

[16] Elexon, “Balancing Mechanism Reporting System Website - What is the Balancing Mechanism?” [Online]. Available: http://www.bmreports.com/bwx_help.htm#BM. [Accessed: 06-Nov-2015].

[17] Elexon, “Balancing Mechanism Reporting System Website - What is NETA?” [Online]. Available: http://www.bmreports.com/bwx_help.htm#whatIsNETA. [Accessed: 06-Nov-2015].

[18] National Grid, *GB Seven Year Statement - The Market Structure*. 2006.

[19] L. Thomas and N. Jenkins, “HubNet Position Paper Series - Smart Metering for the development and operation of the GB Power System,” 2014. [Online]. Available: http://orca.cf.ac.uk/68085/. [Accessed: 14-Nov-2015].

[20] D. Openshaw, “Smart metering - an energy networks perspective,” *IET Seminar on Smart Metering - Gizmo or Revolutionary Technology*. pp. 1–16, 2008.

[21] E. Lakervi and E. J. Holmes, *Electricity distribution network design*, 2nd ed. IET, 1995.

[22] Department of Trade and Industry, *The Electricity Safety, Quality and Continuity Regulations 2002*, no. October. 2002, pp. 1–47.

[23] R. P. Leeuwerke, A. L. Brayford, A. Robinson, and C. Tobias, “Developments in ring main unit design for improved MV network performance,” *IET Power Eng. J.*, vol. 14, no. 6, pp. 270 – 277, 2000.

[24] M. H. Bollen and F. Hassan, *Integration of Distributed Generation in the Power System*. John Wiley & Sons, 2011.

[25] G. Strbac, C. K. Gan, M. Aunedi, V. Stanojevic, P. Djapic, J. Dejvises, P. Mancarella, A. Hawkes, D. Pudjianto, D. Openshaw, S. Burns, P. West, D. Brogden, A. Creighton, and A. Claxton, “Benefits of Advanced Smart Metering for Demand Response based Control of Distribution Networks,” Apr-2010. [Online]. Available:

http://www.energynetworks.org/modx/assets/files/electricity/futures/smart_meters/Smart_Metering_Benerfits_Summary_ENASEDGImperial_100409.pdf. [Accessed: 14-Nov-2015].

[26] R. Silversildes, T. Green, and T. Luth, “HubNet Position Paper Series - Power Electronics in Distribution System Management - Version 0.6,” 2014. [Online]. Available: http://hubnet.org.uk/filebyid/633/PE_Distribution.pdf.

[27] N. Jenkins, J. B. Ekanayake, and G. Strbac, *Distributed Generation*. Institution of Engineering and Technology, 2010.

[28] Western Power Distribution, “Network Equilibrium Project - Low Carbon Network Fund Tier 2 Bid Document,” WPDT206/2, 2014.

[29] B. Hayes, I. Hernando-gil, A. Collin, G. Harrison, and S. Djoki, “Optimal Power Flow for Maximizing Network Benefits From Demand-Side Management,” *IEEE Trans. Power Syst.*, vol. 29, no. 4, pp. 1739–1747, 2014.

[30] N. Venkatesan, J. Solanki, and S. K. Solanki, “Demand response model and its effects on voltage profile of a distribution system,” in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–7.

[31] K. B. Samarakoon, “Use of Smart Meters for Frequency and Voltage Control - PhD Thesis - Chapter 5,” Cardiff University, 2012.

[32] L. J. Thomas, J. Wu, J. B. Ekanayake, and N. Jenkins, “Enabling distributed frequency response using smart meters,” in *3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, 2012, pp. 1–5.

[33] J. M. Bloemink and T. C. Green, “Increasing distributed generation penetration using soft normally-open points,” *IEEE PES Gen. Meet. PES 2010*, pp. 1–8, 2010.

[34] P. Wolfs, M. A. S. Masoum, and X. Su, “Comprehensive optimal photovoltaic inverter control strategy in unbalanced three-phase four-wire low voltage distribution networks,” *IET Gener. Transm. Distrib.*, vol. 8, no. 11, pp. 1848–1859, 2014.

[35] F. Shahnia, A. Ghosh, G. Ledwich, and F. Zare, “Voltage unbalance improvement in low voltage residential feeders with rooftop PVs using custom power devices,” *Int. J. Electr. Power Energy Syst.*, vol. 55, pp. 362–377, 2014.

[36] S. Bala, D. Das, E. Aeloiza, A. Maitra, and S. Rajagopalan, “Hybrid distribution transformer: Concept development and field demonstration,” *2012 IEEE Energy Convers. Congr. Expo.*, pp. 4061–4068, 2012.

[37] R. Silversides and T. Green, “A High Density Converter for Mid Feeder Voltage Regulation of Low Voltage Distribution Feeders,” pp. 1972–1978, 2014.

[38] B. O. Brewin, S. C. E. Jupe, M. G. Bartlett, K. T. Jackson, and C. Hanmer, “New technologies for low voltage distribution networks,” *Innov. Smart Grid Technol. (ISGT Eur. 2011 2nd IEEE PES Int. Conf. Exhib.*, pp. 1–8, 2011.

[39] M. H. J. Bollen and A. Sannino, *Power Engineering Letters - Voltage Control With Inverter-Based Distributed Generation*, vol. 20, no. 1. 2005, pp. 519–520.

[40] P. N. Vovos, A. E. Kiprakis, A. R. Wallace, and G. P. Harrison, “Centralized and Distributed Voltage Control: Impact on Distributed Generation Penetration,” *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 476–483, Feb. 2007.

[41] T. Sansawatt, L. F. Ochoa, and G. P. Harrison, “Smart decentralized control of DG for voltage and thermal constraint management,” *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1637–1645, 2012.

[42] C. M. Hird, H. Leite, N. Jenkins, and H. Li, “Network voltage controller for distributed generation,” *Distribution*, vol. 151, no. 2, 2004.

[43] J. O’Donnell, “Voltage Management of Networks with Distributed Generation,” The University of Edinburgh, 2007.

[44] M. Fila, D. Reid, P. Lang, J. Hiscock, and G. A. Taylor, “Flexible Voltage Control in Distribution Networks with Distributed Generation - Modelling Analysis and Field Trial Comparison,” in *CIRED 20th International Conference on Electricity Distribution*, 2009, pp. 8–11.

[45] A. Kulmala, S. Repo, and P. Järventausta, “Coordinated Voltage Control in Distribution Networks Including Several Distributed Energy Resources,” *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1–11, 2014.

[46] D. J. Rogers, T. C. Green, and R. W. Silversides, “A Low-Wear Onload Tap Changer Diverter Switch for Frequent Voltage Control on Distribution Networks,” *Power Deliv. IEEE Trans.*, vol. 29, no. 2, pp. 860–869, 2014.

[47] D. Roberts, J. Hiscock, and A. Creighton, “Increasing voltage headroom for the connection of Distributed Generation - Site trials of the SuperTAPP N+ AVC relay,” *Electr. Distrib. - Part 1, 2009. CIRED 2009. 20th Int. Conf. Exhib.*, no. 0468, pp. 8–11, 2009.

[48] F. A. Viawan and D. Karlsson, “Coordinated voltage and reactive power control in the presence of distributed generation,” in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008, pp. 1–6.

[49] G. Celli, M. Loddo, F. Pilo, and A. Abur, “On-line network reconfiguration for loss reduction in distribution networks with Distributed Generation,” in *Electricity Distribution, 2005. CIRED 2005. 18th International Conference and Exhibition*

on, 2005, no. June, pp. 1–4.

[50] A. T. Procopiou, C. Long, and L. F. Ochoa, “Voltage control in LV networks: An initial investigation,” *IEEE PES Innov. Smart Grid Technol. Eur.*, pp. 1–6, 2014.

[51] A. Navarro-Espinosa, L. F. Ochoa, and D. Randles, “Assessing the benefits of meshed operation of LV feeders with low carbon technologies,” *2014 IEEE PES Innov. Smart Grid Technol. Conf. ISGT 2014*, no. Mv, pp. 2011–2014, 2014.

[52] H. L. Willis, “Energy storage opportunities related to distributed generation,” *2000 Power Eng. Soc. Summer Meet. (Cat. No.00CH37134)*, vol. 3, no. c, pp. 1517–1518, 2000.

[53] A. K. Srivastava, A. A. Kumar, and N. N. Schulz, “Impact of distributed generations with energy storage devices on the electric grid,” *IEEE Syst. J.*, vol. 6, no. 1, pp. 110–117, 2012.

[54] A. Keane, L. F. Ochoa, E. Vittal, C. J. Dent, and G. P. Harrison, “Enhanced Utilization of Voltage Control Resources With Distributed Generation,” *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 252–260, Feb. 2011.

[55] J. McDonald, “Adaptive intelligent power systems: Active distribution networks,” *Energy Policy*, vol. 36, no. 12, pp. 4346–4351, Dec. 2008.

[56] C. D’Adamo, S. Jupe, and C. Abbey, “Global Survey on Planning and Operation of Active Distribution Networks - Upadate of CIGRE C6.11 Working Group Activites,” in *20th International Conferencce on Electricity Distribution*, 2009.

[57] A. Carter, B. Marshall, V. Hamidi, and M. M. Osborne, “The impact of new technology on network resilience,” *10th IET Int. Conf. AC DC Power Transm. (ACDC 2012)*, pp. 52–52, 2012.

[58] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, “Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges,” *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1743–1752, Nov. 2007.

[59] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, “Multi-Agent Systems for Power Engineering Applications—Part II: Technologies, Standards, and Tools for Building Multi-agent Systems,” *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1753–1759, Nov. 2007.

[60] I. Pisica, C. J. Axon, P. R. Hobson, and G. A. Taylor, “A Multi-Agent Model for Assessing Electricity Tariffs,” in *5th IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), October 12-15, Istanbul*, 2014, pp. 1–6.

[61] P. Pachanapan, O. Anaya-Lara, and K. Lo, “Agent-based control for power quality enhancement in highly distributed generation networks,” in *Universities Power Engineering Conference (UPEC), 2009 Proceedings of the 44th International*, 2009, p. 5.

[62] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” *Computer (Long. Beach. Calif).*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[63] S. McArthur, P. Taylor, G. Ault, J. King, D. Athanasiadis, V. Alimisis, and M. Czaplewski, “The Autonomic Power System-Network operation and control beyond smart grids,” in *3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), Berlin*, 2012.

[64] Southampton Unversity, “Complexity Science Focus.” [Online]. Available: http://www.complexity.ecs.soton.ac.uk/. [Accessed: 14-Nov-2015].

[65] A. H. Bond and L. Gasser, “A Survey of Distributed Artificial Intelligence,” Aug-1988. [Online]. Available: http://petr.olmer.cz/vyuka/agenti/BondGasser.pdf.

[66] N. M. Avouris, *Distributed Artificial Intelligence: Theory and Praxis*, vol. 1992. Springer, 1992.

[67] D. Pountain, *The new Penguin dictionary of computing*. Penguin, 2001.

[68] K. G. Binmore, *Fun and Games: A Text on Game Theory*. D.C. Heath, 1992.

[69] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[70] Semantic Computing Foundation, “Archived Website - Institute of Semantic Computing.” [Online]. Available: http://web.archive.org/web/20121127165356/http://www.isc-home.org/. [Accessed: 14-Nov-2015].

[71] IEEE, “IEEE Smart Grid Website.” [Online]. Available: http://smartgrid.ieee.org/ieee-smart-grid. [Accessed: 30-Jun-2014].

[72] DECC, “Government Response to the Consultation on New Smart Energy Code Content (Stage 2),” 2014. [Online]. Available: https://www.gov.uk/government/consultations/new-smart-energy-code-content-stage-2. [Accessed: 24-Jun-2014].

[73] DECC, “Explanatory document to support the designation of the first version of the Smart Metering Equipment Technical Specifications (SMETS1),” 2012. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/65685/7339-exp-doc-support-smets1.pdf. [Accessed: 24-Jun-2014].

[74] DECC, “Smart Meters , Smart Data , Smart Growth,” 2015. [Online]. Available: https://www.gov.uk/government/publications/smart-meters-smart-data-smart-growth. [Accessed: 14-Nov-2015].

[75] DECC, “A Consultation on New Smart Energy Code Content (Stage 2),” 2013. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/251280/A_Consultation_on_New_Smart_Energy_Code_Content_-_SEC2.pdf. [Accessed: 14-Nov-2015].

[76] DECC, “Smart Metering Implementation Programme Smart Metering Equipment Technical Specifications Version 2,” Jan-2013. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/68898/smart_meters_equipment_technical_spec_version_2.pdf. [Accessed: 17-Jan-2014].

[77] F. Lienert, “DECC Smart metering Impact Assessment,” 2013. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/78666/IA-Feb.pdf. [Accessed: 14-Nov-2015].

[78] J. Boraston, A. Manson, T. Hainey, D. Openshaw, and R. Reekie, “ENA Smart Metering System Use Cases,” 2010. [Online]. Available: http://www.energynetworks.org/modx/assets/files/electricity/futures/smart_meters/ENACR007_002_1-1_UseCases_100412.pdf. [Accessed: 14-Nov-2015].

[79] L. Thomas, A. Burchill, K. Samarakoon, Y. He, J. Wu, J. Ekanayake, and N. Jenkins, “Control of electricity networks using smart meter data,” in *CIGRE Session*, 2012.

[80] Western Power Distribution, “Innovation Funding Report - Regulatory Report 2012/2013,” 2012.

[81] S. Tucker and R. Weller, “UK Power Networks - Engineering Design Standard EDS 06-0016 - LV Network Earthing Design,” 2012.

[82] Institution of Engineering and Technology and BSI, *BS7671:2008(2011) Requirements for Electrical Installations: IET Wiring Regulations*. Institution of Engineering and Technology and BSI, 2011.

[83] C. Brown and A. Graham, “SP Energy Networks Low Voltage Earthing Policy and Application Guide - EART-01-002,” 3, Mar. 2009.

[84] Northern Powergrid, “Northern Powergrid - IMP/010/011 - Code of Practice for Earthing LV Networks and HV Distribution Substations,” Aug. 2012.

[85] W. V. Lyon, *Applications of the Method of Symmetrical Components*. McGraw-Hill, 1937.

[86] L. G. Stokvis, “Analysis of unbalanced three-phase systems (not accessed),” *Electr. world*, vol. 65, pp. 1111–1115, 1915.

[87] C. L. Fortescue, “Method of Symmetrical Co-Ordinates Applied to the Solution of Polyphase Networks,” *Trans. Am. Inst. Electr. Eng.*, vol. XXXVII, no. 2, 1918.

[88] M. Bollen, “Definitions of Voltage Unbalance,” vol. 936, no. November, pp. 49–50, 2002.

[89] P. Pillay and M. Manyage, “Definitions of voltage unbalance,” *IEEE Power Eng. Rev.*, vol. 21, no. 5, pp. 50–51, 2001.

[90] Energy Networks Association, “Engineering Recommendation P29: Planning Limits for Voltage Unbalance in the United Kingdom (Not accessed).”

[91] P. Trichakis, P. C. Taylor, P. F. Lyons, and R. Hair, “Predicting the technical impacts of high levels of small-scale embedded generators on low-voltage networks,” *IET Renew. Power Gener.*, vol. 2, no. 4, p. 249, 2008.

[92] R. D. Zimmerman, “Comphrehensive distribution power flow: modeling, formulation, solution algorithms and analysis - PhD Thesis,” Cornell University, 1995.

[93] R. Berg, E. Hawkins, and W. Pleines, “Mechanized Calculation of Unbalanced Load Flow on Radial Distribution Circuits,” *IEEE Trans. Power Appar. Syst.*, vol. PAS-86, no. 4, pp. 415–421, Apr. 1967.

