ISGT LA Tutorial

Advanced Modelling of Smart Distribution Networks Using OpenDSS

Dr Luis (Nando) Ochoa

luis_ochoa@ieee.org

&

Dr Jairo Quiros-Tortos

jairoquirostortos@ieee.org

The University of Manchester, UK

5th October 2015

Dr Luis(Nando) Ochoa

- Senior Lecturer at The University of Manchester
  - IEEE PES Distinguished Lecturer
  - IEEE Senior Member
  - Chair of the IEEE-PES Modern and Future Distribution System Planning WG
  - Chair of the IEEE-PES Innovative Smart Grid Technologies (ISGT) Europe Steering Committee
  - Member of the European Technology Platform SmartGrids WG1
  - 110+ papers top int'l journals and conf, 1 patent (Psymetrix)
  - 40+ technical reports

- Current Post-Graduate Team:
  - 10 PhD Students, 4 Post-Docs

Dr Jairo Quiros-Tortos

- Post-Doctoral Research Associate (Since May 2014)
  - IEEE Member
  - Vice-Chair of the IEEE-PES UK&RI Chapter
  - Lecturer at the University of Costa Rica (Aug. 2008 – Aug. 2010)
  - Invited Professor at the University of Costa Rica (May. 2016 –)
  - 25+ papers top int'l journals and conf, 1 patent (UoM)
  - 5 technical reports
- Areas of expertise
  - Distribution Network Modelling
  - Distributed Low-Carbon Technologies Modelling
  - Distribution Network Analysis, Operation and Planning

Structure of the Tutorial

First Block:

14:05 to 14:25	*"Smart Distribution Networks" → Modelling requirements and challenges*
14:25 to 14:40	*"Introduction to OpenDSS"*
14:40 to 15:20	*"Basic Modelling of MV and LV networks" → Hands on examples*
15:20 to 15:45	*"Interfacing OpenDSS with MS Excel VBA and Matlab" → Hands on examples*

Second Block:

16:15 to 16:45	*Stochastic Impact Analysis of Low Carbon Technologies*
	Low Voltage Network Solutions Project
16:45 to 17:15	*Management of Electric Vehicle Charging Points*
	My Electric Avenue Project
17:15 to 17:25	*Break*
17:25 to 17:45	*Integrated Optimal Volt-Var Control of MV and LV Networks*
	Smart Street Project
17:45 to 18:00	*Wrap up, Conclusions and Q&A*

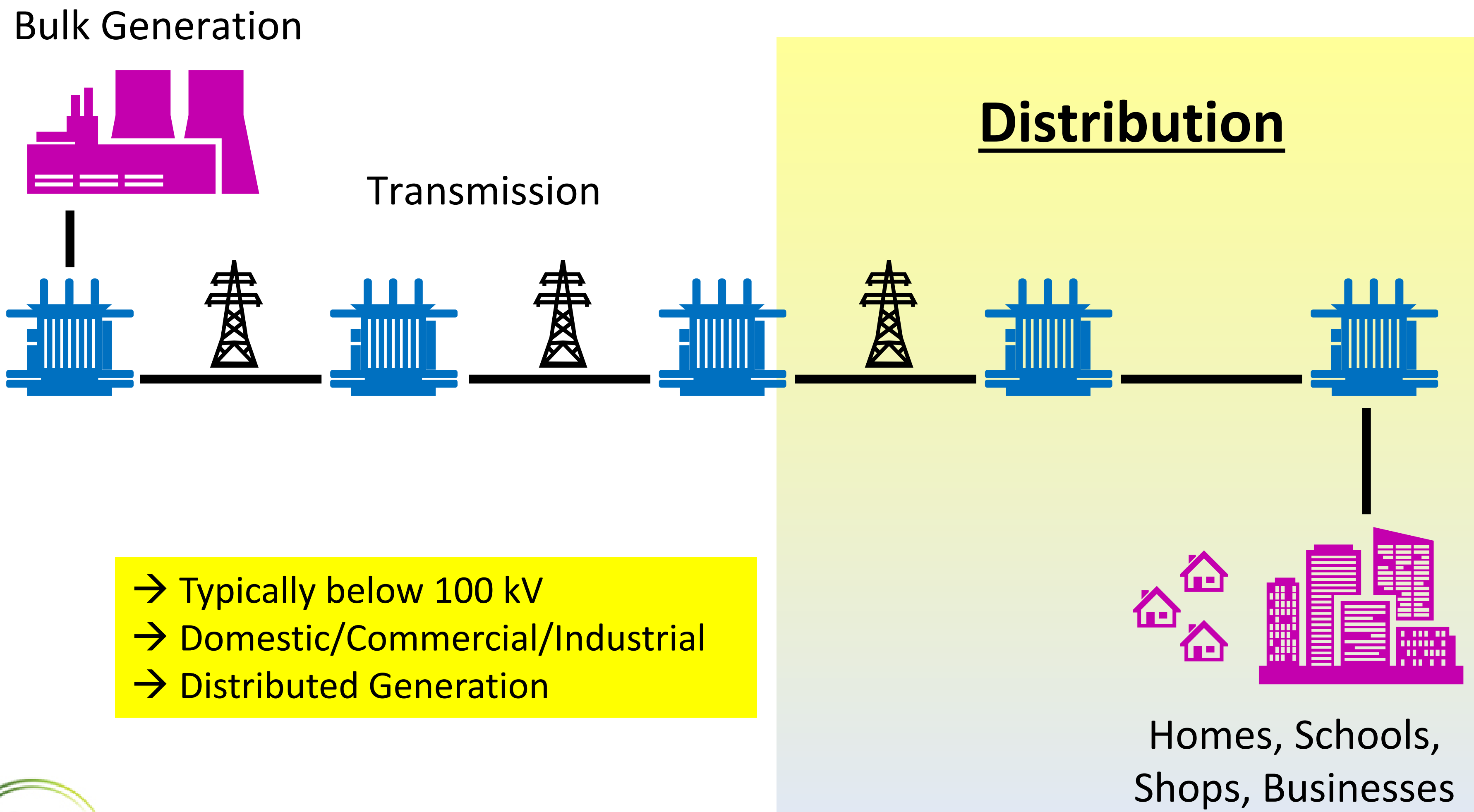
14:05 to 14:25

SMART DISTRIBUTION NETWORKS

Outline – Smart Distribution Networks

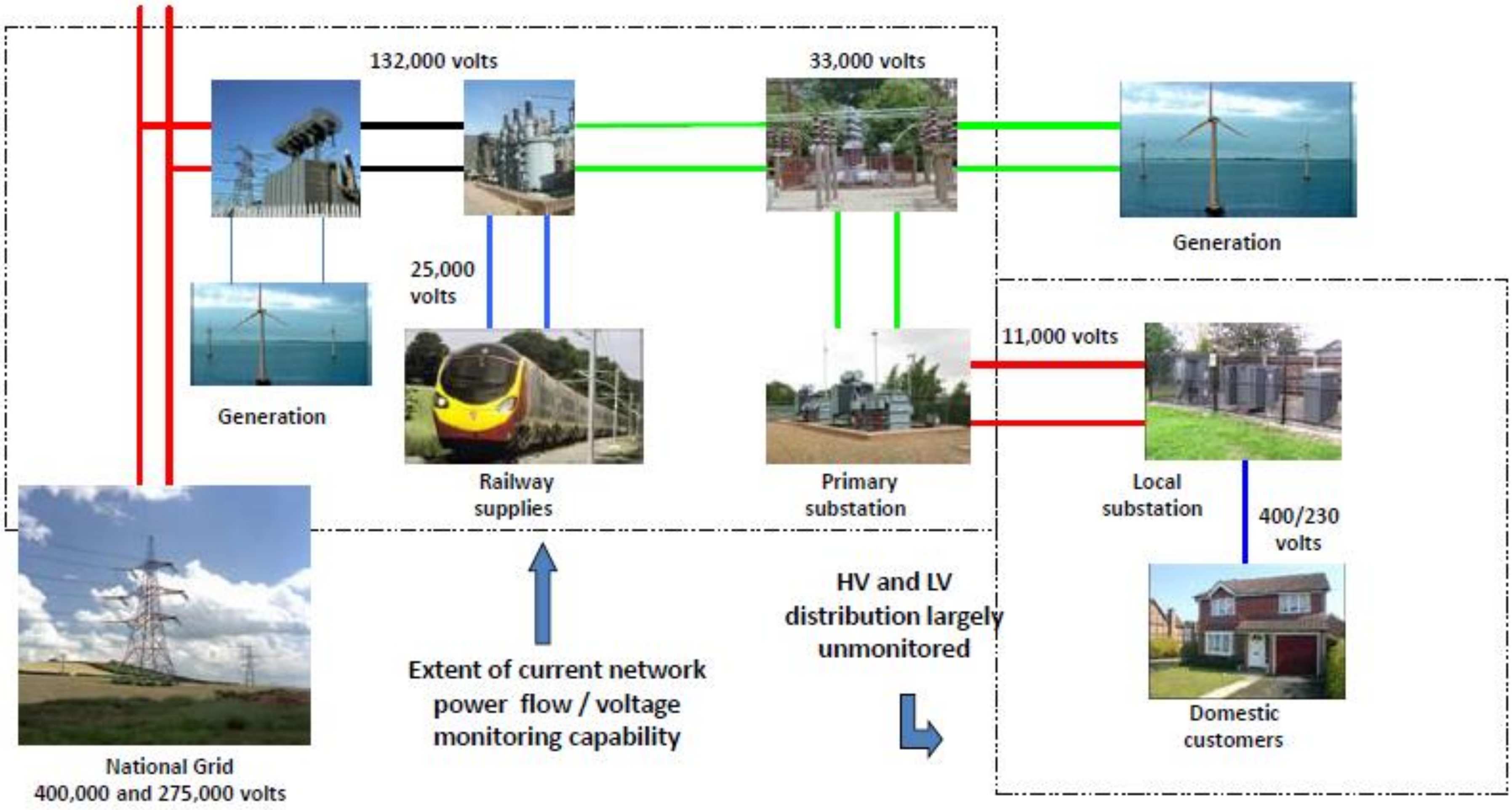
- Introduction to Distribution Networks
- The Context for Distribution (UK)
  - Traditional Electricity Networks → Smart & Low Carbon
- Modelling of Distribution Networks
  - Basic Modelling
  - Towards State-of-the-Art Modelling
- Remarks