[94] C. S. Cheng and D. Shirmohammadi, “A three-phase power flow method for real-time distribution system analysis,” *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 671–679, May 1995.

[95] D. Shirmohammadi, H. W. Hong, A. Semlyen, and G. X. Luo, “A compensation-based power flow method for weakly meshed distribution and transmission networks,” *IEEE Trans. Power Syst.*, vol. 3, no. 2, pp. 753–762, May 1988.

[96] W. H. Kersting and W. H. Phillips, “A Radial Three-phase Power Flow Program for the PC [Unavailable],” in *Proceedings of Frontiers Power Conference, Stillwater*, 1987.

[97] G. X. Luo and a. Semlyen, “Efficient load flow for large weakly meshed networks,” *IEEE Trans. Power Syst.*, vol. 5, no. 4, pp. 1309–1316, 1990.

[98] D. Rajicic, R. Ackovski, and R. Taleski, “Voltage correction power flow,” *Power Deliv. IEEE …*, vol. 9, no. 2, pp. 1056–1062, 1994.

[99] H. Chiang, “A Decoupled Load Flow Method for Distribution Power Networks: Algorithms, Analysis and Convergence Study [Not accessed],” *Electr. Power Energy Syst.*, vol. 13, no. 3, 1991.

[100] W. H. Kersting and D. . Mendive, “An Application of Ladder Network Theory to the Solution of Three-Phase Radial Load-Flow Problems [Not available],” in *IEEE/PES 1976 Winter Meeting, New York*.

[101] R. D. Zimmerman, “Fast decoupled power flow for unbalanced radial distribution systems,” *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 2045–2052, Nov. 1995.

[102] A. Broadwater, A. Chandrasekaran, C. Huddleston, and A. Khan, “Power Flow Analysis of Unbalanced Multiphase Radial Distribution Systems,” *Electr. Power Syst. Res.*, vol. 14, no. 1, pp. 23–33, 1988.

[103] J. Carson, “Wave propagation in overhead wires with ground return,” *Bell Syst. Tech. J.*, vol. xli, pp. 539–554, 1926.

[104] W. H. Kersting and R. K. Green, “The application of Carson’s equation to the steady-state analysis of distribution feeders,” *2011 IEEE/PES Power Syst. Conf. Expo. PSCE 2011*, pp. 1–6, 2011.

[105] G. Kron, “Tensorial Analysis of Integrated Trasmission Systems,” *AIEE Trasactions*, vol. 70, pp. 1239–1248, 1951.

[106] W. H. Kersting, *Distribution System Modeling and Analysis*. CRC Press, 2001.

[107] M. Srinivas, “Distribution load flows: a brief review,” *Power Eng. Soc. Winter Meet. 2000. …*, vol. 00, no. c, pp. 942–945, 2000.

[108] L. Ochoa, R. Ciric, A. Padilha-Feltrin, and G. Harrison, “Evaluation of distribution system losses due to load unbalance,” *Power Syst. Comput. Conf. (PSCC), Liège, Belgium, 2005*, no. August, p. 4, 2005.

[109] J.-H. Teng, “A direct approach for distribution system load flow solutions,” *IEEE Trans. Power Deliv.*, vol. 18, no. 3, pp. 882–887, Jul. 2003.

[110] J. Teng, “Modelling distributed generations in three-phase distribution load flow,” *IET Gener. Transm. Distrib.*, no. April 2007, 2007.

[111] J. Teng and C. Chang, “A Novel and Fast Three-Phase Load Flow for,” *IEEE Trans. Power Syst.*, vol. 17, no. 4, pp. 1238–1244, 2002.

[112] W.-M. Lin and J.-H. Teng, “Three-phase distribution network fast-decoupled power flow solutions,” *Int. J. Electr. Power Energy Syst.*, vol. 22, no. 5, pp. 375–380, Jun. 2000.

[113] V. M. da Costa, M. L. de Oliveira, and M. R. Guedes, “Developments in the analysis of unbalanced three-phase power flow solutions,” *Int. J. Electr. Power Energy Syst.*, vol. 29, no. 2, pp. 175–182, Feb. 2007.

[114] P. Garcia, J. Pereira, S. Carneiro Jr, V. M. da Costa, and N. Martins, “Three-phase power flow calculations using the current injection method,” *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 508–514, 2000.

[115] L. Araujo, D. Penido, S. Carneiro Jr, J. Pereira, and P. Garcia, “A comparative study on the performance of TCIM full Newton versus backward-forward power flow methods for large distribution systems,” *Power Syst. …*, pp. 522–526, 2006.

[116] D. Penido, L. de Araujo, S. Carneiro Jr, J. Pereira, and P. Garcia, “Three-phase power flow based on four-conductor current injection method for unbalanced distribution networks,” *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 494–503, 2008.

[117] S. Khushalani and N. Schulz, “Unbalanced Distribution Power Flow with Distributed Generation,” in *2005/2006 PES TD*, pp. 301–306.

[118] S. Khushalani, J. M. Solanki, and N. N. Schulz, “Development of Three-Phase Unbalanced Power Flow Using PV and PQ Models for Distributed Generation and Study of the Impact of DG Models,” *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1019–1025, Aug. 2007.

[119] S. Segura, L. C. P. da Silva, and R. Romero, “Generalised single-equation load flow method for unbalanced distribution systems,” *IET Gener. Transm. Distrib.*, vol. 5, no. 3, p. 347, 2011.

[120] D. Thukaram, H. M. Wijekoon Banda, and J. Jerome, “A robust three phase power flow algorithm for radial distribution systems,” *Electr. Power Syst. Res.*, vol. 50, no. 3, pp. 227–236, Jun. 1999.

[121] M. Z. Kamh and R. Iravani, “Unbalanced Model and Power-Flow Analysis of Microgrids and Active Distribution Systems,” *IEEE Trans. Power Deliv.*, vol. 25, no. 4, pp. 2851–2858, 2010.

[122] T.-H. Chen, M.-S. Chen, K.-J. Hwang, P. Kotas, and E. A. Chebli, “Distribution system power flow analysis-a rigid approach,” *IEEE Trans. Power Deliv.*, vol. 6, no. 3, pp. 1146–1152, Jul. 1991.

[123] G. R. Cespedes, “New Method for the Analysis of Distribribution Networks,” *IEEE Trans. Power Deliv.*, vol. 5, no. 1, pp. 391–396, 1990.

[124] Y. Zhu and K. Tomsovic, “Adaptive power flow method for distribution systems with dispersed generation,” *IEEE Trans. Power Deliv.*, vol. 17, no. 3, pp. 822–827, Jul. 2002.

[125] M. H. Haque, “Load flow solution of distribution systems with voltage dependent load models,” *Electr. Power Syst. Res.*, vol. 36, no. 3, pp. 151–156, Mar. 1996.

[126] M. Haque, “Efficient load flow method for distribution systems with radial or mesh configuration,” *IEE Gener. Transm. Distrib.*, vol. 143, no. 1, pp. 33–38, 1996.

[127] D. Das, H. Nagi, and D. Kothari, “Novel method for solving radial distribution networks,” *IEE Gener. Transm. Distrib.*, vol. 141, no. 4, pp. 291–298, 1994.

[128] U. Eminoglu and M. H. Hocaoglu, “A new power flow method for radial distribution systems including voltage dependent load models,” *Electr. Power Syst. Res.*, vol. 76, no. 1–3, pp. 106–114, Sep. 2005.

[129] G. Chang, S.-Y. Chu, M.-F. Hsu, C.-S. Chuang, and H.-L. Wang, “An efficient power flow algorithm for weakly meshed distribution systems,” *Electr. Power Syst. Res.*, vol. 84, no. 1, pp. 90–99, Mar. 2012.

[130] T.-H. Chen and N.-C. Yang, “Loop frame of reference based three-phase power flow for unbalanced radial distribution systems,” *Electr. Power Syst. Res.*, vol. 80, no. 7, pp. 799–806, Jul. 2010.

[131] M. AlHajri and M. El-Hawary, “Exploiting the radial distribution structure in developing a fast and flexible radial power flow for unbalanced three-phase networks,” *IEEE Trans. Power Deliv.*, vol. 25, no. 1, pp. 378–389, 2010.

[132] W. C. Wu and B. M. Zhang, “A three-phase power flow algorithm for distribution system power flow based on loop-analysis method,” *Int. J. Electr. Power Energy Syst.*, vol. 30, no. 1, pp. 8–15, Jan. 2008.

[133] A. Augugliaro, L. Dusonchet, S. Favuzza, M. G. Ippolito, and E. R. Sanseverino, “A backward sweep method for power flow solution in distribution networks,” *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 4, pp. 271–280, May 2010.

[134] W. Lin, Y. Su, H. Chin, and J. Teng, “Three-phase unbalanced distribution power flow solutions with minimum data preparation,” *IEEE Trans. Power Syst.*, vol. 14, no. August, pp. 1178–1183, 1999.

[135] J. Vlachogiannis, “Fuzzy logic application in load flow studies,” *IET Gener. Transm. Distrib.*, vol. 148, no. 1, pp. 34–40, 2001.

[136] Z. Wang and F. Alvarado, “Interval arithmetic in power flow analysis,” *IEEE Trans. Power Syst.*, vol. 7, no. 3, pp. 1341–1349, 1992.

[137] B. M. Kalesar and A. R. Seifi, “Fuzzy load flow in balanced and unbalanced radial distribution systems incorporating composite load model,” *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 1, pp. 17–23, Jan. 2010.

[138] B. Das, “Consideration of Input Parameter Uncertainties in Load Flow Solution of Three-Phase Unbalanced,” *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1088–1095, 2006.

[139] J. Allemong, R. Bennon, and P. Selent, “Multiphase power flow solutions using EMTP and Newtons method,” *IEEE Trans. Power Syst.*, vol. 8, no. 4, pp. 1455–1462, 1993.

[140] J. Vieira Jr, W. Freitas, and A. Morelato, “Phase-decoupled method for three-phase power-flow analysis of unbalanced distribution systems,” *IEE Proc. - Gener. Transm. Distrib.*, vol. 151, no. 5, 2004.

[141] J. Teng, “A modified Gauss-Siedel algorithm of three phase power flow analysis in distribution networks,” *Electr. Power Energy Syst.*, vol. 24, pp. 2–7, 2002.

[142] P. Bijwe and G. Raju, “Fuzzy distribution power flow for weakly meshed systems,” *Power Syst. IEEE Trans.*, vol. 21, no. 4, pp. 1645–1652, 2006.

[143] E. Bompard, E. Carpaneto, G. Chicco, and R. Napoli, “Convergence of the backward/forward sweep method for the load-flow analysis of radial distribution systems,” *Int. J. Electr. Power Energy Syst.*, vol. 22, no. 7, pp. 521–530, Oct. 2000.

[144] F. Zhang and C. S. Cheng, “A modified Newton method for radial distribution system power flow analysis,” *IEEE Trans. Power Syst.*, vol. 12, no. 1, pp. 389–397, 1997.

[145] R. M. Ciric, A. P. Feltrin, and L. F. Ochoa, “Power Flow in Four-Wire Distribution Networks — General Approach,” *IEEE Trans. Power Syst.*, vol. 18, no. 4, pp. 1283–1290, 2003.

[146] J. Peralta, F. de Leon, and J. Mahseredjian, “Unbalanced multiphase load-flow using a positive-sequence load-flow program,” *Power Syst. IEEE …*, vol. 23, no. 2, pp. 469–476, 2008.

[147] J. Martinez and J. Mahseredjian, “Load flow calculations in distribution systems with distributed resources. A review,” *Power Energy Soc. …*, pp. 1–8, 2011.

[148] J. J. Grainger and W. D. Stevenson, *Power System Analysis*. McGraw-Hill, 1994.

[149] M. Thomson and D. G. Infield, “Network power-flow analysis for high-penetration distributed micro-generation,” *18th Int. Conf. Exhib. Electr. Distrib. (CIRED 2005)*, vol. 2005, no. 4, pp. v4–22–v4–22, 2005.

[150] M. Thomson, D. Infield, M. Stokes, M. Rylatt, J. Mardaljevic, and K. J. Lomas, “Secondary distribution network power-flow analysis,” in *Proceedings of the IASTED International Conference*, 2003.

[151] M. A. Laughton, “Analysis of unbalanced polyphase networks by the method of phase co-ordinates Part 1 . System representation in phase frame of reference,” vol. 115, no. 8, 1968.

[152] M. A. Laughton, “Analysis of unbalanced polyphase networks by the method of phase co-ordinates,” *Proc. IEEE*, vol. 116, no. 5, 1969.

[153] R. C. Dugan, “A perspective on transformer modeling for distribution system analysis,” in *2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No.03CH37491)*, 2003, vol. 1, pp. 114–119.

[154] B. Borkowska, “Probabalistic Load Flow,” *IEEE Trans. Power Appar. Syst.*, vol. 93, no. 2, pp. 752–759, 1974.

[155] R. Allan and M. Al-Shakarchi, “Probabilistic techniques in ac load-flow analysis,” *Proc. Inst. Electr. Eng.*, vol. 124, no. 2, pp. 154–160, 1977.

[156] R. Allan, L. Da Silva, and R. Burchett, “Evaluation methods and accuracy in probabilistic load flow solutions,” *IEEE Trans. Power Appar. Syst.*, vol. PAS-100, no. 5, pp. 2539–2546, 1981.

[157] R. Allan and A. Da Silva, “Probabilistic load flow using multilinearisations,” *IEE Proc. C (Generation, Transm. …*, vol. 128, no. 5, 1981.

[158] A. M. Leite da Silva and V. L. Arienti, “Probabilistic load flow by a multilinear simulation algorithm,” *IEE Proc. C (Generation, Transm. Distrib.*, vol. 137, no. 4, pp. 276–282, 1990.

[159] P. Zhang and S. Lee, “Probabilistic load flow computation using the method of combined cumulants and Gram-Charlier expansion,” *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 676–682, 2004.

[160] C. Su, “Probabilistic load-flow computation using point estimate method,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1843–1851, 2005.

[161] W. El-Khattam, Y. Hegazy, and M. Salama, “Investigating distributed generation systems performance using Monte Carlo simulation,” *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 524–532, 2006.

[162] J. Morales and J. Perez-Ruiz, “Point estimate schemes to solve the probabilistic power flow,” *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1594–1601, 2007.

[163] H. Yu, C. Chung, K. Wong, L. HW, and H. Zhang, “Probabilistic load flow evaluation with hybrid latin hypercube sampling and cholesky decomposition,” *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 661–667, 2009.

[164] J. Usaola, “Probabilistic load flow in systems with wind generation,” *IET Gener. Transm. Distrib.*, vol. 3, no. 12, p. 1031, 2009.

[165] J. M. Morales, L. Baringo, a. J. Conejo, and R. Mínguez, “Probabilistic power flow with correlated wind sources,” *IET Gener. Transm. Distrib.*, vol. 4, no. 5, p. 641, 2010.

[166] N. D. Hatziargyriou and T. S. Karakatsanis, “Distribution system voltage and reactive power control based on probabilistic load flow analysis,” *IEE Proc. - Gener. Transm. Distrib.*, vol. 144, no. 4, p. 363, 1997.

[167] P. Caramia and G. Carpinelli, “Probabilistic three-phase load flow for unbalanced electrical distribution systems with wind farms,” *Renew. Power Gener.*, vol. 1, no. 2, pp. 115–122, 2007.

[168] A. Bracale, G. Carpinelli, P. Caramia, A. Russo, and P. Varilone, “Point estimate schemes for probabilistic load flow analysis of unbalanced electrical distribution systems with wind farms,” *Proc. 14th Int. Conf. Harmon. Qual. Power - ICHQP 2010*, Sep. 2010.

[169] Y. Wang and M. Yang, “Probabilistic modeling of three-phase voltage unbalance caused by load fluctuations,” *Power Eng. Soc. Winter Meet.*, pp. 2588–2593, 2000.

[170] K. Schneider, D. Chassin, Y. Chen, and J. Fuller, “Distribution power flow for smart grid technologies,” *Power Syst. Conf. Expo. 2009. PSCE ’09. IEEE/PES*, pp. 1–7, 2009.

[171] D. P. Chassin, J. C. Fuller, and N. Djilali, “GridLAB - D: An agent - based simulation framework for smart grids - Article ID 492320,” *J. Appl. Math.*, 2014.

[172] F. Shahnia, R. Majumder, A. Ghosh, G. Ledwich, and F. Zare, “Voltage imbalance analysis in residential low voltage distribution networks with rooftop PVs,” *Electr. Power Syst. Res.*, vol. 81, no. 9, pp. 1805–1814, 2011.

[173] R. C. Dugan, “Reference Guide The Open Distribution System Simulator (OpenDSS) - Program Revision 7.6,” 2013.

[174] “OpenDSS Discussion Webpage - ‘Linux?’” [Online]. Available: http://sourceforge.net/p/electricdss/discussion/861976/thread/3500889e/. [Accessed: 21-Jul-2014].

[175] H. Hahn, S. Meyer-Nieberg, and S. Pickl, “Electric load forecasting methods: Tools for decision making,” *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 902–907, Dec. 2009.

[176] H. K. Alfares and M. Nazeeruddin, “Electric load forecasting: Literature survey and classification of methods,” *Int. J. Syst. Sci.*, vol. 33, no. 1, pp. 23–34, 2002.

[177] L. Suganthi and A. a. Samuel, “Energy models for demand forecasting—A review,” *Renew. Sustain. Energy Rev.*, vol. 16, no. 2, pp. 1223–1240, Feb. 2012.

[178] A. Capasso, W. Grattieri, R. Lamedica, and A. Prudenzi, “A bottom-up approach to residential load modeling,” *IEEE Trans. Power Syst.*, vol. 9, no. 2, pp. 957–964, May 1994.

[179] J. V. Paatero and P. D. Lund, “A model for generating household electricity load profiles,” *Int. J. Energy Res.*, vol. 30, no. 5, pp. 273–290, Apr. 2006.

[180] I. Richardson, M. Thomson, D. Infield, and C. Clifford, “Domestic electricity use: A high-resolution energy demand model,” *Energy Build.*, vol. 42, no. 10, pp. 1878–1887, Oct. 2010.

[181] M. Hand, E. Shove, and D. Southerton, “Explaining Showering: a Discussion of the Material, Conventional, and Temporal Dimensions of Practice,” *Sociol. Res. Online*, vol. 10, no. 2, p. hand, 2005.

[182] B. Drysdale, “Demand Side Management: Flexible demand in the GB domestic electricity sector,” Cardiff University, 2014.

[183] M. Cheng, J. Wu, J. Ekanayake, and N. Jenkins, “Primary frequency response in the Great Britain power system from dynamically controlled refrigerators,” in *Presented at: CIRED 2013, Stockholm, Sweden, 10-13 June 2013.*

[184] A. Bagdanavicius and N. Jenkins, “Power requirements of ground source heat pumps in a residential area,” *Appl. Energy*, vol. 102, pp. 591–600, Feb. 2013.

[185] S. Ingram, S. Probert, and K. Jackson, “DTI - The Impact of Small Scale Embedded Generation on the Operating Parameters of Distribution Networks,” 2003. [Online]. Available: http://bit.ly/A7qHGG. [Accessed: 14-Nov-2015].

[186] P. Trichakis, P. Taylor, L. Cipcigan, P. Lyons, R. Hair, and T. Ma, “An investigation of voltage unbalance in low voltage distribution networks with high levels of SSEG,” *Univ. Power Eng. Conf. (UPEC), 2006*, pp. 182–186, 2006.

[187] M. Thomson and D. G. Infield, “Impact of widespread photovoltaics generation on distribution systems,” *IET Renew. Power Gener.*, vol. 1, no. 1, p. 33, 2007.

[188] A. Navarro, L. F. Ochoa, and D. Randles, “Monte Carlo-based assessment of PV impacts on real UK low voltage networks,” in *IEEE Power and Energy Society General Meeting - Vancouver*, 2013, pp. 1–5.

[189] M. H. F. Wen, R. Arghandeh, A. Von Meier, K. Poolla, and V. O. K. Li, “Phase Identification in Distribution Networks with Micro-Synchrophasors - Pre-print version arXiv:1501.04044,” *arXiv*, 2015. [Online]. Available:

http://arxiv.org/pdf/1501.04044v1. [Accessed: 14-May-2015].

[190] H. Pezeshki and P. Wolfs, “Correlation Based Method for Phase Identification in a Three Phase LV Distribution Network,” in *Universities Power Engineering Conference (AUPEC), 22nd Australasian*, 2012.

[191] B. K. Seal and M. F. McGranaghan, “Automatic identification of service phase for electric utility customers,” *IEEE Power Energy Soc. Gen. Meet.*, pp. 1–3, 2011.

[192] T. A. Short, “Advanced metering for phase identification, transformer identification, and secondary modeling,” *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 651–658, 2013.

[193] C. S. Chen, T. T. Ku, and C. H. Lin, “Design of phase identification system to support three-phase loading balance of distribution feeders,” *IEEE Trans. Ind. Appl.*, vol. 48, no. 1, pp. 191–198, 2012.

[194] V. Arya, V. T. Chakaravarthy, K. J. Dontas, S. T. Hoy, J. R. Kalagnanam, S. Kalagnanam, C. J. Pavlovski, P. Devasenapathi, and D. P. Seetharamakrishnan, “Systems and Methods for Phase Identifcation - United States Patent,” US 8825416B22014.

[195] V. Arya, D. Seetharam, S. Kalyanaraman, K. Dontas, C. Pavlovski, S. Hoy, and J. R. Kalagnanam, “Phase identification in smart grids,” *2011 IEEE Int. Conf. Smart Grid Commun.*, pp. 25–30, 2011.

[196] Z. Fan, Q. Chen, G. Kalogridis, S. Tan, and D. Kaleshi, “The power of data: Data analytics for M2M and smart grid,” *2012 3rd IEEE PES Innov. Smart Grid Technol. Eur. (ISGT Eur.*, pp. 1–8, 2012.

[197] M. Dilek, “Integrated Design of Electrical Distribution Systems : Phase Balancing and Phase Prediction Case Studies Integrated Design of Electrical Distribution Systems : Phase Balancing and Phase Prediction Case Studies,” Virginia Polytechnic Institute and State University, 2001.

[198] K. J. Caird, “Meter Phase Identification - United States Patent,” US008143879B22012.

[199] V. Arya, R. Mitra, R. Mueller, H. Storey, G. Labut, J. Esser, and B. Sullivan, “Voltage Analytics to Infer Customer Phase,” 2014, pp. 1–6.

[200] V. Arya and R. Mitra, “Voltage-based clustering to identify connectivity relationships in distribution networks,” *2013 IEEE Int. Conf. Smart Grid Commun. SmartGridComm 2013*, pp. 7–12, 2013.

[201] T. H. Cormen, E. C. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edi. McGraw-Hill, 2001.

[202] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic, 1997.

[203] P. Judea, *Heuristics : intelligent search strategies for computer problem solving*. Addison-Wesley, 1984.

[204] D. MacKay, *Information Theory, Inference, and Learning Algorithms - Chapter 20*. Cambridge University Press, 2003.

[205] C. Aggarwal and K. Reddy, *Data Clustering - Algorithms and Applications*. Taylor & Francis, 2014.

[206] D. A. Clark, “Electromagnetic Fast-transients in LV Networks with Ubiquitous Small-scale Embedded Generation - Phd Thesis,” Cardiff University, 2012.

[207] P. Papadopoulos, S. Skarvelis-Kazakos, I. Grau, L. M. Cipcigan, and N. Jenkins, “Electric vehicles’ impact on British distribution networks,” *IET Electr. Syst. Transp.*, vol. 2, no. 3, p. 91, 2012.

[208] P. Mitra, G. T. Heydt, and V. Vittal, “The impact of distributed photovoltaic generation on residential distribution systems,” *2012 North Am. Power Symp. NAPS 2012*, 2012.

[209] M. Thomson and D. G. Infield, “Network Power-Flow Analysis for a High Penetration of Distributed Generation,” *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1157–1162, Aug. 2007.

[210] F. Shahnia, R. Majumder, A. Ghosh, G. Ledwich, and F. Zare, “Sensitivity analysis of voltage imbalance in distribution networks with rooftop PVs,” *IEEE PES Gen. Meet.*, pp. 1–8, 2010.

[211] Y. Liu, J. Bebic, B. Kroposki, J. de Bedout, and W. Ren, “Distribution system voltage performance analysis for high-penetration PV,” in *IEEE Energy 2030 Conference*, 2008, no. November, pp. 1–8.

[212] T. E. Oliphant, “Python for Scientific Computing,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 10–20, 2007.

[213] D. Goldberg, “What Every Computer Scientist Should Know About Floating-Point Arithmetic,” *ACM Comput. Surv.*, vol. 23, no. 1, pp. 5–48, 1991.

[214] H. Dommel, “Digital Computer Solution of Electromagnetic Transients in Single- and Multiphase Networks,” *IEEE Trans. Power Appar. Syst.*, vol. PAS-88, no. 4, pp. 388–399, Apr. 1969.

[215] H. W. Dommel and W. S. Meyer, “Computation of electromagnetic transients,” *Proc. IEEE*, vol. 62, no. 7, pp. 983–993, 1974.

[216] L. Grcev and M. Popov, “On high-frequency circuit equivalents of a vertical ground rod,” *Power Deliv. IEEE Trans.*, vol. 20, no. 2, pp. 1598–1603, 2005.

[217] I. Richardson and M. Thomson, “Integrated simulation of photovoltaic micro-generation and domestic electricity demand: a one-minute resolution open-source model,” *Proc. Inst. Mech. Eng. Part A J. Power Energy*, Aug. 2012.

[218] I. Richardson, M. Thomson, and D. Infield, “A high-resolution domestic building occupancy model for energy demand simulations,” *Energy Build.*, vol. 40, no. 8, pp. 1560–1566, Jan. 2008.

[219] D. MacKay, “Sustainable Energy: Without the Hot Air - Chapter 6, Page 39,” 2009. [Online]. Available: http://www.withouthotair.com/c6/page_39.shtml. [Accessed: 11-Aug-2014].

[220] Prysmian Cables and Systems, “Standard Power Cables Low Voltage (600/1000V) BS5467 Data Sheet, REF: BS5467/07/08.” .

[221] G. Tsagarakis, A. J. Collin, and A. E. Kiprakis, “Modelling the electrical loads of UK residential energy users,” *2012 47th Int. Univ. Power Eng. Conf.*, pp. 1–6, Sep. 2012.

[222] “StackOverflow.com question - What similarity metric should be used to classify these sequences?” [Online]. Available: http://stackoverflow.com/q/28411296/1461850. [Accessed: 01-Dec-2015].

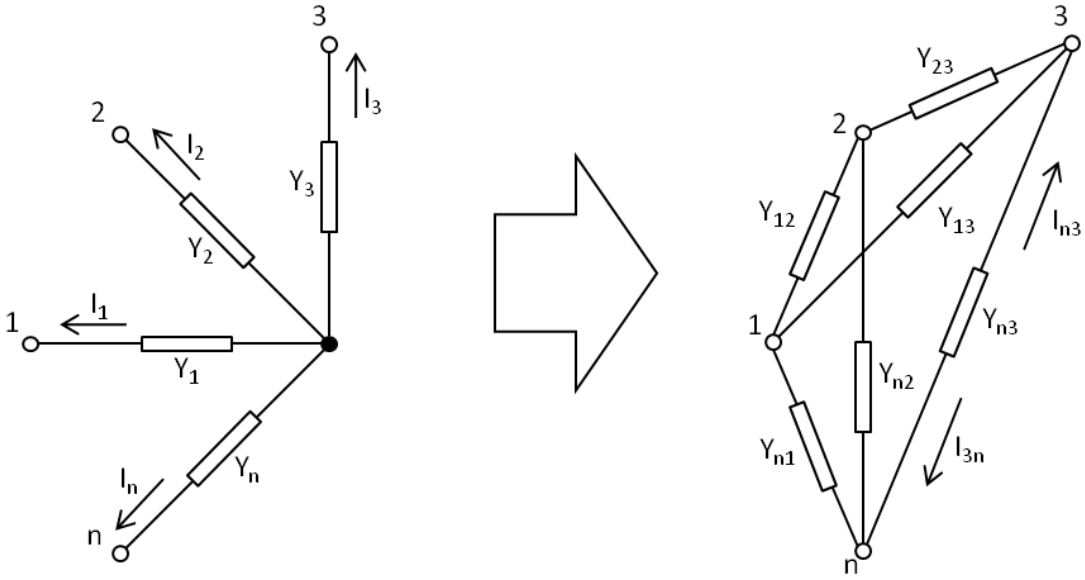
[223] E. Harder, *Westinghouse Electrical Transmission and Distribution Reference Book - Chapter 5*, Third Edit. 1944.

[224] R. Kerchner and G. Corcoran, *Alternating Current Circuits*, 4th ed. Wiley International, 1960.

A1 Appendix 1 – Star to Mesh Transformation and Nodal Analysis

Star to Mesh Transformation

After reference [223]. This is the “Star to Mesh Conversion” indicated in Figure 3.4.



“A network can be solved by eliminating one junction point after another until a single branch mesh remains.”

“The mesh contains n(n-1) branches, where n is the number of star branches.”

“A mesh branch is the product of adjacent star branches divided by the sum of all star branches.”

$D = Y_1 + Y_2 + Y_3 + ... + Y_n$

$Y_{12} = Y_1Y_2/D$ $\qquad$ $I_1 = I_{21} + I_{31} + ... + I_{n1}$

$Y_{13} = Y_1Y_3/D$ $\qquad$ $I_2 = I_{12} + I_{32} + ... + I_{n2}$

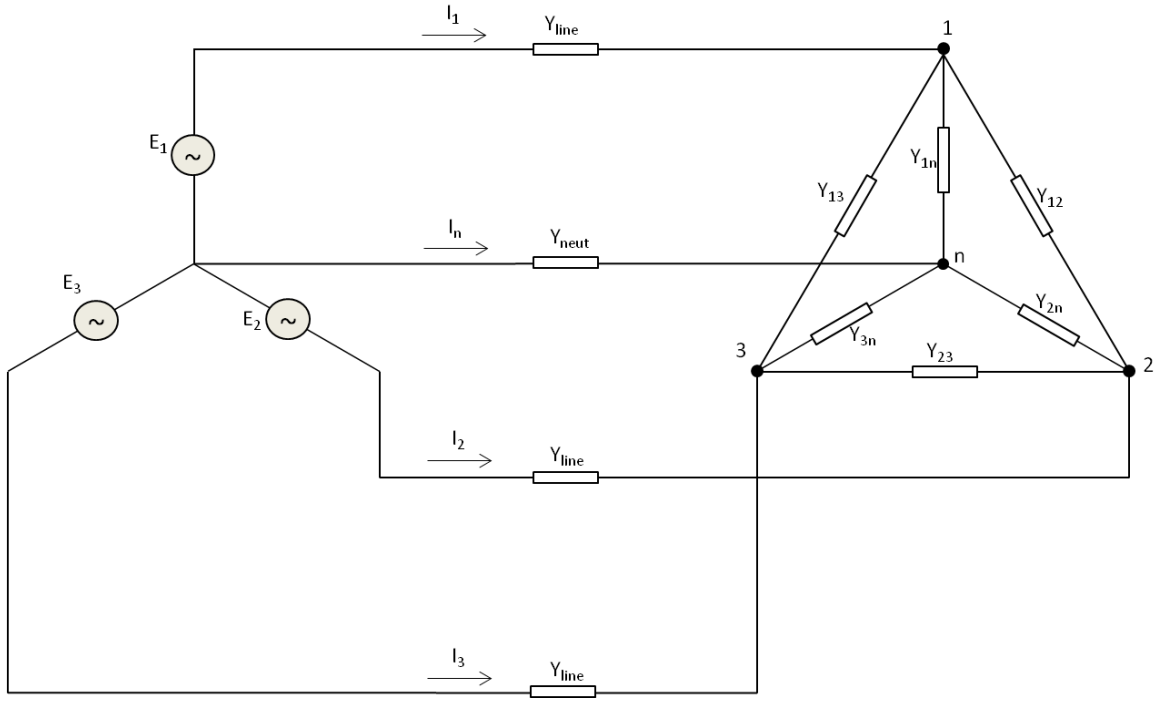
... $\qquad$ ...