Distribution Networks



→ Typically below 100 kV
→ Domestic/Commercial/Industrial
→ Distributed Generation

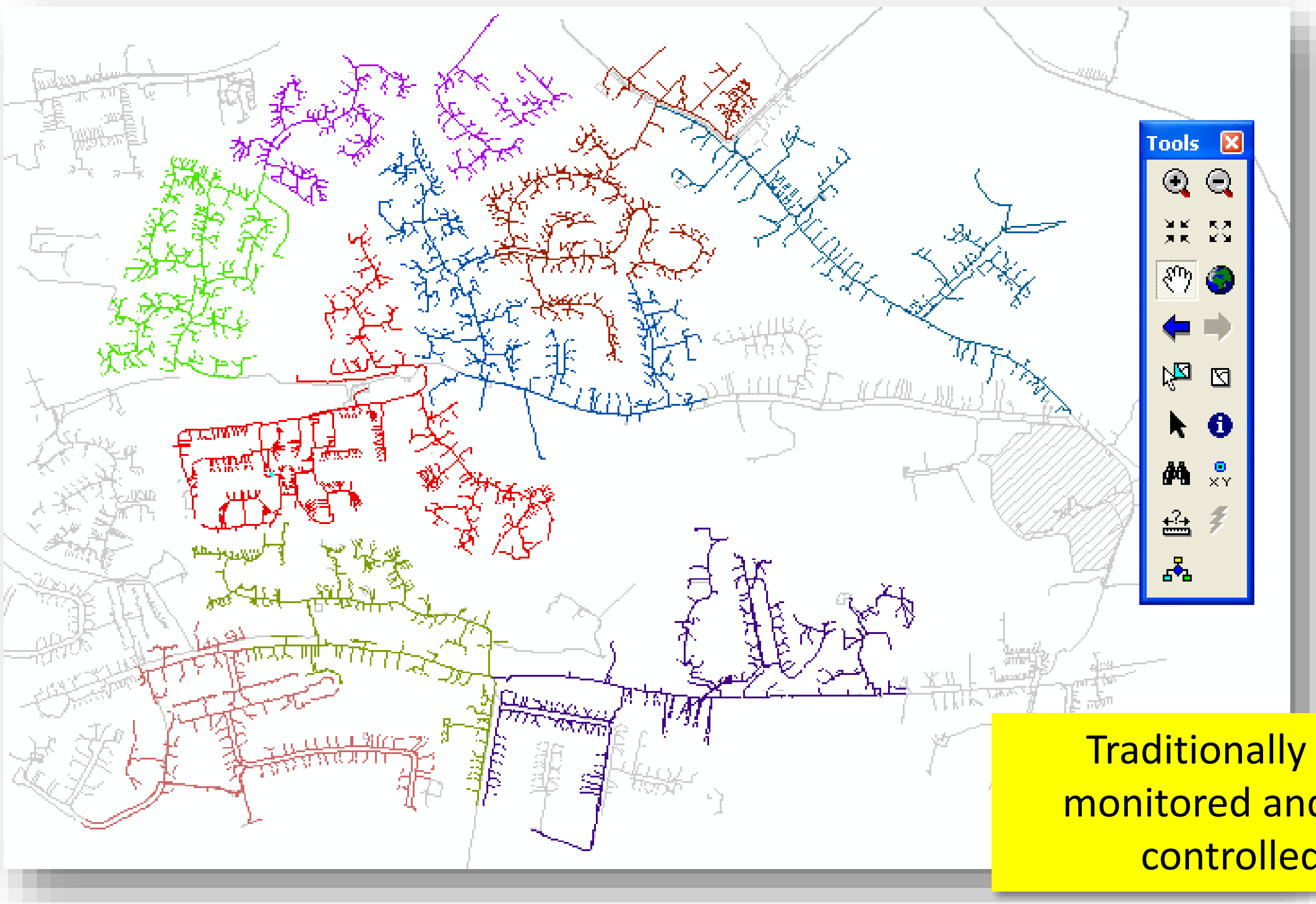
UK Distribution Networks



Source: Energy Networks Association

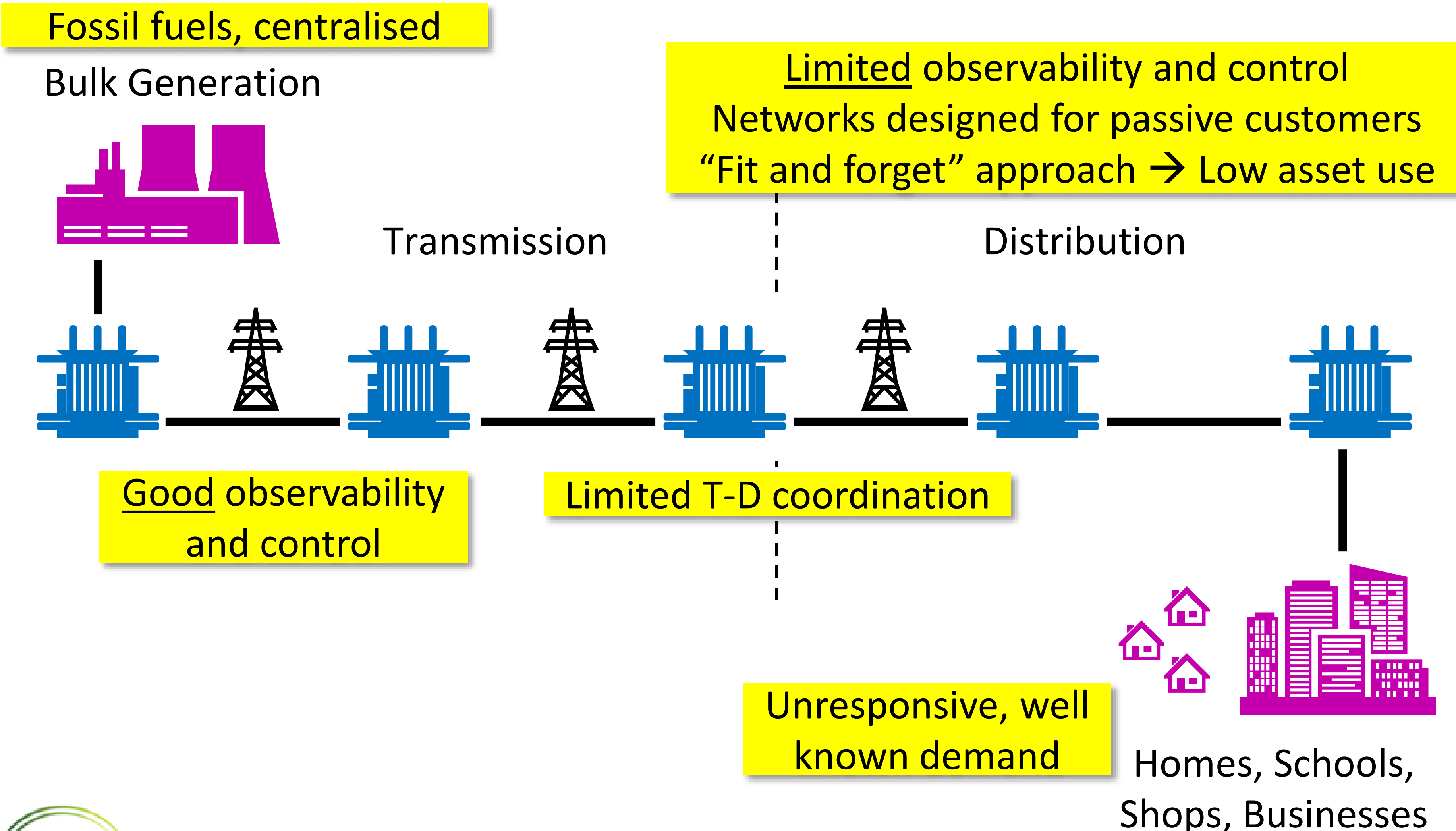
Examples of LV Networks in the UK

- Topology of urban networks (Courtesy of ENWL)