$Y_{pq} = Y_pY_q/D$ $\qquad$ $I_p = I_{1p} + I_{2p} + ... + I_{np}$

Figure A1.1 - General star to mesh transformation, or elimination of a junction - admittance form.

Nodal Analysis

Based on [224]. This is the “Nodal Analysis” indicated in Figure 3.4.



$Y_{11}V_1 - Y_{12}V_2 - Y_{13}V_3 - Y_{1n}V_n = I_1 = Y_{line}E_1$

$-Y_{12}V_1 + Y_{22}V_2 - Y_{23}V_3 - Y_{2n}V_n = I_2 = Y_{line}E_2$

$-Y_{13}V_1 - Y_{23}V_2 + Y_{33}V_3 - Y_{1n}V_n = I_3 = Y_{line}E_3$

$-Y_{1n}V_1 - Y_{2n}V_2 - Y_{3n}V_3 + Y_{nn}V_n = I_n = Y_{neut}E_n$

In matrix form:

$\mathbf{YV} = \mathbf{I}$

$\mathbf{V} = \mathbf{Y}^{-1}\mathbf{I}$

Where

$$\boldsymbol{Y} = \begin{bmatrix} Y_{11} & -Y_{12} & -Y_{13} & -Y_{1n} \\ -Y_{12} & Y_{22} & -Y_{23} & -Y_{2n} \\ -Y_{13} & -Y_{23} & Y_{33} & -Y_{3n} \\ -Y_{1n} & -Y_{2n} & -Y_{3n} & Y_{nn} \end{bmatrix}, \boldsymbol{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_n \end{bmatrix} \text{ and } \boldsymbol{I} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{line}E_1 \\ Y_{line}E_2 \\ Y_{line}E_3 \\ Y_{neut}E_n \end{bmatrix}$$

$Y_{11} = Y_{line} + Y_{12} + Y_{13} + Y_{1n}$

$Y_{22} = Y_{line} + Y_{12} + Y_{23} + Y_{2n}$

$Y_{33} = Y_{line} + Y_{13} + Y_{23} + Y_{3n}$

$Y_{nn} = Y_{neut} + Y_{1n} + Y_{2n} + Y_{3n}$

Figure A1.2 - Nodal analysis to get the voltages given known source voltages and admittances

A2 Appendix 2 – Chapter 3 Code – Unbalanced load flow with variable input connection imbalance

```
def lvsolve(slackV,L1P,L1Q,L2P,L2Q,L3P,L3Q,phaseconfig):
        import numpy as np
        import VUFer
        reload(VUFer)
        # this is a solution based on Berg et al and the nodal method
        # it accepts a phaseconfig array describing the phase configuration at each branch point along the main feeder.
        # e.g. [3,0,0] means that all three loads are connected to
        # It has been validated as having close agreement with PSCAD within 0.0181% of load voltage magnitudes (e.g. with Fixed apparent power loads, real power values sampled from a gamma distribution* at 0.9 power factor. Unbalanced: Selected randomly with a bias towards phase 1 (40%,30%,30%). Totals on each phase were 25, 16 and 19. ).
        nonodes=len(L1P)
        vsource= np.array([slackV,slackV *(-0.5-0.8660254037844387j),slackV *(-0.5+0.8660254037844387j)]).reshape(3,)
        Iload=np.zeros((1,3),dtype=complex)
        Vload=np.zeros((1,3),dtype=complex)

        # Initialise working array
        nodedata = np.zeros((3,13,nonodes),dtype=complex)
        S= np.ones((3,nonodes),dtype=complex)
        S[0,:] = L1P + L1Q*1j
        S[1,:] = L2P + L2Q*1j
        S[2,:] = L3P + L3Q*1j
        S[S==0]=0.0000001# set any zeros to very small number to avoid divide by 0 warning and subsequent errors

        for node in range(0,nonodes):
                nodedata[:,11,node] = S[:,node] #
                if np.any(phaseconfig[node,:]==2):
                        z=np.asarray(np.where(phaseconfig[node,:]==0))
                        y=np.asarray(np.where(phaseconfig[node,:]==2))
```

```
                S[y,node]=S[y,node]+S[z,node]
                S[z,node] =0.0000001
        if np.any(phaseconfig[node,:]==3):
                y=np.asarray(np.where(phaseconfig[node,:]==3))
                S[y,node]=np.sum(S[:,node])
                for r in [0,1,2]:
                        if r == y:
                                pass
                        else:
                                S[r,node] =0.0000001

# # impedances: for 32 nodes
# zlinea=0.00154+0.000694j # line impedance of each phase
# zlineb=0.003+0.000703j # line impedance of each phase
# zservice = 0.0255+0.00123j
# zna=0.00154+0.000694j# neutral impedance
# znb=0.003+0.00015j# neutral impedance

#impedances for 20 nodes
zlinea=0.00246+0.00111j# line impedance of each phase
zlineb=0.0048+0.001125j# line impedance of each phase
zservice = 0.0255+0.00123j#

zna=0.00246+0.00111j# neutral impedance
znb=0.0048+0.00024j# neutral impedance

# admittances:
ylinea = 1/zlinea
ylineb = 1/zlineb
yservice = 1/zservice
yna = 1/zna
ynb = 1/znb

# put initial values working array
```

```
for node in range(0,nonodes):
        nodedata[:,7,node]=vsource
        if node >(nonodes/2)-1:
                nodedata[0,1,node] = ylineb
                nodedata[0,2,node] = ynb
        else:
                nodedata[0,1,node] = ylinea
                nodedata[0,2,node] = yna


        nodedata[1,1,node] = yservice
        nodedata[1,2,node] = yservice


        if np.any(phaseconfig[node,:]==2):
                nodedata[:,3,node] =  1/(nodedata[:,7,node]*np.conj(nodedata[:,7,node]/nodedata[:,11,node])+[2*zservice])
                nodedata[:,0,node] = 1/(nodedata[:,7,node]*np.conj(nodedata[:,7,node]/S[:,node])+[2*zservice])

                z=np.asarray(np.where(phaseconfig[node,:]==0))
                w=np.asarray(np.where(phaseconfig[node,:]==2))
                x=np.asarray(np.where(phaseconfig[node,:]==1))
                Y1 = 1/((nodedata[w,7,node]*np.conj(nodedata[w,7,node]/nodedata[w,11,node])) + [2*zservice])
                Y2 = 1/((nodedata[z,7,node]*np.conj(nodedata[z,7,node]/nodedata[z,11,node])) + [2*zservice])
                nodedata[w,0,node] = Y1 + Y2
                nodedata[z,0,node] = 0
                nodedata[w,12,node] = 1/(nodedata[w,7,node]*np.conj(nodedata[w,7,node]/nodedata[w,11,node]))
                nodedata[z,12,node] = 1/(nodedata[w,7,node]*np.conj(nodedata[w,7,node]/nodedata[z,11,node]))
                nodedata[x,12,node] = 1/(nodedata[x,7,node]*np.conj(nodedata[x,7,node]/nodedata[x,11,node]))

        elif np.any(phaseconfig[node,:]==3):
                w=np.asarray(np.where(phaseconfig[node,:]==3))
                z=np.asarray(np.where(phaseconfig[node,:]!=3))
                nodedata[:,3,node] =  1/(nodedata[w,7,node]*np.conj(nodedata[w,7,node]/nodedata[:,11,node])+[2*zservice])
                nodedata[w,0,node] = 1/(nodedata[w,7,node]*np.conj(nodedata[w,7,node]/S[w,node])+[2*zservice])
                nodedata[z,0,node] = 0
```

```
                nodedata[:,12,node] = 1/(nodedata[w,7,node]*np.conj(nodedata[w,7,node]/nodedata[:,11,node]))
        else:
                nodedata[:,3,node] =  1/(nodedata[:,7,node]*np.conj(nodedata[:,7,node]/nodedata[:,11,node])+[2*zservice])
                nodedata[:,0,node] = 1/(nodedata[:,7,node]*np.conj(nodedata[:,7,node]/nodedata[:,11,node])+[2*zservice])
                nodedata[:,12,node] = 1/(nodedata[:,7,node]*np.conj(nodedata[:,7,node]/nodedata[:,11,node]))
        nodedata[:,4,node] = S[:,node]

#       Yloadcalc = The Ys used in the calculations, includes 'dummy' very high values to represent open circuits for certain phase configurations
#   Yloadserv the 3 Ys associated with the input S values combined with the service cable impedance
#   Yload  - the 3 Ys associated with the input S values alone

#Program structure
# 1. set up dpns ('driving point network' - see Berg-Hawkins-Pleines Paper
# 2. calc vs dpns
# 3. calc load vs
# 4 recalc yload using calculated V and given S
# repeat

Vabsprev = np.ones((3,1,nonodes),dtype=complex)
n=0

for r in range(1,150):#150
        for k in range(0,nonodes):                          # 1. set up dpns
                node = nonodes-k-1 # from nonodes to 1
                if node != nonodes-1:
                        # combine line Ys to to-node dpn - star to mesh transformations
                        # remove node1
                        A= nodedata[0,1,node+1] # Line Admittance
                        B= nodedata[0,5,node+1] # Ydpn1
                        C= nodedata[0,6,node+1] # YDdpn12
                        D = nodedata[2,6,node+1] # YDdpn31
                        E = nodedata[1,6,node+1] # YDdpn23
                        F = nodedata[1,5,node+1] # Ydpn2
                        G = nodedata[2,5,node+1] # Ydpn3
```

```
D1 = A+B+C+D
nodedata[2,6,node] = A*D/D1 # =Yli*YDdpn31/sum
nodedata[0,6,node] = A*C/D1 # =Yli*YDdpn12/sum
nodedata[0,5,node] = A*B/D1 # =Yli*Ydpn1/sum
nodedata[1,6,node] = C*D/D1 + E # =YDdpn12*YDdpn31/sum + YDdpn23
nodedata[1,5,node] = B*C/D1 + F # =Ydpn1*YDdpn12/sum + Ydpn2
nodedata[2,5,node] = B*D/D1 + G # =Ydpn1*YDdpn31/sum + Ydpn3
# remove node1
B= nodedata[0,5,node]
C= nodedata[0,6,node]
D = nodedata[2,6,node]
E = nodedata[1,6,node]
F = nodedata[1,5,node]
G = nodedata[2,5,node]

# remove node2
D2 = C+F+E+A
nodedata[1,6,node] = A*E/D2
nodedata[0,6,node] = C*A/D2
nodedata[1,5,node] = A*F/D2
nodedata[0,5,node] = F*C/D2 + B
nodedata[2,6,node] = C*E/D2 + D
nodedata[2,5,node] = F*E/D2 + G

B= nodedata[0,5,node]   #Ydpn1
C= nodedata[0,6,node]   #YDdpn12
D = nodedata[2,6,node]   #YDdpn31
E = nodedata[1,6,node]   #YDdpn23
F = nodedata[1,5,node]   #Ydpn2
G = nodedata[2,5,node]   #Ydpn3
# remove node3
D3 = D+E+G+A
nodedata[2,6,node] = A*D/D3
nodedata[1,6,node] = A*E/D3
nodedata[2,5,node] = A*G/D3
```

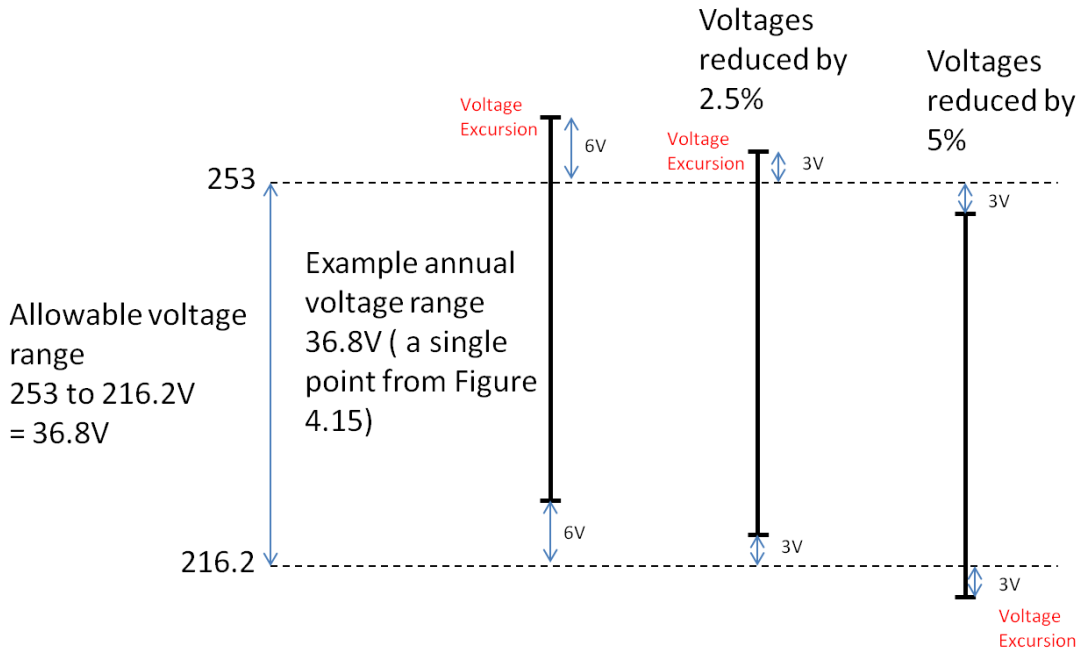
```
                nodedata[0,5,node] = G*D/D3 + B
                nodedata[1,5,node] = G*E/D3 + F
                nodedata[0,6,node] = D*E/D3 + C

                # remove nodeN
                B= nodedata[0,5,node]
                C= nodedata[0,6,node]
                D = nodedata[2,6,node]
                E = nodedata[1,6,node]
                F = nodedata[1,5,node]
                G = nodedata[2,5,node]
                H = nodedata[0,2,node+1]#H = nodedata[0,2,node+1]
                D4 = B+F+G+H
                nodedata[0,5,node] = H*B/D4
                nodedata[1,5,node] = H*F/D4
                nodedata[2,5,node] = H*G/D4
                nodedata[0,6,node] = B*F/D4 + C
                nodedata[1,6,node] = F*G/D4 + E
                nodedata[2,6,node] = B*G/D4 + D
                # add dpn addmittances to Y addmittances of current node (as in parallel)
                nodedata[:,5,node] = nodedata[:,0,node] + nodedata[:,5,node]
                #print 'node:'+repr(node)
        else:
                nodedata[:,6,node] = 0  # No delta admittances at end node
                nodedata[:,5,node] = nodedata[:,0,node] #Ydpn admittances = Y admittances of end node


for node in range(0,nonodes):                       # 2. calc v dpns - Nodal Analysis
        if node < nonodes-1:
                Y = np.array([[nodedata[0,5,node]+nodedata[0,1,node],-nodedata[0,6,node],-nodedata[2,6,node],-nodedata[0,5,node]],
                        [-nodedata[0,6,node],nodedata[1,5,node]+nodedata[0,1,node],-nodedata[1,6,node],-nodedata[1,5,node]],
                        [-nodedata[2,6,node],-nodedata[1,6,node],nodedata[2,5,node]+nodedata[0,1,node],-nodedata[2,5,node]],
                        [-nodedata[0,5,node],-nodedata[1,5,node],-nodedata[2,5,node],nodedata[0,2,node]+np.sum(nodedata[:,5,node])]])
```

```
                Iin = np.append(nodedata[:,7,node],nodedata[2,10,node])*np.append([nodedata[0,1,node]]*3,nodedata[0,2,node]) #Iin = Vin*Yin [Zline
Zneut]

                V = np.dot(np.linalg.inv(Y),Iin) ## reverted back (from above line) 28/3/2014
                nodedata[:,7,node+1]= V[0:3] ###
                nodedata[2,10,node+1]= V[3]

        else:
                Y = np.array([[nodedata[0,5,node]+nodedata[0,1,node],-nodedata[0,6,node],-nodedata[2,6,node],-nodedata[0,5,node]],
                        [-nodedata[0,6,node],nodedata[1,5,node]+nodedata[0,1,node],-nodedata[1,6,node],-nodedata[1,5,node]],
                        [-nodedata[2,6,node],-nodedata[1,6,node],nodedata[2,5,node]+nodedata[0,1,node],-nodedata[2,5,node]],
                        [-nodedata[0,5,node],-nodedata[1,5,node],-nodedata[2,5,node],nodedata[0,2,node]+np.sum(nodedata[:,5,node])]])
                Iin = np.append(nodedata[:,7,node],nodedata[2,10,node])*np.append([nodedata[0,1,node]]*3,nodedata[0,2,node])
                V = np.dot(np.linalg.inv(Y),Iin)

        # get phase currents

        Vtrunkn = V[0:3]-V[3]
        Iphase = Vtrunkn*nodedata[:,0,node]

        #get load currents (current dividers where unbalanced phase configuration)

        if np.any(phaseconfig[node,:]==2):
                z=np.asarray(np.where(phaseconfig[node,:]==0))
                w=np.asarray(np.where(phaseconfig[node,:]==2))
                x=np.asarray(np.where(phaseconfig[node,:]==1))
                Vload = Vload.reshape((1,3))
                Iload[:,x]  = Iphase[x]
                Iload[:,w]= nodedata[w,3,node]*(Iphase[w])/(nodedata[w,3,node]+nodedata[z,3,node])
                Iload[:,z]  = nodedata[z,3,node]*(Iphase[w])/(nodedata[w,3,node]+nodedata[z,3,node])
                Vload[:,x] = (1/nodedata[x,12,node])*Vtrunkn[x]/((1/nodedata[x,12,node])+2*zservice)
                Vload[:,w] = (1/nodedata[w,12,node])*Vtrunkn[w]/((1/nodedata[w,12,node])+2*zservice)
                Vload[:,z] = (1/nodedata[z,12,node])*Vtrunkn[w]/((1/nodedata[z,12,node])+2*zservice)
        elif np.any(phaseconfig[node,:]==3):
                w=np.asarray(np.where(phaseconfig[node,:]==3))
                Iload  = nodedata[:,3,node]*Iphase[w]/(np.sum(nodedata[:,3,node]))
```

```
                        Vload  = nodedata[:,11,node]/np.conj(Iload[:])
                        Vload  = Iload[:]/nodedata[:,12,node]
                        Vload = (1/nodedata[:,12,node])*Vtrunkn[w]/((1/nodedata[:,12,node])+2*zservice)
                else:
                        Iload  = Iphase.reshape((1,3))
                        Vload  = Iload[:]/nodedata[:,12,node]
                        Vload = (1/nodedata[:,12,node])*Vtrunkn/((1/nodedata[:,12,node])+2*zservice)
                nodedata[:,8,node] = Iload ###
                nodedata[:,9,node] = Vload ###

                nodedata[0,10,node]= V[3]
                if node < nonodes-1:
                        nodedata[1,10,node]=np.sum(Iphase) + nodedata[1,10,node+1]#np.sum(nodedata[:,8,node]) + nodedata[1,10,node+1]
                else:
                        nodedata[1,10,node]=np.sum(Iphase)#np.sum(nodedata[:,8,node])

        # Check for convergence of voltage values:
        if np.all(np.abs((np.absolute(nodedata[:,9,:])-np.absolute(Vabsprev[:,0,:]))) <0.0001):#0.0001
                for node in range(0,nonodes):
                        nodedata[2,1,node] = VUFer.VUFer(nodedata[0,7,node],nodedata[1,7,node],nodedata[2,7,node])
                print      'converged in ' + str(r+1) + ' iterations'
                #print nodedata.dtype
                return nodedata
        else:
                Vabsprev[:,0,:] = nodedata[:,9,:]
                n=n+1
                ###print 'iteration' + str(r)
                if r > 148:
                        ##print "did not converge"
                        return 'ERROR'

#         3 recalc yload (only gets here if no convergence)
        for node in range(0,nonodes):
                if np.any(phaseconfig[node,:]==2):
                        z=np.asarray(np.where(phaseconfig[node,:]==0))
```

```
                w=np.asarray(np.where(phaseconfig[node,:]==2))
                x=np.asarray(np.where(phaseconfig[node,:]==1))
                nodedata[:,12,node] = 1/(nodedata[:,9,node]*np.conj(nodedata[:,9,node]/nodedata[:,11,node]))
                Vtrunkn = nodedata[:,7,node]-nodedata[0,10,node]
                nodedata[:,3,node] = ((nodedata[:,12,node])/(1 + 2*nodedata[:,12,node]*zservice))
                Y1 = 1/((nodedata[w,9,node]*np.conj(nodedata[w,9,node]/nodedata[w,11,node])) + [2*zservice])
                Y2 = 1/((nodedata[z,9,node]*np.conj(nodedata[z,9,node]/nodedata[z,11,node])) + [2*zservice])
                nodedata[w,0,node] = nodedata[w,3,node] + nodedata[z,3,node]
                nodedata[z,0,node] = 0
                nodedata[x,0,node] = nodedata[x,3,node]
        elif np.any(phaseconfig[node,:]==3):
                w=np.asarray(np.where(phaseconfig[node,:]==3))
                z=np.asarray(np.where(phaseconfig[node,:]!=3))
                Vtrunkn = nodedata[:,7,node]-nodedata[0,10,node]
                nodedata[:,12,node] = 1/(nodedata[:,9,node]*np.conj(nodedata[:,9,node]/nodedata[:,11,node]))
                nodedata[:,3,node] = ((nodedata[:,12,node])/(1 + 2*nodedata[:,12,node]*zservice))
                nodedata[z,0,node] = 0
                nodedata[w,0,node] = np.sum(nodedata[:,3,node])
        else:
                nodedata[:,12,node] = 1/(nodedata[:,9,node]*np.conj(nodedata[:,9,node]/nodedata[:,11,node]))
                nodedata[:,3,node] = ((nodedata[:,12,node])/(1 + 2*nodedata[:,12,node]*zservice))
                nodedata[:,0,node] = nodedata[:,3,node]

# WORKING ARRAY STRUCTURE (nodedata):
#         0       1           2     3         4        5         6      7          8      9        10         11             12
#0 [   Yloadcalc1 Yli      Yli-n   Yloadserv1 Sload1 Zdpn1 ZDdpn12  vsource1 iload1 Vload1 Vneut       Sloadact1        Yload1
#1        Yloadcalc2 Yserv   Yserv-n Yloadserv2 Sload2 Zdpn2 ZDdpn23  vsource2 iload2 Vload2 Ineut     Sloadact2        Yload2
#2        Yloadcalc3 0       0        Yloadserv3 Sload3 Zdpn3 ZDdpn31  vsource3 iload3 Vload3 VsourceN  Sloadact3        Yload3]
```

A3 Appendix 3 – Observed annual voltage range and tap changer step size



If the observed voltage range is 36.8V and lies in between tap settings, either a high or low excursion will occur. See above.
Therefore, the allowable voltage range is reduced by 253 x 2.5% x 0.5 ~= 3V
2.5% is the fixed tap change step on the distribution transformer.