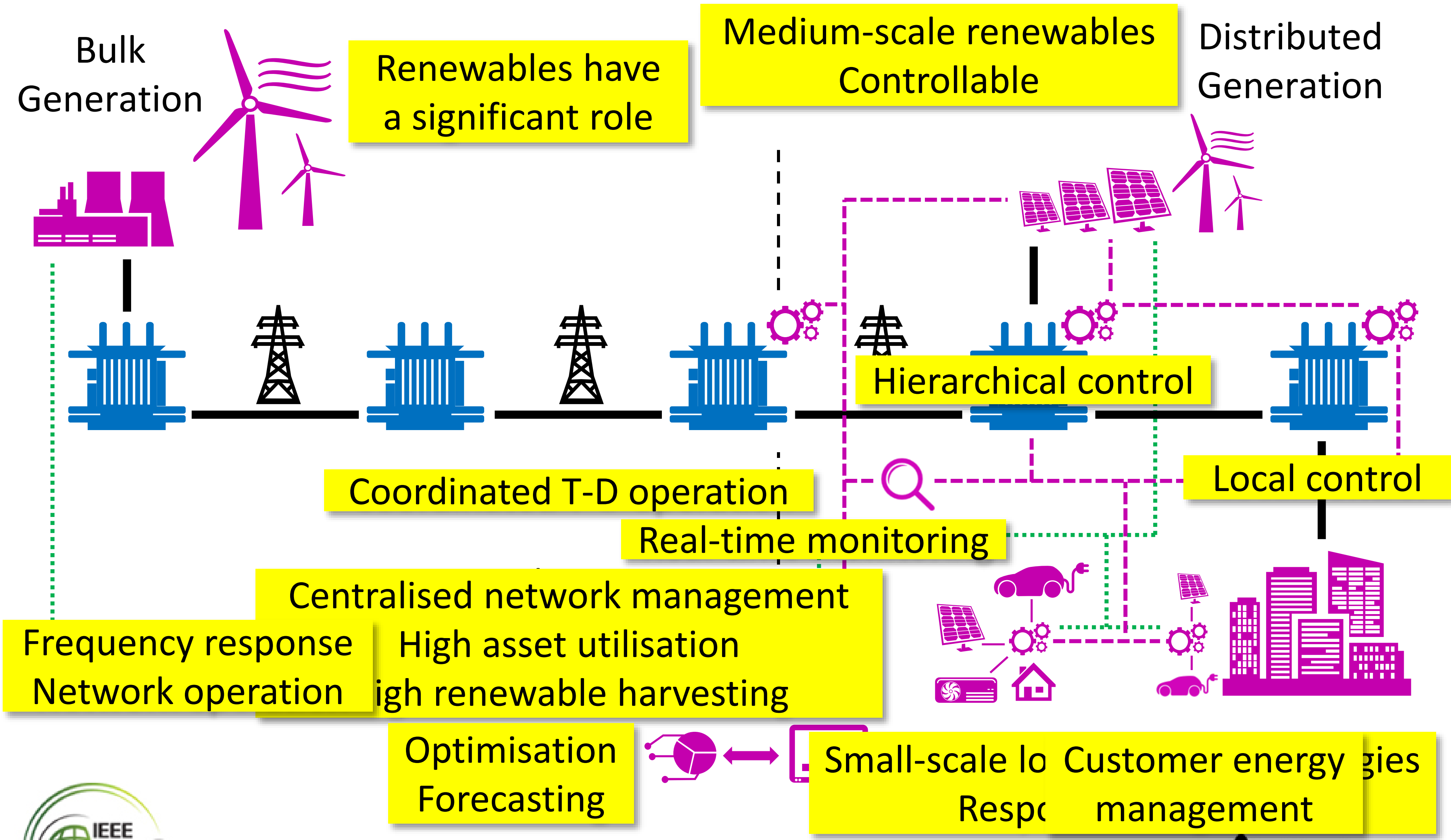


Traditionally not monitored and not controlled

Traditional Electricity Networks



Smart & Low Carbon



(Some) Technical Challenges

- **LV** Distribution **Networks** (< 1kV)
  - Voltage rise/drops due to PV panels/EVs
  - Thermal limits: Are the wires fit for purpose?
  - More unbalances? Harmonics? etc.
- **MV** Distribution **Networks** (> 1kV and < 150kV)
  - Voltage rise due to wind power (rural networks)
  - Increase in short circuit level (urban underground)
  - Power quality, "Islanding" and Protection
  - Thermal limits
  - Increased energy losses? Variability?
  - Stability and reserve requirements

Observability

Controllability

Voltage Management

…

Thermal, Fault Mgmt

Integration of Solutions

Traditional Modelling

1. Typically considering **balanced** networks
   - Not realistic but an **OK approximation for MV** networks (>1kV)
   - **Inadequate when closer to end customers** (LV < 1KV)

2. Typically considering **snapshots or hourly profiles**
   - **Helps simplifying planning problems** (e.g., reconfiguration or location of capacitors for power loss minimisation during peak)
   - **Neglects the actual load and (renewable) generation changes** that can significantly affect the main objectives (e.g., energy efficiency, asset utilisation)
   - Simplified profiles (15, 30, 60 min) **neglects the actual control of network elements** (e.g., OLTCs act in ~1 min)

Traditional Modelling

3. Typically **deterministic**
   - The **uncertainties** due to the variability, location, size of ‘common’ and future **loads** as well as **renewable generation are neglected**
   - This simplifies the problem but **can over or underestimate problems/benefits**

4. Typically **analysing a single voltage level**
   - Considered due to its simplicity (smaller networks)
   - The **impacts** of new technologies on **one voltage level** (e.g., MV) **can be over or underestimated** if neglecting the interdependencies with other voltage levels (e.g., LV or HV)

Traditional Modelling

5. Typically considering a **single simulation platform**
   - Problems, most commonly optimisation ones, tend to provide solutions that do not consider the potential real implementation, i.e., **simulation results finish on one platform**.
   - This simplifies the problem but **can over or underestimate problems/benefits**
   - Can we have multiple hardware/software simulation platforms?
6. Typically considering **only the electricity systems**
   - Again, for simplicity.
   - **ICT** infrastructure (e.g., delays, **effects on the measurements** and hence on control)?
   - **Transport, water, heat, cooling systems and their interdependencies** with electricity?

Towards State-of-the-Art Modelling

1. Balanced → **Unbalanced**
2. Snapshot → **Realistic time-series profiles**
3. Deterministic → **Probabilistic**
4. Single voltage level → **Integrated LV-MV-HV modelling**
5. Single simulation platform → **Co-simulation**
6. Only the electricity system → **Multiple systems**

Much more complexity…
But we have the tools and computational power

Key Remarks – Smart Distribution Networks

- The are many **challenges**… but also potential **solutions**
- 1st major challenge for DNOs is the **lack of observability** of their LV and MV circuits.
- 2nd major challenge: transition from stand alone operation of solutions towards an **integrated DSO approach**
  - Reliable and cost-effective ICT is not trivial
  - Regulatory barriers need to be overcome
- **R&D is essential for deployment**

14:25 to 14:40

INTRODUCTION TO OPENDSS

Acknowledgement

Most slides in this presentation have used and/or adapted content produced by Roger Dugan (EPRI, USA) who has kindly granted the corresponding permission.

Main repository of slides >> ftp://ftp.epri.com/
Other sources >> OpenDSS Training Slides

Important: These slides are meant to be used in conjunction with OpenDSS documentation
...\OpenDSS\Doc\

Outline – Introduction to OpenDSS

- What is OpenDSS?
  - Distribution network models
- Why was OpenDSS developed?
- What are the key features?
  - Built-in solution modes
  - Controls
  - Overall model concept
  - Models implemented
  - Input data requirements
  - Advanced types of data
- What can OpenDSS be used for?
- User interfaces
- Additional Information

What is OpenDSS?

- **Script-driven**, frequency-domain electrical circuit simulation tool
  - Limited graphical user interface
  - Extremely flexible (can be driven from Excel VBA, MATLAB, etc.)

- Specific **models** for:
  - Supporting utility distribution networks analysis
  - Unbalanced, multi-phase power distribution networks
    - North American and European-style networks

What is OpenDSS?

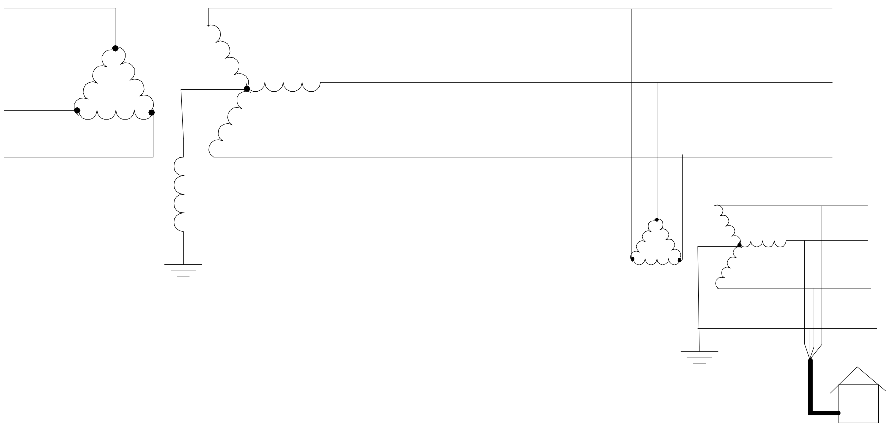
- **Heritage**
  - Harmonics solvers rather than power flow
    - Gives OpenDSS extraordinary distribution system modelling capability
  - Simpler to solve power flow problem with a harmonics solver than vice-versa
- Supports **all rms steady-state** (i.e., frequency domain) analyses commonly performed for utility distribution network planning
  - And many new types of analyses
  - Original purpose: distributed generation (DG) interconnection analysis

What is OpenDSS?