A4 Appendix 4 - Chapter 4 Code

Chapter 4 Code – Demand profile generation

```
#    Copyright (C) 2008, 2011 Ian Richardson*, Murray Thomson*,
#    Rewritten in Numerical Python, in 2013, by Lee Thomas**

#    *CREST (Centre for Renewable Energy Systems Technology),
#    Department of Electronic and Electrical Engineering
#    Loughborough University, Leicestershire LE11 3TU, UK
#    Tel. +44 1509 635326. Email address: I.W.Richardson@lboro.ac.uk

#    ** CIREGS (Centre for Integration of Renewable Energy Generation and Supply),
#    Cardiff School of Engineering
#    Cardiff University, CF24 3AA
#    Tel. +442920870674. Email address: ThomasL62@cf.ac.uk

#    This program is free software: you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation, either version 3 of the License, or
#    (at your option) any later version.

#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.

#    You should have received a copy of the GNU General Public License
#    along with this program.  If not, see <http://www.gnu.org/licenses/>.
import numpy
import pprint

def mainfunc():
        print 'IN CREATE profiles2 month 1 PV = 0'
        month = $MONTH                              # $MONTH replaced with (1,3,5,7,9 and 11)
        daytype = 'weekday'
        no_its = 3840
        PV_proportion = $PV                    # $PV varied from 0 to 100 in 10% steps
        from random import random
        import math
        import numpy
        import csv
        appliances = numpy.genfromtxt('appliances.dat',skip_header=27,dtype=(None))
        sim_dataP_for_file = numpy.zeros([no_its,1440])
        sim_dataQ_for_file = numpy.zeros([no_its,1440])
        #appliances_in_dwelling_for_file = numpy.empty([no_its,33],dtpye='a18')
```

```
appliances_in_dwelling_for_file = [[] for i in xrange(no_its)]
occ_profile_for_file = numpy.zeros([no_its,144])
idstring = str(no_its) + 'x_' + 'month-' + str(month) + '_' + 'daytype-' + str(daytype)+ '_PVprop-' + repr(int(PV_proportion))
oMonthlyRelativeTemperatureModifier = gettemperaturedata()
for i in range (0,no_its):
        appliances_in_dwelling = ConfigureAppliancesInDwelling(appliances)
        randno = random()
        activity_stats = numpy.genfromtxt('activity_stats.dat',skip_header=11,dtype=(None))
        sim_dataP = numpy.zeros([len(appliances_in_dwelling[:]),1440])
        sim_dataQ = numpy.zeros([len(appliances_in_dwelling[:]),1440])
        occ_profile = get_occ_profile(daytype)
        occ_profile_for_file[i][:] = occ_profile
        lighting_demand_data = RunLightingSimulation(month,occ_profile)
        for appliance in range(0,len(appliances_in_dwelling[:])-1):
                sApplianceType = appliances_in_dwelling[appliance][15]
                iMeanCycleLength = appliances_in_dwelling[appliance][3]
                iCyclesPerYear = appliances_in_dwelling[appliance][2]
                iStandbyPower = appliances_in_dwelling[appliance][5]
                iRatedPower = appliances_in_dwelling[appliance][4]
                dCalibration = appliances_in_dwelling[appliance][18]
                dOwnership = appliances_in_dwelling[appliance][0]
                iTargetAveragekWhYear = appliances_in_dwelling[appliance][21]
                sUseProfile = appliances_in_dwelling[appliance][16]
                iRestartDelay = appliances_in_dwelling[appliance][6]

                # Initialisation
                iCycleTimeLeft = 0
                dActivityProbability = 0
                # Randomly delay the start of appliances that have a restart delay (e.g. cold appliances with more regular intervals)
                iRestartDelayTimeLeft = random() * iRestartDelay * 2 # Weighting is 2 just to provide some diversity

                # Make the rated power variable over a normal distribution to provide some variation
                iRatedPower = GetMonteCarloNormalDistGuess(iRatedPower,iRatedPower/10)

                for iMinute in range (-198,1441):
                        if iMinute <=0:
                                #print iMinute
                                iMinute = 1440+iMinute
                                #print iMinute
                                #Get the ten minute period count
                                iTenMinuteCount = ((iMinute - 1)/10)
                                # Get the number of current active occupants for this minute (convert from 10 minute to 1 minute resolution)
                                iActiveOccupants = occ_profile[iTenMinuteCount]

                                # If this appliance is off having completed a cycle (ie. a restart delay)
                                if (iCycleTimeLeft <= 0) and (iRestartDelayTimeLeft > 0):
                                        #Decrement the cycle time left
                                        iRestartDelayTimeLeft = iRestartDelayTimeLeft - 1

                                # Else if this appliance is off
                                elif iCycleTimeLeft <= 0:
                                        # There must be active occupants, or the profile must not depend on occupancy for a start event to occur
```

```
                                        if (iActiveOccupants > 0 and sUseProfile != "CUSTOM") or (sUseProfile == "LEVEL"):
                                                # Variable to store the event probability (default to 1)
                                                dActivityProbability = 1
                                                # For appliances that depend on activity profiles and is not a custom profile ...
                                                if (sUseProfile != "LEVEL") and (sUseProfile != "ACTIVE_OCC") and (sUseProfile != "CUSTOM") and
(sUseProfile != "ARRIVAL"):

                                                        if daytype == 'weekday':
                                                                dayflag = 0
                                                        else:
                                                                dayflag = 1

                                                        [activity_days] = [activity_stats[:][x] for x in numpy.where(activity_stats['f0']==dayflag)]
                                                        [activity_occs] = [activity_days[:][x] for x in
numpy.where(activity_days['f1']==iActiveOccupants)]
                                                        [activity_use_profile] = [activity_occs[:][x] for x in
numpy.where(activity_occs['f2']==sUseProfile)] # sUseProfile = appliances[16] = activity type (string)
                                                        dActivityProbability = activity_use_profile[0][iTenMinuteCount+3] # Get the probability for this
activity profile for this time step


                                                        # For electric space heaters ... (excluding night storage heaters)
                                                elif sApplianceType == "ELEC_SPACE_HEATING":
                                                        # If this appliance is an electric space heater, then activity probability is a function of the
month of the year
                                                        dActivityProbability = round(oMonthlyRelativeTemperatureModifier[month],4)

                                                # Check the probability of a start event
                                                if (random() < ((dCalibration) * dActivityProbability)):
                                                        # This is a start event
                                                        [iPower, iCycleTimeLeft,iRestartDelayTimeLeft] = StartAppliance(iRestartDelay, iMeanCycleLength,
iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType)
                                                # Custom appliance handler: storage heaters have a simple representation
                                        elif sUseProfile == "CUSTOM" and sApplianceType == "STORAGE_HEATER":

                                                # The number of cycles (one per day) set out in the calibration sheet
                                                # is used to determine whether the storage heater is used
                                                # This model does not account for the changes in the Economy 7 time
                                                # It assumes that the time starts at 00:30 each day
                                                if iTenMinuteCount == 4:  # ie. 00:30 - 00:40
                                                        #Assume January 14th is the coldest day of the year
                                                        #Dim oDate, oDateOn, oDateOff As Date
                                                        #Dim monthOn, monthOff As Integer
                                                        #oDate = #1/14/1997#

                                                        # Get the month and day when the storage heaters are turned on and off, using the number of
cycles per year

                                                        monthOff = (14+(iCyclesPerYear/2))/4.3
                                                        monthOn = (365+14+(0-iCyclesPerYear/2))/4.3

                                                        # If this is a month in which the appliance is turned on of off
```

```
                                                        if month == monthOff or month == monthOn:
                                                                # Pick a 50% chance since this month has only a month of year resolution
                                                                dProbability = 0.5 / 10  # (since there are 10 minutes in this period)
                                                        elif month > monthOff and month < monthOn:
                                                                # The appliance is not used in summer
                                                                dProbability = 0
                                                        else:
                                                                # The appliance is used in winter
                                                                dProbability = 1

                                                        # Determine if a start event occurs
                                                        if random() <= dProbability:
                                                                # This is a start event
                                                                [iPower, iCycleTimeLeft,iRestartDelayTimeLeft] = StartAppliance(iRestartDelay,
iMeanCycleLength, iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType)

                                        else:
                                                # The appliance is on - if the occupants become inactive, switch off the appliance
                                                if iActiveOccupants == 0 and sUseProfile != "LEVEL" and sUseProfile != "ACT_LAUNDRY" and sUseProfile != "CUSTOM"
and sUseProfile != "ARRIVAL":
                                                        pass
                                                        # Do nothing. The activity will be completed upon the return of the active occupancy.
                                                        # Note that LEVEL means that the appliance use is not related to active occupancy.
                                                        # Note also that laundry appliances do not switch off upon a transition to inactive occupancy.
                                                else:
                                                        # Set the power
                                                        #do nothing here as pre 0 mins (working out what is on from previous night) #iPower =
GetPowerUsage(iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType)
                                                        # Decrement the cycle time left
                                                        iCycleTimeLeft = iCycleTimeLeft - 1

                                        # get lighting data and add to iPower
                                        # Set the appliance power at this time step

                                else:
                                        sim_dataP[appliance,iMinute-1] = iStandbyPower

                                        #Get the ten minute period count
                                        iTenMinuteCount = ((iMinute - 1)/10)
                                        # Get the number of current active occupants for this minute (convert from 10 minute to 1 minute resolution)
                                        iActiveOccupants = occ_profile[iTenMinuteCount]

                                        # If this appliance is off having completed a cycle (ie. a restart delay)
                                        if (iCycleTimeLeft <= 0) and (iRestartDelayTimeLeft > 0):
                                                #Decrement the cycle time left
                                                iRestartDelayTimeLeft = iRestartDelayTimeLeft - 1

                                        # Else if this appliance is off
                                        elif iCycleTimeLeft <= 0:
                                                # There must be active occupants, or the profile must not depend on occupancy for a start event to occur
                                                if (iActiveOccupants > 0 and sUseProfile != "CUSTOM") or (sUseProfile == "LEVEL"):
                                                        # Variable to store the event probability (default to 1)
```

```
                                                dActivityProbability = 1
                                                # For appliances that depend on activity profiles and is not a custom profile ...
                                                if (sUseProfile != "LEVEL") and (sUseProfile != "ACTIVE_OCC") and (sUseProfile != "CUSTOM") and
(sUseProfile != "ARRIVAL"):

                                                        if daytype == 'weekday':
                                                                dayflag = 0
                                                        else:
                                                                dayflag = 1
                                                        [activity_days] = [activity_stats[:][x] for x in numpy.where(activity_stats['f0']==dayflag)]
                                                        [activity_occs] = [activity_days[:][x] for x in
numpy.where(activity_days['f1']==iActiveOccupants)]
                                                        [activity_use_profile] = [activity_occs[:][x] for x in
numpy.where(activity_occs['f2']==sUseProfile)] # sUseProfile = appliances[16] = activity type (string)
                                                        dActivityProbability = activity_use_profile[0][iTenMinuteCount+3] # Get the probability for this
activity profile for this time step
                                                        # For electric space heaters ... (excluding night storage heaters)
                                                elif sApplianceType == "ELEC_SPACE_HEATING":
                                                        # If this appliance is an electric space heater, then activity probability is a function of the
month of the year
                                                        dActivityProbability = round(oMonthlyRelativeTemperatureModifier[month],4)

                                                # Check the probability of a start event
                                                if (random() < ((dCalibration) * dActivityProbability)):
                                                        # This is a start event
                                                        [iPower, iCycleTimeLeft,iRestartDelayTimeLeft] = StartAppliance(iRestartDelay, iMeanCycleLength,
iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType)

                                        # Custom appliance handler: storage heaters have a simple representation
                                        elif sUseProfile == "CUSTOM" and sApplianceType == "STORAGE_HEATER":

                                                # The number of cycles (one per day) set out in the calibration sheet
                                                # is used to determine whether the storage heater is used
                                                # This model does not account for the changes in the Economy 7 time
                                                # It assumes that the time starts at 00:30 each day
                                                if iTenMinuteCount == 4:  # ie. 00:30 - 00:40
                                                        #Assume January 14th is the coldest day of the year
                                                        #Dim oDate, oDateOn, oDateOff As Date
                                                        #Dim monthOn, monthOff As Integer
                                                        #oDate = #1/14/1997#
                                                        # Get the month and day when the storage heaters are turned on and off, using the number of
cycles per year
                                                        monthOff = (14+(iCyclesPerYear/2))/4.3
                                                        monthOn = (365+14+(0-iCyclesPerYear/2))/4.3

                                                        # If this is a month in which the appliance is turned on of off
                                                        if month == monthOff or month == monthOn:
                                                                # Pick a 50% chance since this month has only a month of year resolution
                                                                dProbability = 0.5 / 10  # (since there are 10 minutes in this period)
                                                        elif month > monthOff and month < monthOn:
                                                                # The appliance is not used in summer
                                                                dProbability = 0
```

```
                                                                        else:
                                                                                # The appliance is used in winter
                                                                                dProbability = 1

                                                                        # Determine if a start event occurs
                                                                        if random() <= dProbability:
                                                                                # This is a start event
                                                                                [iPower, iCycleTimeLeft,iRestartDelayTimeLeft] = StartAppliance(iRestartDelay,
iMeanCycleLength, iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType)

                                                else:
                                                        # The appliance is on - if the occupants become inactive, switch off the appliance
                                                        if iActiveOccupants == 0 and sUseProfile != "LEVEL" and sUseProfile != "ACT_LAUNDRY" and sUseProfile != "CUSTOM"
and sUseProfile != "ARRIVAL":
                                                                pass
                                                                # Do nothing. The activity will be completed upon the return of the active occupancy.
                                                                # Note that LEVEL means that the appliance use is not related to active occupancy.
                                                                # Note also that laundry appliances do not switch off upon a transition to inactive occupancy.
                                                        else:
                                                                # Set the power
                                                                iPower = GetPowerUsage(iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType)
                                                                # Decrement the cycle time left
                                                                iCycleTimeLeft = iCycleTimeLeft - 1

                                                # get lighting data and add to iPower
                                                # Set the appliance power at this time step
                                                        sim_dataP[appliance,iMinute-1] = round(iPower,1)
                                                        #sim_dataP[appliance][iMinute-1] = iPower
                                                        if round(appliances_in_dwelling[appliance][23],2) != 1:
                                                                sim_dataQ[appliance][iMinute-1] = round(iPower *
math.tan(math.acos(round(appliances_in_dwelling[appliance][23],2))),1)
                                                        else:
                                                                sim_dataQ[appliance][iMinute-1] = 0

                                        with open('AppProfiles'+idstring+'.dat', 'a') as f:
                                                writer = csv.writer(f, delimiter =' ',lineterminator='\n')
                                                writer.writerow([i] + ["P"] + [appliances_in_dwelling[appliance][15]] + sim_dataP[appliance][:].tolist())
                                                writer.writerow([i] + ["Q"] + [appliances_in_dwelling[appliance][15]] + sim_dataQ[appliance][:].tolist())

                                sim_data_outputP = numpy.sum(sim_dataP, axis=0)
                                sim_data_outputQ = numpy.sum(sim_dataQ, axis=0)

                                with open('AppProfiles'+idstring+'.dat', 'a') as f:
                                        writer = csv.writer(f, delimiter =' ',lineterminator='')
                                        writer.writerow([i] + ["P"] + ["LIGHTING"] + lighting_demand_data.tolist())

                                for k in range(1440):
                                        sim_data_outputP[k] = sim_data_outputP[k] + lighting_demand_data[k]
                                        sim_data_outputQ[k] = sim_data_outputQ[k] + lighting_demand_data[k]*0.75
                                        # note 0.75 factor represents a mean power factor of 0.8 for lighting.
                                sim_dataP_for_file[i][:] = sim_data_outputP
                                sim_dataQ_for_file[i][:] = sim_data_outputQ
```

```
######################### Un-comment this section to see a plot of each generated profile ######################
                # import matplotlib.pyplot as plt
                # plt.plot(sim_data_outputP)
                # plt.savefig('out_'+idstring+'.png')
#############################################################################################################
        # save sim_data to file here
        Pfile = open('Pfile_'+idstring+'.dat', 'w')
        numpy.savetxt('Pfile_'+idstring+'.dat',sim_dataP_for_file,fmt="%d", delimiter='\t')
        Pfile.close
        Qfile = file('Qfile_'+idstring+'.dat', 'a')
        numpy.savetxt('Qfile_'+idstring+'.dat',sim_dataQ_for_file,fmt="%d", delimiter='\t')
        Qfile.close
        Occfile = file('Occfile_'+idstring+'.dat', 'a')
        numpy.savetxt('Occfile_'+idstring+'.dat',occ_profile_for_file,fmt="%d", delimiter='\t')
        Occfile.close
        Appliancesfile = file('Appliancesfile_'+idstring+'.dat', 'a')
        #ppliancesfile.writelines(["%s\n" % item for item in appliances_in_dwelling_for_file])
        for item in appliances_in_dwelling_for_file:
                for item in item:
                        for item in item:
                                Appliancesfile.writelines("%s\t" % item)
                Appliancesfile.write("\n")
        Appliancesfile.close

def GetPowerUsage(iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType):
        # Some appliances have a custom (variable) power profile depending on the time left
        if sApplianceType == "WASHING_MACHINE" or sApplianceType == "WASHER_DRYER":
                # Calculate the washing cycle time
                if (sApplianceType == "WASHING_MACHINE"):
                        iTotalCycleTime = 138
                else: # (sApplianceType = "WASHER_DRYER")
                        iTotalCycleTime = 198
                # This is an example power profile for an example washing machine
                # This simplistic model is based upon data from personal communication with a major washing maching manufacturer
                if (iTotalCycleTime - iCycleTimeLeft + 1) >=0 and  (iTotalCycleTime - iCycleTimeLeft + 1) <=8:
                        return 73         # Start-up and fill
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >=9 and  (iTotalCycleTime - iCycleTimeLeft + 1) <=31:
                        return 2056         # Heating
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >=30 and  (iTotalCycleTime - iCycleTimeLeft + 1) <=92:
                        return 73         # Wash and drain and spin
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >=93 and  (iTotalCycleTime - iCycleTimeLeft + 1) <=94:
                        return 250        # Rinse
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >=95 and  (iTotalCycleTime - iCycleTimeLeft + 1) <=105:
                        return 73         # Spin
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >=106 and  (iTotalCycleTime - iCycleTimeLeft + 1) <=107:
                        return 250         # Rinse
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >= 108 and  (iTotalCycleTime - iCycleTimeLeft + 1) <= 118:
                        return 73         # Spin
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >= 119 and  (iTotalCycleTime - iCycleTimeLeft + 1) <= 120:
                        return 250        # Rinse
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >= 121 and  (iTotalCycleTime - iCycleTimeLeft + 1) <= 131:
                        return 73         # Spin
```

```
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >= 132 and  (iTotalCycleTime - iCycleTimeLeft + 1) <= 133:
                        return 250        # Rinse
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >= 134 and  (iTotalCycleTime - iCycleTimeLeft + 1) <= 138:
                        return 568        # Fast spin
                elif (iTotalCycleTime - iCycleTimeLeft + 1) >=139 and  (iTotalCycleTime - iCycleTimeLeft + 1) <=198:
                        return 2500         # Drying cycle
                else:
                        return iStandbyPower
        else: #(appliance is not a washing machine
                # Set the return power to the rated power
                return iRatedPower

def StartAppliance(iRestartDelay, iMeanCycleLength, iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType):
        iCycleTimeLeft = CycleLength(iMeanCycleLength,sApplianceType)
        iRestartDelayTimeLeft = iRestartDelay
        iPower = GetPowerUsage(iCycleTimeLeft,iRatedPower, iStandbyPower, sApplianceType)
        iCycleTimeLeft = iCycleTimeLeft - 1
        return [iPower, iCycleTimeLeft,iRestartDelayTimeLeft]

def CycleLength(iMeanCycleLength,sApplianceType):
        from math import log
        from random import random
        # Use the TV watching length data approximation, derived from the TUS data
        if sApplianceType == "TV1" or sApplianceType == "TV2" or sApplianceType == "TV3":
                # The cycle length is approximated by the following function
                # The avergage viewing time is approximately 73 minutes
                return int(70 * ((0 - log(1 - random()))**1.1))
        elif sApplianceType == "STORAGE_HEATER" or sApplianceType == "ELEC_SPACE_HEATING":
                # Provide some variation on the cycle length of heating appliances
                return GetMonteCarloNormalDistGuess(iMeanCycleLength, iMeanCycleLength/10)
        else:
                return iMeanCycleLength

def get_occ_profile(daytype):
        from random import random
        import numpy
        household_size = get_household_size()
        iCurrentState = get_start_state(daytype, household_size)
        # Step 3: Determine the active occupancy transitions for each ten minute period of the day
        tpm = get_transistion_probability_matrix(household_size, daytype)
        occ_sim_data = numpy.zeros([144])
        # work out the transition steps
        for iTimeStep in range(0,143):
                # Get a random number
                fRand = random()
                # Reset the cumulative probability count
                fCumulativeP = 0
                # Cycle through the probabilities for this state
                for i in range(0,7):
                        # Add this probability
                        fCumulativeP = fCumulativeP + tpm[iTimeStep*7+iCurrentState][i+2]
                        # See if this is a state transition
```

```
                        if fRand < fCumulativeP:
                                # Transition to another or same state
                                iCurrentState = i
                                # Store the next state
                                occ_sim_data[iTimeStep] = iCurrentState
                                break
        return occ_sim_data

def get_transistion_probability_matrix(household_size, daytype):
        import numpy
    #load tranistion probability matix
        if daytype == "weekday":
                if household_size == 1:
                        return numpy.genfromtxt('tpm1_wd.dat',skip_header=22,dtype=(None))
                elif household_size == 2:
                        return numpy.genfromtxt('tpm2_wd.dat',skip_header=22,dtype=(None))
                elif household_size == 3:
                        return numpy.genfromtxt('tpm3_wd.dat',skip_header=22,dtype=(None))
                elif household_size == 4:
                        return numpy.genfromtxt('tpm4_wd.dat',skip_header=22,dtype=(None))
                else:
                        return numpy.genfromtxt('tpm5_wd.dat',skip_header=22,dtype=(None))
        else:
                if household_size == 1:
                        return numpy.genfromtxt('tpm1_we.dat',skip_header=22,dtype=(None))
                elif household_size == 2:
                        return numpy.genfromtxt('tpm2_we.dat',skip_header=22,dtype=(None))
                elif household_size == 3:
                        return numpy.genfromtxt('tpm3_we.dat',skip_header=22,dtype=(None))
                elif household_size == 4:
                        return numpy.genfromtxt('tpm4_we.dat',skip_header=22,dtype=(None))
                else:
                        return numpy.genfromtxt('tpm5_we.dat',skip_header=22,dtype=(None))

def get_start_state(daytype,household_size):
        from random import random
        import numpy


        if daytype == 'weekday':
                start_states = numpy.genfromtxt('weekday_start_states.dat',skip_header=21,dtype=(None))
        else:
                start_states = numpy.genfromtxt('weekend_start_states.dat',skip_header=21,dtype=(None))

        # Pick a random number to determine the start state
        fRand = random()
        iCurrentstate = 0
        # Reset the cumulative probability count
        fCumulativeP = 0
        # Determine the start state at time 00:00 by checking the random number against the distribution
        for iCurrentState in range(0,6):
                # Add the probability for this number of active occupants
```

```
                fCumulativeP = fCumulativeP + start_states[iCurrentstate][household_size-1]
                if fRand <= fCumulativeP:
                        # This is the start state
                        return iCurrentState

def get_household_size():
        # returns randomly generated household size based on ONS statistics, by LT
        from random import random
        randno = random()
        if randno < 0.294:
                return 1
        elif randno < 0.640:
                return 2
        elif randno < 0.800:
                return 3
        elif randno < 0.932:
                return 4
        else:
                return 5

def gettemperaturedata():
        # Data derived from MetOffice temperature data for the Midlands in 2007 (http://www.metoffice.gov.uk/climate/uk/2007/) Crown Copyright
        return numpy.array([0, 1.63, 1.821, 1.595, 0.867, 0.763, 0.191, 0.156, 0.087, 0.399, 0.936, 1.561, 1.994])

def ConfigureAppliancesInDwelling(appliances):
        from random import random
        appliances_to_remove = []
        appliances_in_dwelling = appliances
        # For each appliance
        for i in range(0,33):
                # Get a random number
                dRan = random()
                # Get the proportion of houses with this appliance
                dProportion = round(appliances[i][0],3)
                # Determine if this simulated house has this appliance
                if dRan > dProportion:
                        appliances_to_remove.append(i)
        appliances_in_dwelling = numpy.delete(appliances_in_dwelling,appliances_to_remove,0)
        return appliances_in_dwelling

def GetMonteCarloNormalDistGuess(dMean, dSD):
        from random import random
        from math import exp
    # Guess a value from a normal distribution for a given mean and standard deviation
        if dMean == 0:
                return 0
        while 1:
                # Guess a value
                iGuess = (random() * (dSD * 8)) - (dSD * 4) + dMean
                # See if this is likely
                px = (1 / (dSD * ((2 * 3.14159) ** 0.5))) * exp(-((iGuess - dMean) ** 2) / (2 * dSD * dSD))
                # End the loop if this value is okay
```

```
                if (px >= random()):
                        return iGuess

def RunLightingSimulation(month,occ_profile):
        from random import randint
        from random import random
        import math
        import numpy
        Ext_ext_glob_irr_threshold_mean = 60 # House external global irradiance threshold mean [W/m^2]
        Ext_ext_glob_irr_threshold_sd = 10 # House external global irradiance threshold standard deviation [W/m^2]
        # Determine the irradiance threshold of this house
        iIrradianceThreshold = GetMonteCarloNormalDistGuess(Ext_ext_glob_irr_threshold_mean, Ext_ext_glob_irr_threshold_sd)
        # Choose a random house from the list of 100 provided in the bulbs sheet
        iRandomHouse = randint(0,99)
        bulbs = numpy.genfromtxt('bulbs.dat',skip_header=14,delimiter='\t',missing_values="",filling_values="0",dtype=(None))
        # Get the bulb data
        vBulbArray = bulbs[iRandomHouse][:]
        # Get the number of bulbs
        iNumBulbs = vBulbArray[1]
        # Declare an array to store the simulation data
        lighting_demand_data = numpy.zeros([iNumBulbs,1440])
        # Load the irradiance array
        vIrradianceArray = numpy.genfromtxt('irradiance.dat',skip_header=25,usecols=month-1,dtype=(None))
        #This calibration scaler is used to calibrate the model to so that it provides a particular average output over a large number of runs.
        fCalibrationScalar = 0.00815368639667705
        # For each bulb
        for i in range(2,iNumBulbs):
                # Get the bulb rating
                iRating = vBulbArray[i]
                # Assign a random bulb use weighting to this bulb
                # Note that the calibration scalar is multiplied here to save processing time later
                fCalibratedRelativeUseWeighting = -fCalibrationScalar * math.log(random())
                #vSimulationArray(3, i) = fCalibratedRelativeUseWeighting
                # Calculate the bulb usage at each minute of the day:
                iTime = 1
                while (iTime <= 1440):
                        # Is this bulb switched on to start with?
                        # This concept is not implemented in this example.
                        # The simplified assumption is that all bulbs are off to start with.

                        # Get the irradiance for this minute
                        iIrradiance = vIrradianceArray[iTime-1]

                        # Get the number of current active occupants for this minute
                        # Convert from 10 minute to 1 minute resolution
                        iActiveOccupants = occ_profile[((iTime)/10)-1]
                        # Determine if the bulb switch-on condition is passed
                        # ie. Insuffient irradiance and at least one active occupant
                        # There is a 5% chance of switch on event if the irradiance is above the threshold
                        bLowIrradiance = ((iIrradiance < iIrradianceThreshold) or (random() < 0.05))

                        # Effective occupancy represents the sharing of light use.
```

```
# Derived from; U.S. Department of Energy, Energy Information Administration, 1993 Residential Energy Consumption Survey,
# Mean Annual Electricity Consumption for Lighting, by Family Income by Number of Household Members
fEffectiveOccupancyArray = numpy.array([0.0,1.0,1.528,1.694,1.983,2.094])
# Get the effective occupancy for this number of active occupants to allow for sharing
fEffectiveOccupancy = fEffectiveOccupancyArray[iActiveOccupants]
iLightDuration =0
# Check the probability of a switch on at this time
if (bLowIrradiance and (random() < (fEffectiveOccupancy * fCalibratedRelativeUseWeighting))):
        # This is a switch on event
        # Determine how long this bulb is on for
        r1 = random()

        # below taken from the lighting event duration model on the CREST model light_config tab
        # This model defines how long a bulb will stay on for, if a switch-on event occurs.
        # Original source: M. Stokes, M. Rylatt, K. Lomas, A simple model of domestic lighting demand, Energy and Buildings 36 (2004) 103-
116
        event_duration_lower_values = numpy.array([1,2,3,5,9,17,28,50,92])
        # taken from  CREST model light_config tab (cells C55:E63)
        event_duration_upper_values = numpy.array([1,2,4,8,16,27,49,91,259])
        # taken from CREST model light_config tab (cells D55:E63)

        for j in range(1,9):
                # Get the cumulative probability of this duration
                cml = j/9.0 # Note default settings in CREST model light_config tab (cells E55:E63) have this relation
                # Check to see if this is the type of light
                if r1 < cml:

                        # Get the durations
                        iLowerDuration = event_duration_lower_values[j-1]
                        iUpperDuration = event_duration_upper_values[j-1]
                        # Get another random number
                        r2 = random()
                        # Guess a duration in this range
                        iLightDuration = int((r2 * (iUpperDuration - iLowerDuration)) + iLowerDuration)
                        # Exit the loop
                        break
        for j in range(1,iLightDuration):
        # Range check
                if iTime > 1440:
                        break
        # Get the number of current active occupants for this minute
                iActiveOccupants = occ_profile[((iTime - 1)/10)]
                # If there are no active occupants, turn off the light
                if iActiveOccupants == 0:
                        break
        # Store the demand
                lighting_demand_data[i,iTime] = iRating
# Increment the time
                iTime = iTime + 1
else:
        # The bulb remains off
        lighting_demand_data[i,iTime-1] =  0
```

```
                            # Increment the time
                            iTime = iTime + 1
        # return the simulation data
        return numpy.sum(lighting_demand_data, axis=0)
```

Chapter 4 Code – adding PV to demand profiles

```
import numpy
month = $MONTH # 1,3,5,7,9 and 11 used
daytype = 'weekday'
no_its = 3840
PV_proportion = [0,10,20,30,40,50,60,70,80,90,100] #should be a list
from random import random
import math
import numpy as np
import csv
import matplotlib.pyplot as plt
files = ['$INFILE1','$INFILE2']  # P and Q profiles for 0PV

Panel_radiation = np.genfromtxt('PVgen.dat',skip_header=63,dtype=(None)) # Data generated using Richardson's model,
# see Chapter 4
panel_efficiency = 0.1
panel_size = 10
P_min= np.genfromtxt('Pfile_'+repr(3840) + 'x_' + 'month-' + str(month) + '_daytype-' + daytype + '_PVprop-0.dat')
# get previously generated demand from file, with no PV.
P_min2 = np.zeros((P_min.shape))
NoOfProfs = len(P_min)
PVprofile= Panel_radiation[month-1]#*panel_efficiency*panel_size# commented out as =1
PVhouses = np.arange(0,NoOfProfs)
PVhouses = np.random.permutation(PVhouses)
np.savetxt('PVhouses.dat',PVhouses)
for prop in PV_proportion:
      P_min2[:]=P_min[:]
      noPVhouses = int(NoOfProfs*prop/100)
      print noPVhouses
      idstring = str(no_its) + 'x_' + 'month-' + str(month) + '_' + 'daytype-' + str(daytype)+ '_PVprop-' +
repr(int(prop))
      for x in PVhouses[0:noPVhouses]:
```

```
            P_min2[x]=P_min[x]- PVprofile[:]
        Pfile = open('Pfile_'+idstring+'.dat', 'w')
        np.savetxt('Pfile_'+idstring+'.dat',P_min2,fmt="%d", delimiter='\t')
        Pfile.close
```

Chapter 4 Code - Stochastic phase configuration creation

```
def pconf(nonodes,p1conlik):
        import numpy as np
    phase1, phase2, phase3 = 0,0,0
    phaseconfig = np.ones((1,3,nonodes))
    for p in range(nonodes):
        for n in range(0,3):
            x= np.random.rand()
            #print x
            if x<p1conlik:
                phase1 = phase1+1
            elif x < p1conlik+(1-p1conlik)/2:
                phase2 = phase2+1
            else:
                phase3 = phase3+1
        phaseconfig[:,:,p]=np.array([phase1,phase2,phase3])
        phase1 =0
        phase2 =0
        phase3 =0
    return phaseconfig[0,:,:]
```

Chapter 4 Code - Running unbalanced load flow

```
nofeeders =200
daytype = 'weekday'
PVlist = [$PVPen]
monthlist = [$MONTH]
phaseconf = np.zeros((200,32,3))
x=0
y=0
nomins=1440
output = np.zeros((200,nomins,3,13,32),dtype=complex)

pcc=np.load('phaseconfigurations.npz')
phaseconf[:] = pcc['pc']
mcount=0

for m in monthlist:
        for PV in PVlist:
                Pidstring = str(96*40) + 'x_' + 'month-' + str(m) + '_' + 'daytype-' + str(daytype)+ '_PVprop-' + repr(PV)
                Qidstring = str(96*40) + 'x_' + 'month-' + str(m) + '_' + 'daytype-' + str(daytype)+ '_PVprop-0'
                P= np.genfromtxt('Pfile_'+Pidstring+'.dat', delimiter='\t')
                Q= np.genfromtxt('Qfile_'+Qidstring+'.dat', delimiter='\t')


                x=0
                P_L1=P[x:x+32]
                Q_L1=Q[x:x+32]
                P_L2=P[x+32:x+64]
                Q_L2=Q[x+32:x+64]
                P_L3=P[x+64:x+96]
                Q_L3=Q[x+64:x+96]
                for feeder in range(nofeeders):

                        pc = phaseconf[feeder,:,:]

                        x=int(feeder*96)
                        for min in range(nomins):
                                out=l.lvsolve(243.75,P_L1[:,min],Q_L1[:,min],P_L2[:,min],Q_L2[:,min],P_L3[:,min],Q_L3[:,min],pc)
                                if out == 'ERROR':
                                        print 'feeder: ' + repr(feeder)
                                        print 'PV: ' + repr(PV)
                                        print 'month: ' + repr(m)
                                        print 'x: ' + repr(x)
                                        print 'load flow failed\n'
                                        print 'min = ' + repr(min)
                                        print pc
```

```
                                print P_L1
                                print Q_L1
                                print P_L2
                                print Q_L2
                                print P_L3
                                print Q_L3
                                print '\**************************n\n\n'
                                output[feeder,min,:,:,:] = outprev
                        else:
                                if out == []:
                                        print 'empty output'
                                        print 'feeder: ' + repr(feeder)
                                        print 'PV: ' + repr(PV)
                                        print 'month: ' + repr(m)
                                        print 'x: ' + repr(x)
                                        print 'P_L1: '
                                        print P_L1[:,min]
                                        print 'P_L3: '
                                        print P_L3[:,min]
                                        print 'min'
                                        print min
                                        # print 'out'
                                        # print out
                                        print 'pc'
                                        print pc
                                        print '**************************\n\n\n'
                                output[feeder,min,:,:,:] = out
                                outprev = out
                output[feeder,0,2,1,:] = 100*np.sum(pc,axis=0)[np.sum(pc,axis=0).argmax()]/(96)#highest prop on phase

                output[feeder,0,2,2,:] = 100*np.sum(pc,axis=0)[np.sum(pc,axis=0).argmax()]/(96) #weighted highest prop on phase
        np.savez_compressed('volts243_'+str(96*nofeeders) + 'x_' + 'month-' + str(m) + '_' + 'daytype-' + str(daytype)+ '_PVprop-' +
repr(PV),output,phaseconf)


    mcount=mcount+1
```

Chapter 4 Code – Post processing – Calculating voltage maxima and minima

```
import numpy as np
nonodes = 32
nofeeders = 200
months = [$MNTH]
daytype = 'weekday'
PVPen = [$PVPEN]
nofeeders=200
maximaV=np.zeros((len(PVPen),nofeeders))
minimaV=np.zeros((len(PVPen),nofeeders))
maximaP=np.zeros((len(PVPen),nofeeders))
minimaP=np.zeros((len(PVPen),nofeeders))
PVno =0
feederno=0
for m in months:
      PVno =0
      for PV in PVPen:
            feederno = 0
            try:
                  a=np.load('volts243_19200x_month-'+repr(m)+'_daytype-'+daytype+'_PVprop-
'+repr(PV)+'.npz')
                  feeders = a['arr_1']
            except:
                  print "load failed:"
                  print 'volts243_19200x_month-'+repr(m)+'_daytype-'+daytype+'_PVprop-'+repr(PV)+'.npz\n'
            for feeder in feeders:
                  V=np.absolute(a['arr_0'][feederno,:,:,9,:])
                  maximaV[PVno,feederno] = np.around([np.max(V)],decimals=2)
                  minimaV[PVno,feederno] = np.around([np.min(V)],decimals=2)
                  feederno+=1
            PVno = PVno+1
      np.savez_compressed('maxs243_'+repr(m)+repr(PVPen[0]),maximaV)
      np.savez_compressed('mins243_'+repr(m)+repr(PVPen[0]),minimaV)
```

A5 Appendix 5 - Why normalisation was performed

An example meter half-hourly voltage profile is shown in Figure 8.23. The lower section of Figure 8.23 shows how the voltage profile can be represented in greyscale – where the colour is varied from black (at the minimum voltage) to white (at the maximum voltage).