- **What it Isn't**
  - An *Electromagnetic* transients solver (Time Domain)
    - It can solve Electromechanical transients
  - A Power Flow program
    - <u>It is much more than that</u>
  - A radial circuit solver
    - Does meshed networks just as easily
  - A distribution data management tool
    - It is a simulation engine designed to work with data extracted from one or more utility databases

Distribution Network Models

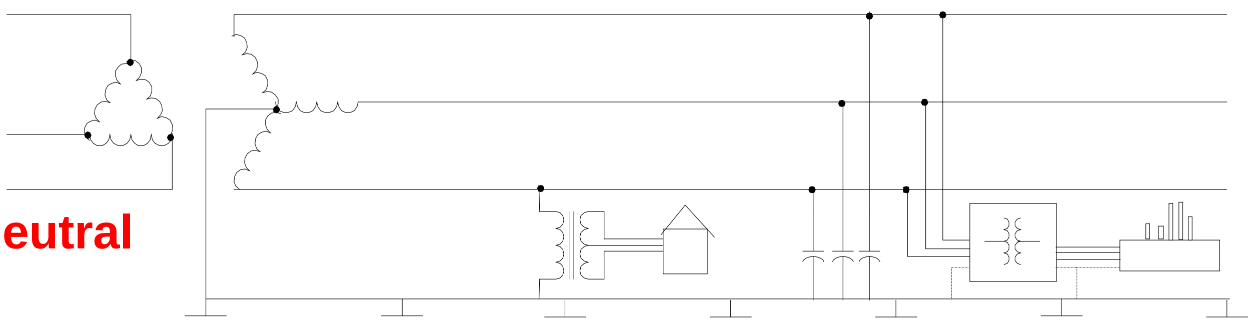
- European and North American distribution networks



3-wire unigrounded substations

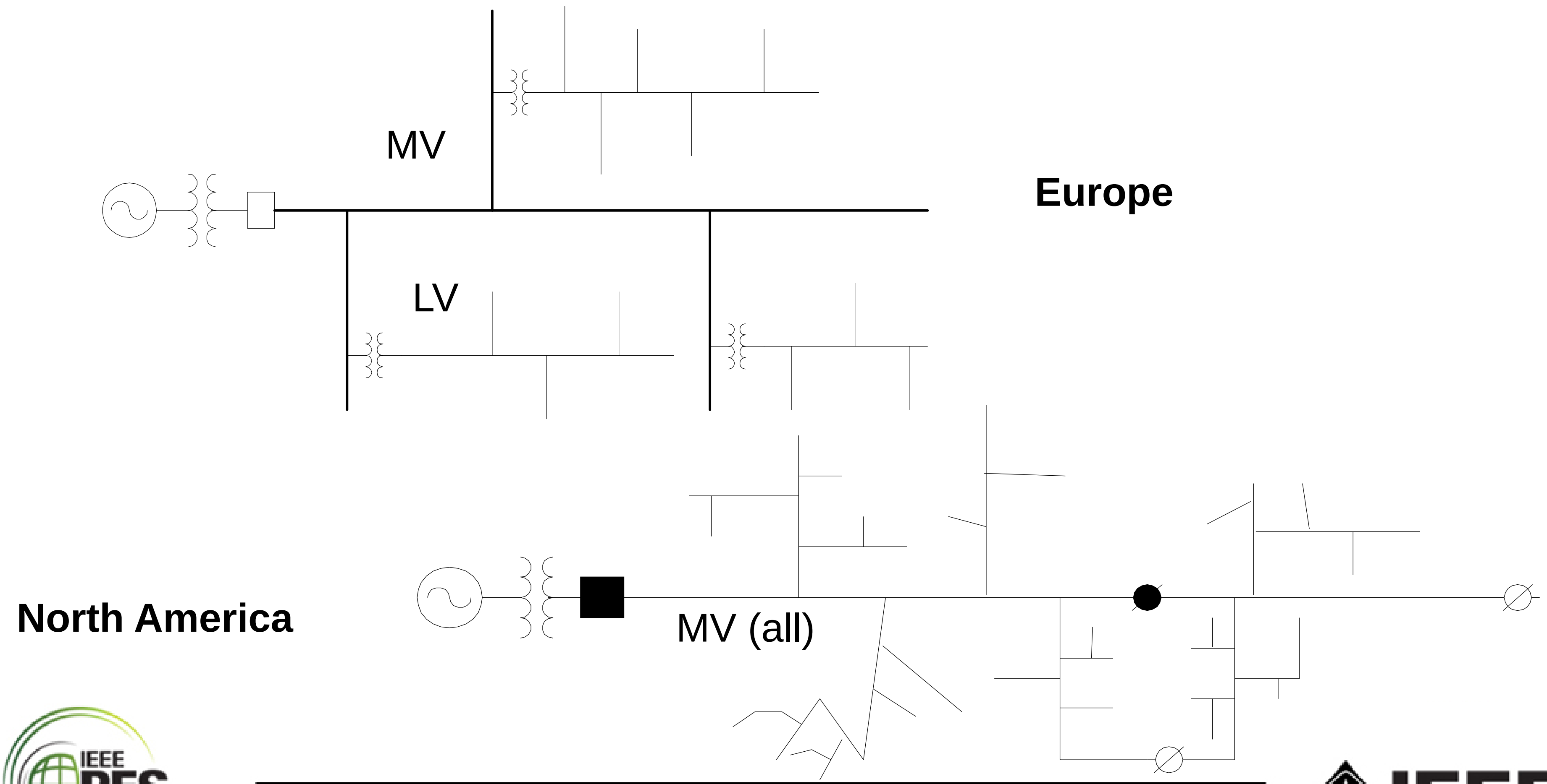
Three-phase throughout, including LV

4-wire multi-grounded neutral



Distribution Network Models

- European and North American distribution networks



Distribution Network Models

- **Europe**
  - MV system has simpler structure
  - LV system (400 V) is extensive
  - Hundreds of houses on MV/LV transformer
  - 230/400 V three-phase
  - Extended by adding wire
  - Fewer transformers

- **North America**
  - MV system is extensive, complex
  - LV is short
  - 4-5 houses per distribution transformer
  - 120/240 V single-phase ("split phase") service
  - 1 Industrial customer per distribution transformer
  - Or multiple transformers per customer
  - Extended by adding transformer + wire

Why was OpenDSS developed?

- **Initially**
  - Support all types of distribution planning for DG
  - High-level area economics to engineering interconnection issues
- Modelling the **Time Dimension**
  - Have to capture time value to get the right answer to DG and other smart grid issues
  - Key feature of OpenDSS
  - Enables study of volt-var optimisation and delivery efficiency
  - Sequential power flow from 1 s to 1 hr
  - Dynamics ~1 ms

What are the Key Features?

- See Documentation for more details
  - Main page in Wiki
    sourceforge.net/apps/mediawiki/electricdss/index.php?title=Main_Page
  - Within the installation files
    http://sourceforge.net/projects/electricdss/

- Designed to **allow expansion** indefinitely
  - Impossible to anticipate everything users will want to do
  - COM interface allows easier customisation
    - MS Excel VBA, Matlab, etc.

Built-in Solution Modes

- **Snapshot (static) Power Flow**
- Direct (non-iterative)
- **Daily mode (default: 24 1-hr increments)**
- **Yearly mode (default 8760 1-hr increments)**
- **Duty cycle** (1 to 5s increments)
- Dynamics (electromechanical transients)
- Fault study
- Monte Carlo fault study
- Harmonic
- Custom user-defined solutions

Controls

- A key feature is that **controls are modelled separately** from the devices being controlled
  - Capacitors
  - Regulators/tap changers
- **Control Modes**
  - Static
    - Power flows with large time steps
  - Time
    - Control queue employed to delay actions
    - Control acts when time is reached
  - Event

(Some) Models Implemented

- **Power Delivery** Elements
  - **Line** (All types of lines, cables)
  - **Transformer** (multi-phase, multi-winding transformer models)
  - **Capacitor, reactor** (Series and shunt)
- **Power Conversion** Elements
  - **Generator** (General generator models)
  - **Load** (General load models)
  - **PVsystem** (Solar PV system with panel and inverter)
  - **Storage** (Generic storage element models)
- **Meters**
  - **EnergyMeter** (Captures energy quantities and losses)
  - **Monitor** (Captures selected quantities at a point in the circuit)
  - **Sensor** (Simple monitor used for state estimation)

What can OpenDSS be used for?