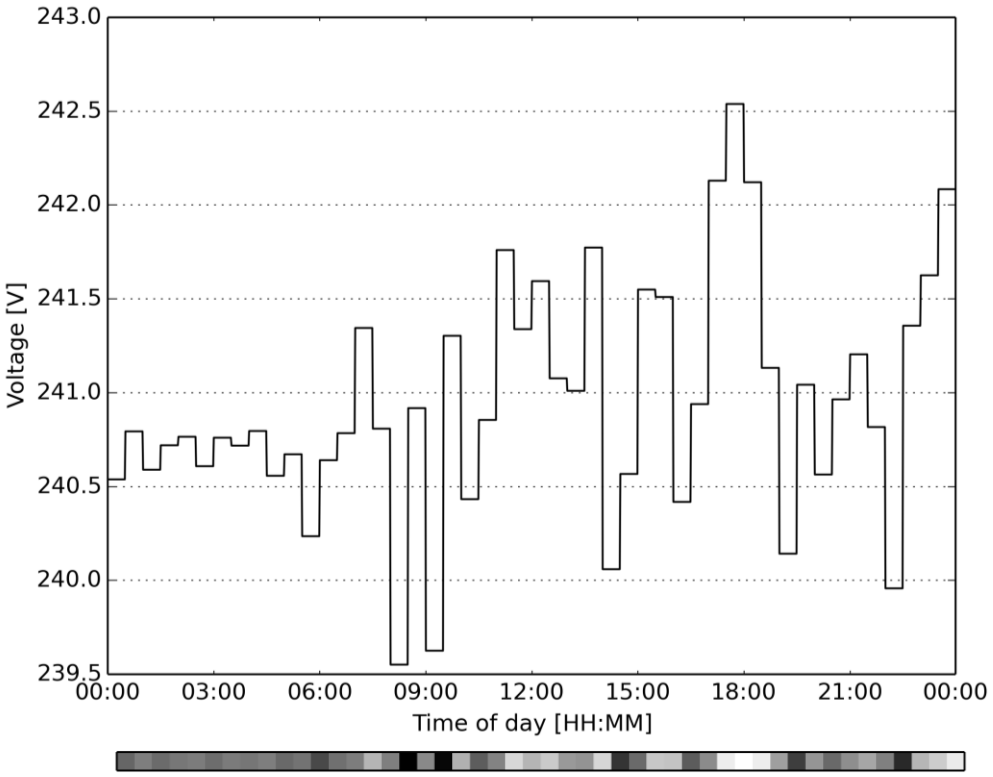


Figure A5.1 - Above - an example mean half-hourly RMS voltage for a smart meter with the voltage profile represented in greyscale

To see why normalisation of the profiles (scaling them so that the maximum value is 1 and the minimum value is 0) is done see Figures A5.2 and A5.3. These figures show stacked greyscale voltage profiles (of the sort introduced in Figure A5.1). In both Figures A5.2 and A5.3, the left hand side represents 96 unsorted profiles and the right shows them when sorted by phase. Note that these Figures are intended to show the importance of normalising the values - they are sorted based on a priori knowledge of the phase groupings. In the sorted profiles, three clear groups, representing the three phases, can be made out by eye. Also, similarities in the patterns of vertical stripes (representing peaks and troughs in similar voltage profiles) can be made out. Notice that the continuation of the vertical stripes is clearer in the

normalised profiles than in the non-normalised profiles. The voltage profiles in each phase grouping tend to become more similar to each other and more distinct from profiles in the other phase groupings – it is for this reason that the profiles are normalised.

Normalisation tends to make voltages profiles on different phases less similar as it reduces similarities based on impedance (i.e. distance) between the substation and houses. For example, one might expect voltages at households nearer to the substation to be higher than those that are far away, regardless of phase connection. The intention is to remove these similarities so that the profiles are grouped on the position of the peaks and troughs (which are more related to the phase connection) as opposed to the mean magnitude (which can relate to both the phase connection and the distance to the substation).

Note also the answers at StackOverflow.com, suggesting normalisation (and clustering), to a question by the author of this thesis [222].

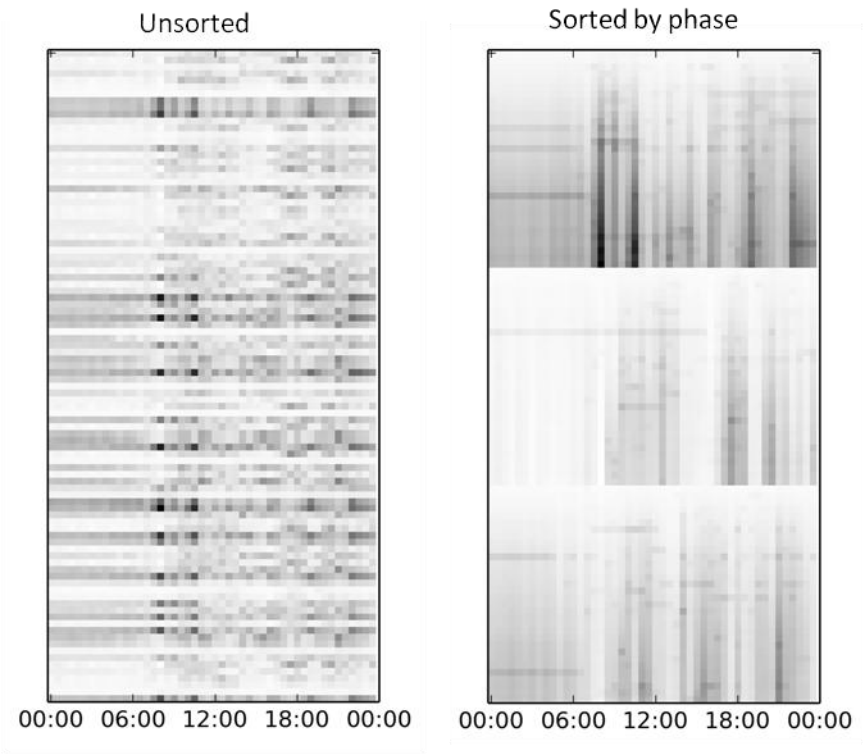


Figure A5.2 Mean half-hourly voltage levels for 96 smart meter connected to the same feeder. (Dark grey represents relatively high voltages)

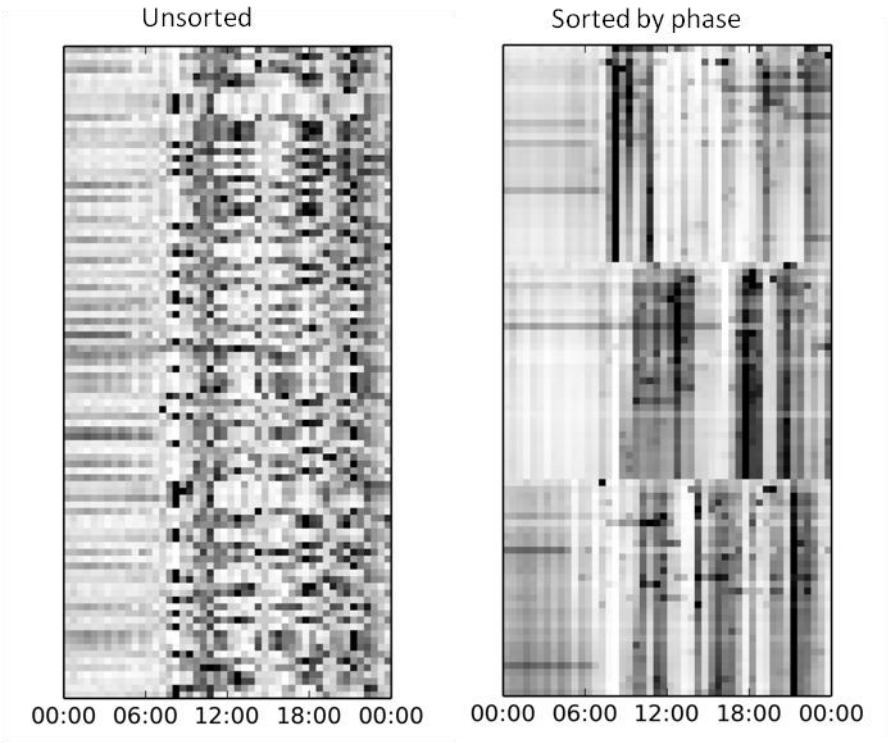


Figure A5.3 - Mean half-hourly normalised voltage levels for 96 smart meters connected to the same feeder.

A6 Chapter 5 Code

Chapter 5 Code - Post-processing - Calculating time of voltage minima and maxima

```
import numpy as np
nonodes = 32
nofeeders = 200
months = [$MNTH]
daytype = 'weekday'
PVPen = [0,10,20,30,40,50,60,70,80,90,100]
nofeeders=200

whemax=np.zeros((nofeeders,len(PVPen),3),dtype=int)
whemin=np.zeros((nofeeders,len(PVPen),3),dtype=int)
PVno =0
feederno=0

for m in months:
      PVno =0
      for PV in PVPen:
            feederno = 0
            #print 'volts243_19200x_month-'+repr(m)+'_daytype-'+daytype+'_PVprop-'+repr(PV)+'.npz\n'
            try:
                  a=np.load('volts243_19200x_month-'+repr(m)+'_daytype-'+daytype+'_PVprop-
'+repr(PV)+'.npz')
            except:
                  print "load failed:"
                  print 'volts243_19200x_month-'+repr(m)+'_daytype-'+daytype+'_PVprop-'+repr(PV)+'.npz\n'
            for feeder in a['arr_1']:
                  try:
                        V=np.absolute(a['arr_0'][feederno,:,:,9,:])
                        #print np.where(V==V.max())
```

```
                    whemax[feederno,PVno,:] = np.where(V==V.max())
                    whemin[feederno,PVno,:] = np.where(V==V.min())
               except:
                    print 'where failed'
                    print 'month: ' + str(m) + ' PV: ' + str(PV)


               feederno+=1
          PVno = PVno+1
     np.savez_compressed('where243_'+repr(m),whemax,whemin)
```

Chapter 5 Code - Normalising a voltage profile

```
def normalisebyrow(inarray):
     import numpy as np
     a=np.zeros_like(inarray)
     a[:]=inarray
     row=0
     for x in inarray:
          rowmean=np.max(x)

          a[row]=(x-x.min())/(x-x.min()).max()#(x-x.min()).max()
          row+=1
     return a
```

Chapter 5 Code - The distance between two arrays

```
def getsumsquarediffs2(inarray):
      import numpy as np
      a=np.zeros((inarray.shape[0],inarray.shape[0]))
      xind=0

      for x in inarray:
            yind=0
            for y in inarray:
                  diff = x-y
                  a[xind,yind] = np.sum(np.square(diff))
                  yind+=1
            xind+=1
      return a
```

Chapter 5 Code - The proportion of correctly grouped meters

```
def successfactor(actualgroups,predictedgroups):
      # reads in 2 lists, both of length 3, of lists containing the actual and predicted meter IDs on each
phase. Returns the proportion of correctly grouped meters indicating how closely they match.
      import numpy as np
      import itertools as it
      sf=np.zeros((3,6))
      sfprev=0
      for i in it.permutations(range(0,3),3):
            sf = sfr(actualgroups,[predictedgroups[k] for k in list(i)])
            if sf>sfprev:
                  sfprev=sf
      return sfprev*100
```

Chapter 5 Code - The complete phase identification algorithm

```
def PhaseIDAlg(inarray,pcgroups,imags,itots):
      import numpy as np
      import itertools as it
      import time
      import PhaseIDTester as pid
      reload(pid)
      # Igroupcombs = np.zeros((968,3))
      # Igroupsums = np.zeros((16,))

      b=np.zeros_like(inarray)
      b[:]=inarray #normalised voltages
      a=np.zeros((inarray.shape[0],inarray.shape[0]))
      a[:]=getsumsquarediffs2(inarray)
      meterlist=[[x] for x in range(inarray.shape[0])]
      done=False
      yind=0
      xind=0
      failpointflag=False
      failpoint=97
      while done ==False:

            #1. find closest meters
            xind+=1
            #print np.where(a==a[a!=0].min())
            c=np.where(a==a[a!=0].min())
            #print c[0]
            #print c[1]
            minlocs = [c[0][0],c[0][1]]

            if failpointflag == False:
                  if checkgroup(meterlist[minlocs[0]]+meterlist[minlocs[1]],pcgroups)=='FAILED':
                        # print 'failed at:'
                        # print xind
                        failpoint = xind
                        t1=meterlist[minlocs[0]]
```

```
                        t2=meterlist[minlocs[1]]
                        t1.sort()
                        t2.sort()


      group=((b[minlocs[0]])*len(meterlist[minlocs[0]])+(b[minlocs[1]])*len(meterlist[minlocs[1]]))/(len(meterlist[m
inlocs[0]])+len(meterlist[minlocs[1]]))
                        temp1= b[minlocs[0]]
                        temp2=b[minlocs[1]]
                        failpointflag = True

            if xind ==86: #86
                  Igroupsums = getgroupsums(imags,meterlist)
                  ps = list(binner(Igroupsums))


                  ps_inds = list(binner(range(len(Igroupsums))))
                  # find the member of power set that is closest to smallest sum
                  iprev=np.inf
                  ind=0
                  itarg = itots# was min
                  for p in ps:
                        isum=np.array([np.sum(xs) for xs in p])

                        if np.sum(np.abs(isum-itarg))<iprev:
                              iprev=np.sum(np.abs(isum-itarg))
                              iind=ind
                        ind+=1
                  meterlist1 = [list(it.chain.from_iterable([meterlist[x] for x in ps_inds[iind][y]])) for y in
range(3)]
                  meterlist=meterlist1
                  done=True
            else:
```

```
      group=((b[minlocs[0]])*len(meterlist[minlocs[0]])+(b[minlocs[1]])*len(meterlist[minlocs[1]]))/(len(meterlist[m
inlocs[0]])+len(meterlist[minlocs[1]]))
                  temp1= b[minlocs[0]]
                  temp2=b[minlocs[1]]

                  b=np.delete(b,[minlocs[0],minlocs[1]],0)

                  b=np.vstack((b,group))
                  a=getsumsquarediffs2(b)

                  metermins = []
                  metermins=metermins+meterlist[minlocs[0]]+meterlist[minlocs[1]]

                  minlocs.sort()
                  meterlist.pop(minlocs[0])
                  meterlist.pop(minlocs[1]-1)
                  meterlist=meterlist+[metermins]

                  if len(meterlist) <4:
                        done=True
                  else:
                        pass

      meterlist[0].sort()
      meterlist[1].sort()
      meterlist[2].sort()
      sf=pid.successfactor(pcgroups,meterlist)
      # print 'Group 1 (' +str(len(meterlist[0]))+ ' meters): ' + str(meterlist[0])
      # print 'Group 2 (' +str(len(meterlist[1]))+ ' meters): ' + str(meterlist[1])
      # print 'Group 3 (' +str(len(meterlist[2]))+ ' meters): ' + str(meterlist[2])
      # print 'Success factor: ' + str(sf)
      return sf,failpoint
```

```
def getgroupsums(metersums,meterlist):
      import numpy as np
      Igroupsums = np.zeros((len(meterlist),))
      mind=0
      #metersums= np.sum(imags,axis=0)
      for group in meterlist:
            #print group
            Igroupsums[mind]=np.sum(metersums[meterlist[mind]])
            mind+=1
      #print Igroupsums
      return list(Igroupsums)



def binner(seq, width=3):
      # credit: User DSM at StackOverflow - http://stackoverflow.com/a/30458025/1461850
      import itertools
      for locs in itertools.product(range(width), repeat=len(seq)):
            output = [[] for _ in range(width)]
            for elem, loc in zip(seq, locs):
                  output[loc].append(elem)
            yield output
```

Chapter 5 Code - Array subtraction timing – console input and output

Using an IPython console in Spyder IDE on a Intel® Core i5 CPU M460@2.53Ghz with 2GB RAM running Windows 7.

```
In [1]: import numpy as np
In [2]: a=np.random.random((96,1))
In [3]: b=np.random.random((96,1))
In [4]: a.nbytes
Out[4]: 768
In [5]: b.nbytes
Out[5]: 768
In [6]:%%timeit
   …  :a-b
   …  :
1000000 loops, best of 3: 1.37 µs per loop
```