- **Simple** power flow (unbalanced, n-phase)
- **Daily** loading simulations
- **Yearly** loading simulations
- **Duty** cycle simulations
  - Impulse loads (e.g., rock or car crushers)
  - Renewable generation
- **DG**
  - Interconnection studies/screening
  - Impacts of wind/solar PV
    - high penetration, variability, voltage rise, etc.)
  - Harmonic distortion
  - Dynamics/islanding

What can OpenDSS be used for?

- Hybrid simulation of communications and power networks
- Power delivery loss evaluations (EPRI Green circuits program → 80 feeders)
- Voltage optimization
- PEV/PHEV impact simulations
- Community energy storage (EPRI Smart Grid Demo)
- High-frequency harmonic/inter-harmonic interference
- Various unusual transformer configurations
- Transformer frequency response analysis
- Distribution automation control algorithm assessment
- Impact of tankless electric water heaters
- Wind farm collector simulations
- Wind farm interaction with transmission
- Wind generation impact on capacitor switching and regulator/LTC tap changer operations
- Protection system simulation
- Open-conductor fault conditions
- Circulating currents on transmission skywires
- Ground voltage rise during faults on lines
- Stray voltage simulations
- Industrial load harmonics studies/filter design
- Distribution feeder harmonics analysis, triplen harmonic filter design

And Many More

User Interfaces

- A **stand-alone executable** program that provides a text-based interface (multiple windows)
  - Some graphical output is also provided.
  - No graphical input is provided.

- An in-process **COM server** (for MS Windows) that supports driving the simulator from user-written programs.
  - An out-of-process COM server is under development to support execution from 64-bit programs.

Key Remarks – Introduction to OpenDSS

- Extremely **flexible** software package, but **limited interface**
- Provides **models** for utility distribution networks analysis
  - Unbalanced, multi-phase (USA and EU-style networks)
- Supports **all rms steady-state** analyses
- Models the **time dimension**
- **Controls are modelled separately** from the devices
  - Provides a lot of flexibility
- **Some models** are already implemented
- Can be used for **many type of studies**

Additional Information: Useful Links (Download and Info)

- OpenDSS Download
  - http://sourceforge.net/projects/electricdss/
- OpenDSS Wiki
  - http://sourceforge.net/apps/mediawiki/electricdss/index.php?title=Main_Page
- Roger Dugan's links for OpenDSS
  - http://www.rogerdugan.com/OpenDSS
- OpenDSS Presentations by EPRI (FTP)
  - ftp://opendss_2010:OpenDSS_ftp_2010@ftp.epri.com/
- OpenDSS Forum
  - http://sourceforge.net/projects/electricdss/forums/forum/861976

Additional Information: Basic OpenDSS Files

- OpenDSS.EXE — Standalone EXE
- **OpenDSSEngine.DLL — In-process COM server**
- KLUSolve.DLL — Sparse matrix solver
- DSSgraph.DLL — DSS graphics output

- Copy these files to the directory (folder) of your choice
  - Typically c:\OpenDSS or c:\Program Files\OpenDSS

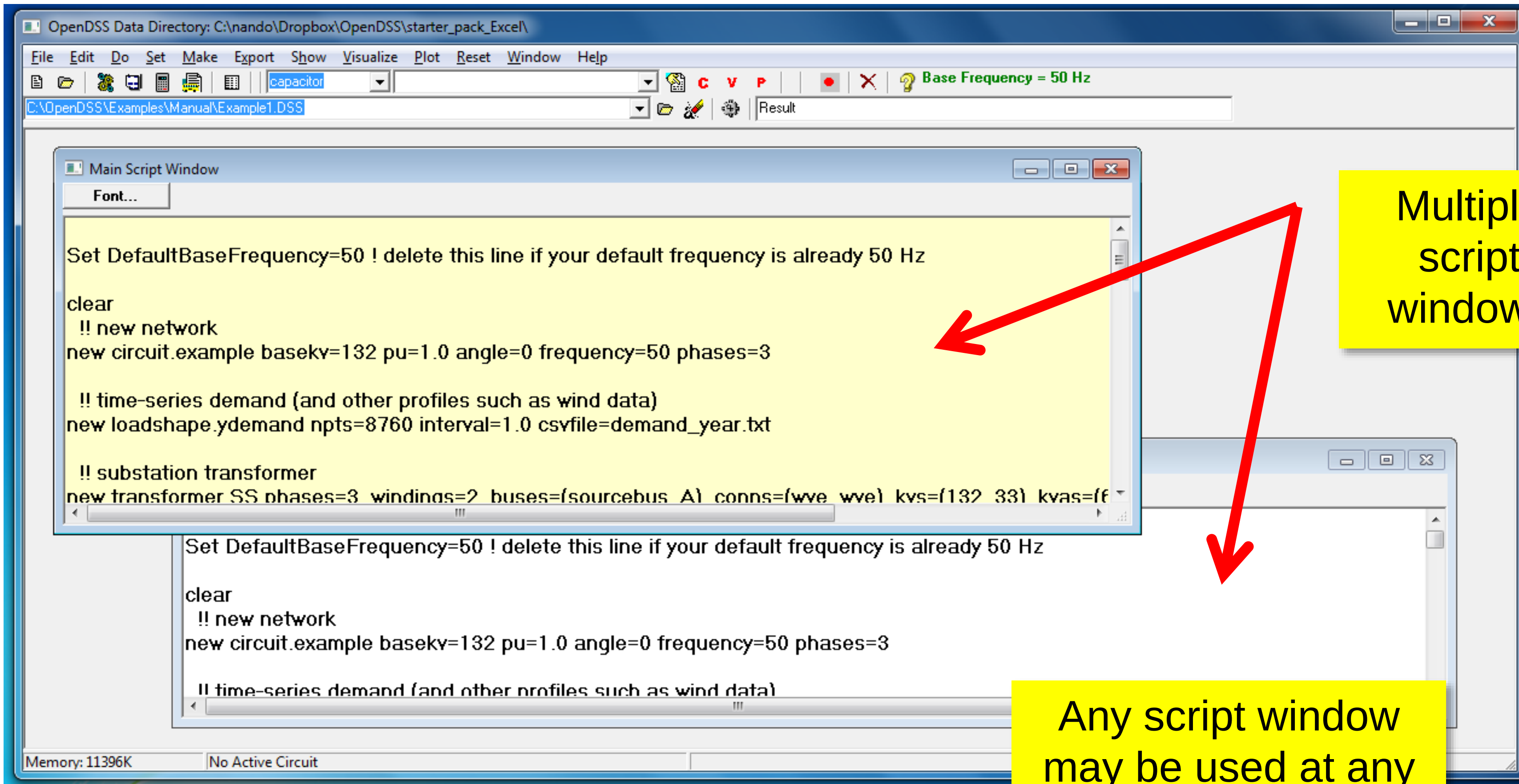
- The current OpenDSS installer automatically registers the COM server (OpenDSSEngine.DLL)
  - However, not the ParserX!

Additional Information: Accessing the COM Server

Examples of accessing the COM server in various languages

- In MATLAB:
  - DSSobj = actxserver('OpenDSSEngine.DSS');
- In VBA:
  - Public DSSobj As OpenDSSEngine.DSS
    Set DSSobj = New OpenDSSEngine.DSS
- In Dephi
  - {Import Type Library}
  - DSSObj := coDSS.Create;
- In PYTHON:
  - self.engine = win32com.client.Dispatch("OpenDSSEngine.DSS")

Additional Information: OpenDSS Standalone EXE User Interface



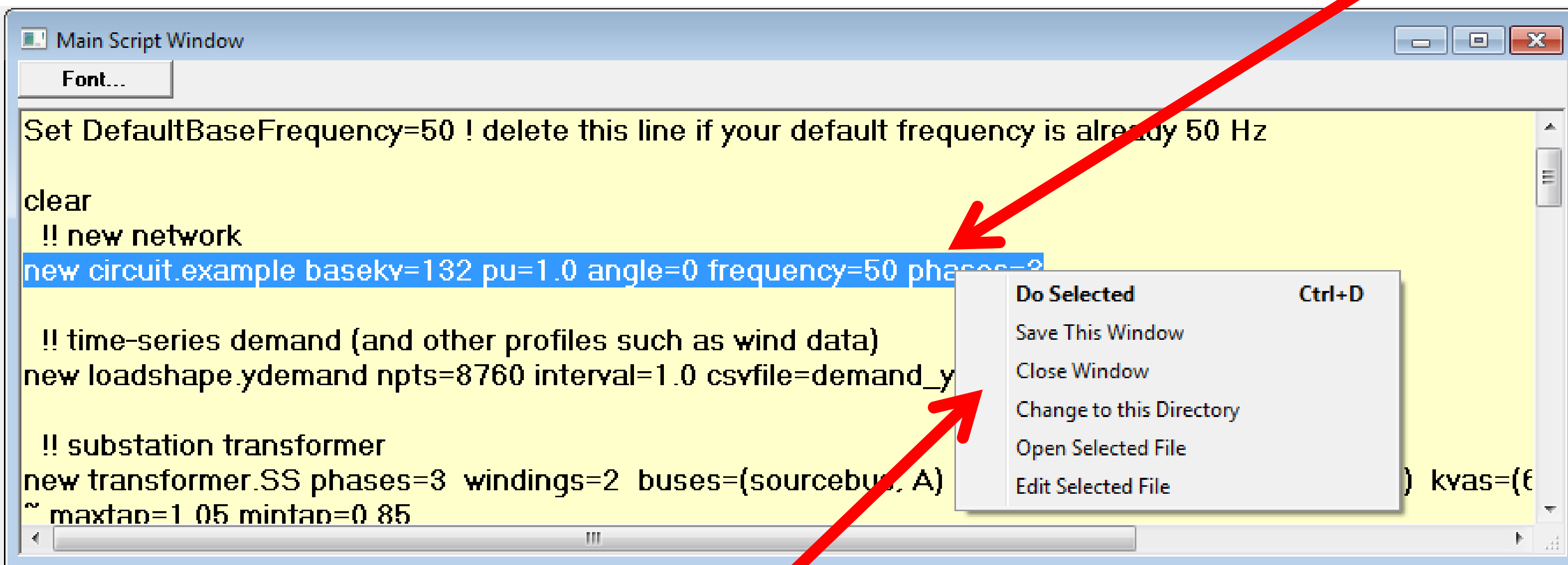
Multiple script windows

Any script window may be used at any time

Additional Information: Executing Scripts in the Standalone EXE

- DSS executes selected line or opens selected file name

Select all or part of a line



Right-click to get this pop-up menu

14:40 to 15:20

BASIC MODELLING OF MV AND LV NETWORKS

Outline – Basic Modelling

- Creation of a simple MV test network
  - Execution of an snapshot power flow
  - Graphical interface of OpenDSS to retrieve results
  - Export monitors and energy meters
  - Model an automatic voltage control (AVC) system for the transformer
  - Loadshapes and hourly time-series simulations
  - Model a three-phase medium scale distributed generator
  - Special commands
- Creation of simple LV test network
  - Loadshapes and run minute by minute time-series simulations
  - Model small-scale distributed generators
  - Special commands

Before you start

- First, use the right frequency
  - Always set the default base frequency to the one you normally use. Otherwise, if a circuit is specified with a different one, it will not run.

```
Set DefaultBaseFrequency=50
```

- Don't forget the "**sourcebus**"
  - This might be obvious, but do not forget to define the "sourcebus" (i.e., the slack/reference bus) when appropriate (through a transformer or a line).

- Remember: OpenDSS mimics real-life networks
  - This means, you need to **place monitors/meters**

Creation of a Simple MV Network: Basic Steps

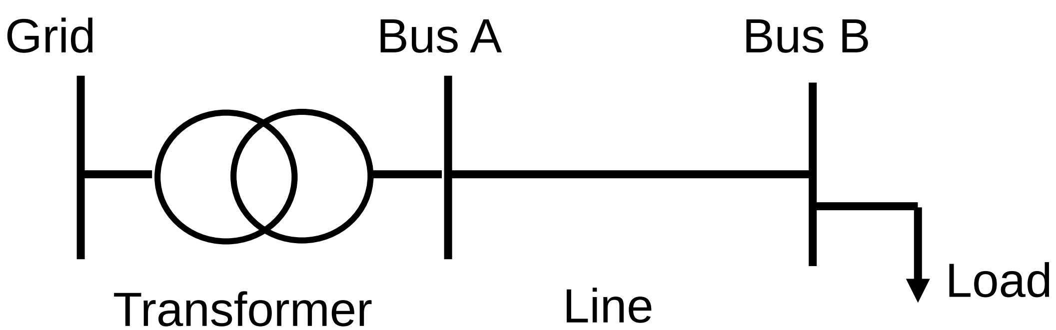
- “Clear” command
- Set your own datapath (to read/write files – be organised!)

```
set datapath=C:\OpenDSS-Training-Material\
```

- New circuit

```
New circuit.example basekV=132 pu=1.0 angle=0 frequency=50 phases=3
```

… if the slack bus (“Grid” in the figure and **sourcebus** for OpenDSS) has a voltage different from 1.0pu, then change it accordingly (e.g., pu=1.01)



Creation of a Simple MV Network: Basic Steps

- Substation transformer

```
new transformer.SS phases=3  windings=2  buses=(sourcebus, A)  conns=(wye, wye)
kvs=(132, 33)  kvas=(30000, 30000) %loadloss=0 xhl=12.5
```

- Lines
  - Produce “**linecodes**” (it is a more organised way)

```
new linecode.ABC nphases=3 R1=2.13444 X1=1.554003 units=km
```

  - Then create the corresponding **lines**

```
new line.lineA-B bus1=A bus2=B length=1 phases=3 units=km linecode=ABC
```

- Loads

```
new load.loadB bus1=B phases=3 kV=33 kW=5000 kvar=1640 model=1
```

Be careful with Vmaxpu and Vminpu

Creation of a Simple MV Network: Basic Steps

- Control modes

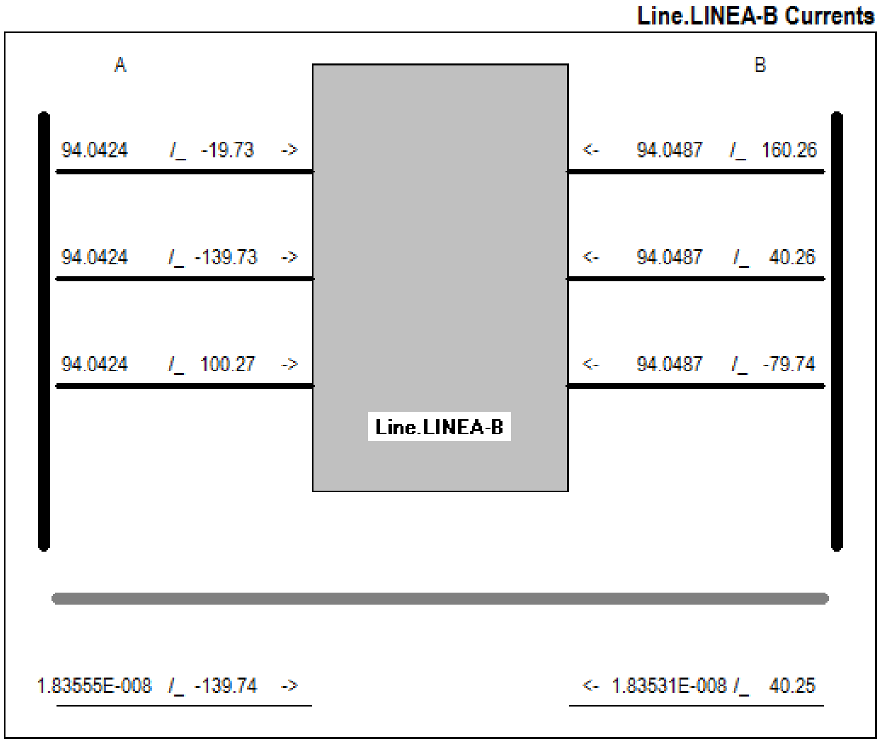
  set controlmode=static

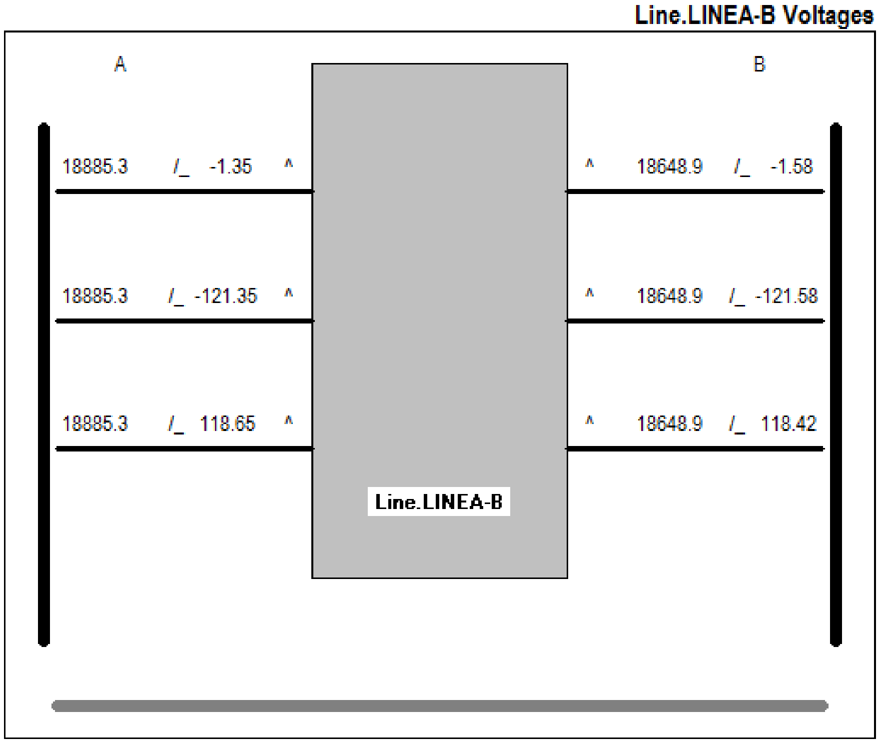
  – Ideal for snapshot and hourly time-series analyses
- Type of solution

  set mode=snapshot
- Solve
  – Write the "**solve**" command
  – **Select** all the corresponding text and **CTRL+D** (as before)
- Save this DSS file
  – Be organised and save the file in a folder where other files will also be exported to (CSV results) or imported from (loadshapes)

Creation of a Simple MV Network: Results

- Results
  - Select "**line**", then "**LineA-B**". Click on C, V or P





Creation of a Simple MV Network: Monitors/Meters

- Although you can always export CSV files with all voltages, currents, etc. it is advisable to **'deploy' monitoring** devices (particularly when doing time – series analyses and using the **COM server** as is more manageable).
  - These monitoring devices are divided into two categories: energy meters and monitors.
- It is sensible to have one at the **sourcebus** (or relevant substation) to meter net imports/exports (in the presence of DG) and total losses.
  - **Meters can only be used in power delivery elements** (e.g., lines, transformers). Either one needs to use monitors or create a line connecting the load and actual connection point so the meter can be 'attached' to one of the ends.
  - If fictitious lines need to be created, use **very low values of R and X=0**.

Creation of a Simple MV Network: Monitors/Meters

- Energy Meter/Monitor at **Terminal 1** will consider **load-led current as positive**, given that the model assumes currents flowing to the 'right'.
- Energy Meter/Monitor at **Terminal 2** will consider **load-led current as negative**, given that the model assumes currents flowing to the 'left'.

```
|                                   |
|              _________            |
|    >>>    |           |    <<<    |
|-----------|           |-----------|
|           |           |           |
|    >>>    |  network  |    <<<    |
|-----------|  element  |-----------|----> Load
|           |           |           |
|    >>>    |           |    <<<    |
|-----------|___________|-----------|
|                                   |
|                                   |
Terminal 1                          Terminal 2
```

Creation of a Simple MV Network: Monitors/Meters

- To see what happens from a time-series perspective
  - Code placed **before solving** the circuit
- P, Q

```
new monitor.SS element=transformer.SS terminal=1 mode=1 ppolar=no
```

- V, I

```
new monitor.Vbb element=transformer.SS terminal=2 mode=0
```

- Tap position

```
new monitor.Tap element=transformer.SS terminal=1 mode=2
```

… the terminal has to be consistent with the location of the taps

- Meters (to check energy exports/imports, losses)

```
new energymeter.Grid element=transformer.SS terminal=1
```

```
new energymeter.mloadB element=line.lineA-B terminal=2
```

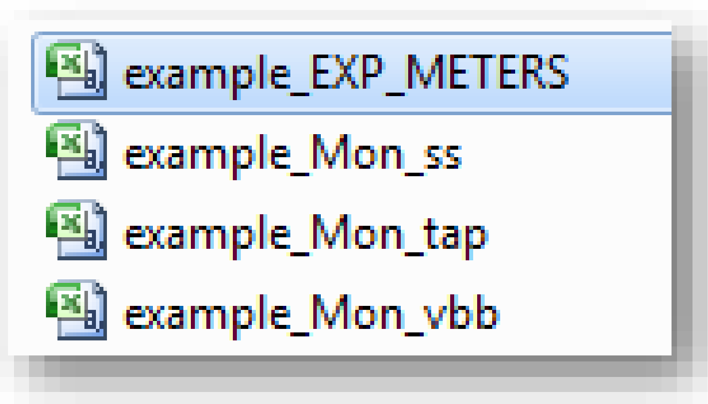
Creation of a Simple MV Network: Monitors/Meters

- Exporting data **after** the "**solve**" command
- Given that this is an 'snapshot' we need to 'force' the monitors and energy meters to record data

```
monitor.SS.action=take
monitor.Vbb.action=take
monitor.Tap.action=take
energymeter.Grid.action=take
energymeter.mloadB.action=take
```

- Then, to export the data

```
export monitors SS
export monitors Vbb
export monitors Tap
export meters
```



Creation of a Simple MV Network: Monitors/Meters

- Monitor SS

P1 (kW)	Q1 (kvar)	P2 (kW)	Q2 (kvar)	P3 (kW)	Q3 (kvar)
1685.42	600.131	1685.42	600.131	1685.42	600.131

- Monitor Vbb

V1	VAngle1	V2	VAngle2	V3	VAngle3
18885.3	-1.34791	18885.3	-121.348	18885.3	118.652

- Energy Monitors

Meter	"kWh"	"kvarh"	"Max kW"	"Max kVA"
"GRID"	5056	1800	5056	5367
"MLOADE	-5000	-1640	-5000	5262

Creation of a Simple MV Network: AVC

- Automatic Voltage Control (AVC) System

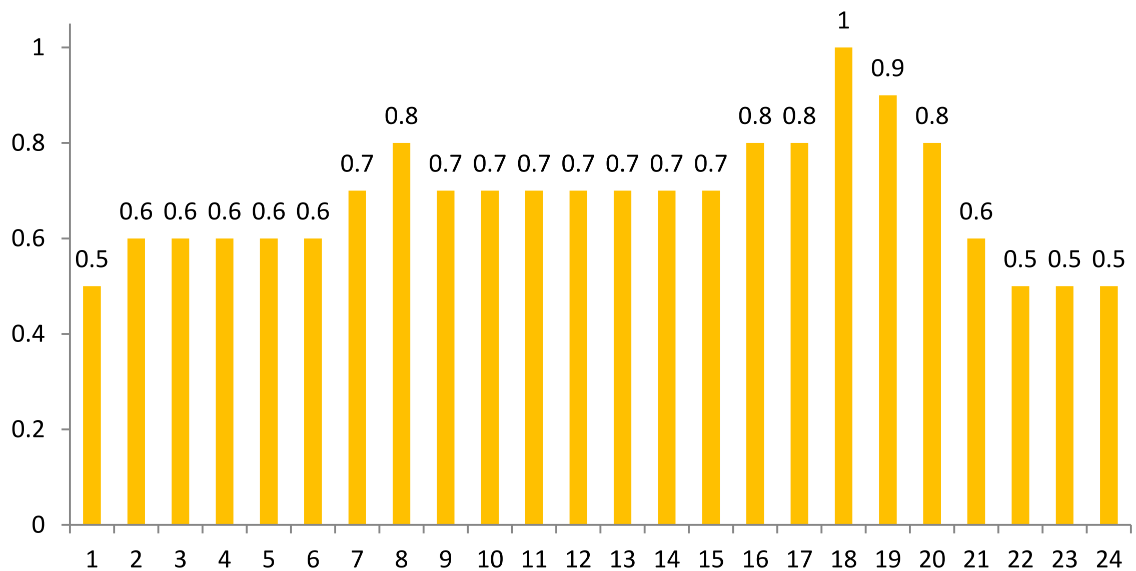
```
(add to the transformer) ~ maxtap=1.05 mintap=0.85
new regcontrol.SS transformer=SS winding=2 tapwinding=1 vreg=(100) ptratio=(33000 3 sqrt / 100 /) band=1 !this band will lead to a final voltage within 0.995-1.005pu
```

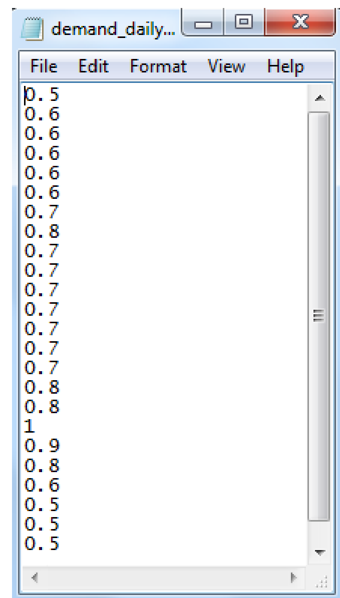
- Note
  - *winding* specifies the controlled winding. *tapwinding* specifies the location of the taps. It has to be written after *winding*
  - vreg*ptratio gives the actual target voltage at the winding (line-to-neutral for Wye connection)
  - 100V is a suggested vreg as the band becomes more intuitive (e.g., for 2V of band, the final voltage will be within 0.99-1.01pu). If the target is 1.03pu, then vreg=103 (1.02-1.04pu for 2V band)
  - (33000 3 sqrt / 100 /) is RPN for (33000/sqrt(3))/100 = 190.5256

RPN: Reverse Polish Notation

Creation of a Simple MV Network: Loadshapes

- Create a txt file “demand_daily.txt”
  - Normalised values (0 to 1)
  - One value per line
  - Save this file in the **same folder** as the DSS file





1	0.5
2	0.6
3	0.6
4	0.6
5	0.6
6	0.6
7	0.7
8	0.8
9	0.7
10	0.7
11	0.7
12	0.7
13	0.7
14	0.7
15	0.7
16	0.8
17	0.8
18	1
19	0.9
20	0.8
21	0.6
22	0.5
23	0.5
24	0.5

IEEE PES Power & Energy Society®

Creation of a Simple MV Network: Loadshapes

- Create the corresponding loadshape. Let's call it "daily"

```
new loadshape.daily npts=24 interval=1.0 csvfile=demand_daily.txt
```

Before "solve", after "new circuit"

- The load now has to be 'linked' to the loadshape we just created

```
(add to the load) ~ daily=daily status=variable
```

- Change the mode to daily

```
set mode=daily
```

- Comment/delete the code to force monitors/energy meters to record data (only for 'snapshot')

```
! monitor.SS.action=take!
energymeter.Grid.action=take
```

Creation of a Simple MV Network: Monitors/Meters

- **Monitor SS**

hour	t(sec)	P1 (kW)	Q1 (kvar)	P2 (kW)	Q2 (kvar)	P3 (kW)	Q3 (kvar)
1	0	837.943	286.135	837.943	286.135	837.943	286.135
2	0	1006.6	346.44	1006.6	346.44	1006.6	346.44
3	0	1006.59	346.438	1006.59	346.438	1006.59	346.438
4	0	1006.59	346.438	1006.59	346.438	1006.59	346.438
5	0	1006.59	346.438	1006.59	346.438	1006.59	346.438
6	0	1006.59	346.438	1006.59	346.438	1006.59	346.438
7	0	1175.54	407.947	1175.54	407.947	1175.54	407.947
8	0	1344.98	470.607	1344.98	470.607	1344.98	470.607
9	0	1175.82	408.074	1175.82	408.074	1175.82	408.074
10	0	1175.68	408.01	1175.68	408.01	1175.68	408.01
11	0	1175.68	408.01	1175.68	408.01	1175.68	408.01
12	0	1175.68	408.01	1175.68	408.01	1175.68	408.01
13	0	1175.68	408.01	1175.68	408.01	1175.68	408.01
14	0	1175.68	408.01	1175.68	408.01	1175.68	408.01
15	0	1175.68	408.01	1175.68	408.01	1175.68	408.01
16	0	1344.98	470.607	1344.98	470.607	1344.98	470.607
17	0	1345.16	470.697	1345.16	470.697	1345.16	470.697
18	0	1685.28	599.465	1685.28	599.465	1685.28	599.465
19	0	1515.24	534.636	1515.24	534.636	1515.24	534.636
20	0	1345.33	470.789	1345.33	470.789	1345.33	470.789
21	0	1006.6	346.44	1006.6	346.44	1006.6	346.44
22	0	837.971	285.982	837.971	285.982	837.971	285.982
23	0	837.892	285.965	837.892	285.965	837.892	285.965
24	0	837.892	285.965	837.892	285.965	837.892	285.965

Creation of a Simple MV Network: Monitors/Meters

- **Monitor Vbb**

hour	t(sec)	V1	VAngle1	V2	VAngle2	V3	VAngle3
1	0	18971.7	-0.66709	18971.7	-120.667	18971.7	119.333
2	0	19075.1	-0.79299	19075.1	-120.793	19075.1	119.207
3	0	19075.1	-0.79298	19075.1	-120.793	19075.1	119.207
4	0	19075.1	-0.79298	19075.1	-120.793	19075.1	119.207
5	0	19075.1	-0.79298	19075.1	-120.793	19075.1	119.207
6	0	19075.1	-0.79298	19075.1	-120.793	19075.1	119.207
7	0	19058.2	-0.9269	19058.2	-120.927	19058.2	119.073
8	0	19041	-1.06146	19041	-121.061	19041	118.939
9	0	19058.1	-0.92712	19058.1	-120.927	19058.1	119.073
10	0	19058.1	-0.92701	19058.1	-120.927	19058.1	119.073
11	0	19058.1	-0.92701	19058.1	-120.927	19058.1	119.073
12	0	19058.1	-0.92701	19058.1	-120.927	19058.1	119.073
13	0	19058.1	-0.92701	19058.1	-120.927	19058.1	119.073
14	0	19058.1	-0.92701	19058.1	-120.927	19058.1	119.073
15	0	19058.1	-0.92701	19058.1	-120.927	19058.1	119.073
16	0	19041	-1.06146	19041	-121.061	19041	118.939
17	0	19041	-1.0616	19041	-121.062	19041	118.938
18	0	19005.9	-1.33249	19005.9	-121.332	19005.9	118.668
19	0	19023.5	-1.19693	19023.5	-121.197	19023.5	118.803
20	0	19040.9	-1.06174	19040.9	-121.062	19040.9	118.938
21	0	19075.1	-0.79299	19075.1	-120.793	19075.1	119.207
22	0	19091.8	-0.65957	19091.8	-120.66	19091.8	119.34
23	0	19091.8	-0.65951	19091.8	-120.66	19091.8	119.34
24	0	19091.8	-0.65951	19091.8	-120.66	19091.8	119.34

Creation of a Simple MV Network: Monitors/Meters

- Monitor Tap
- Energy Meters
  - Note the different signs. In this case,
    - positive for terminal 1
    - negative for terminal 2

Meter	"kWh"	"kvarh"	"Max kW"	"Max kVA"
"GRID"	82133	28511	5056	5366
"MLOADB"	-81500	-26732	-2500	5262

hour	t(sec)	Tap (pu)
1	0	1
2	0	0.99375
3	0	0.99375
4	0	0.99375
5	0	0.99375
6	0	0.99375
7	0	0.99375
8	0	0.99375
9	0	0.99375
10	0	0.99375
11	0	0.99375
12	0	0.99375
13	0	0.99375
14	0	0.99375
15	0	0.99375
16	0	0.99375
17	0	0.99375
18	0	0.99375
19	0	0.99375
20	0	0.99375
21	0	0.99375
22	0	0.99375
23	0	0.99375
24	0	0.99375

Creation of a Simple MV Network: DG

- Modelling a generator

```
new generator.CHP bus1=B phases=3 kV=33 kW=5000 pf=0.95 model=1
```

  - This means, constant injection of 5MW

- Impact on network performance
  - In terms of energy, our test system is now a net ‘exporter’
  - Also, note the signs of the values below

Meter	"kWh"	"kvarh"	"Max kW"	"Max kVA"
"GRID"	-38347	-12302	0	2608
"MLOADB"	38500	12710	2500	2632

Special Commands 1/2

- Per unit voltages
  - set voltagebases=[132 33]
  - calcvoltagebases
  - This command is for internal purposes (not for CSV exports)

Before "solve", after "new circuit"

```
SYMMETRICAL COMPONENT VOLTAGES BY BUS (for 3-phase buses)

Bus          Mag:   V1 (kV)    p.u.     V2 (kV)   %V2/V1    V0 (kV)    %V0/V1

sourcebus     76.26      1.001  3.727E-008  4.887E-008 4.961E-009 6.505E-009
a             19.13      1.004  9.348E-009  4.887E-008 1.244E-009 6.505E-009
b             19.24       1.01  9.404E-009  4.887E-008 1.245E-009 6.471E-009
```

- "Show" command
  - show summary
  - show voltages
  - show currents
  - show deltaV

After "solve"

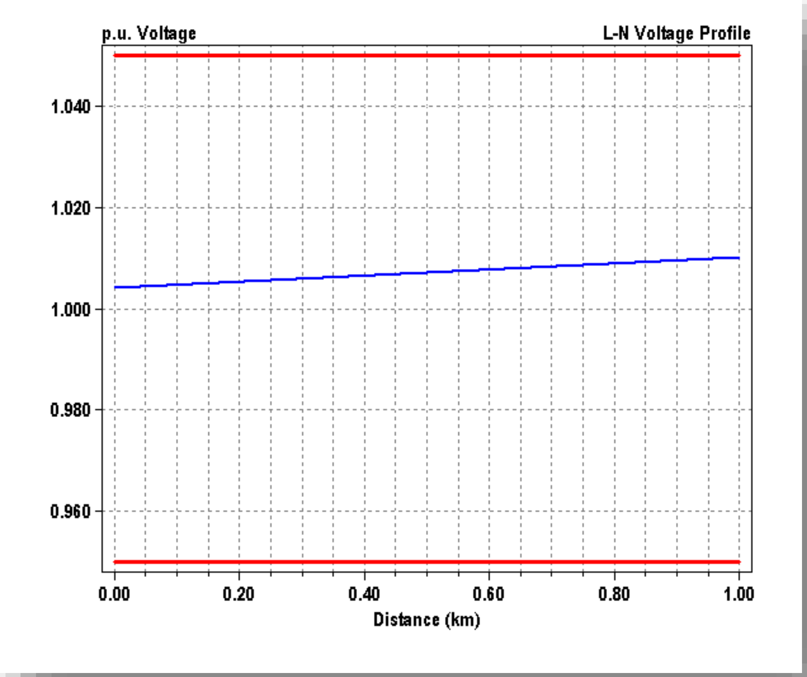
```
VOLTAGES ACROSS CIRCUIT ELEMENTS WITH 2 TERMINALS

Source Elements

Element,             Conductor,     Volts,    Percent,           kVBase,   Angle


Power Delivery Elements

Element,             Conductor,     Volts,    Percent,           kVBase,   Angle

Transformer.SS       ,     1,          57135,             0,        76.21,   -0.1
Transformer.SS       ,     2,          57135,             0,        76.21,  -120.1
Transformer.SS       ,     3,          57135,             0,        76.21,   119.9

Line.LINEA-B         ,     1,         120.38,       0.63185,       19.053,  -161.4
Line.LINEA-B         ,     2,         120.38,       0.63185,       19.053,    78.6
Line.LINEA-B         ,     3,         120.38,       0.63185,       19.053,   -41.4

= = = = = = = = = = = = = = = = = = =  = = = = = = = = = = =  = =

Power Conversion Elements

Element,             Conductor,     Volts,    Percent,           kVBase,   Angle
```

Special Commands 2/2

After "solve"

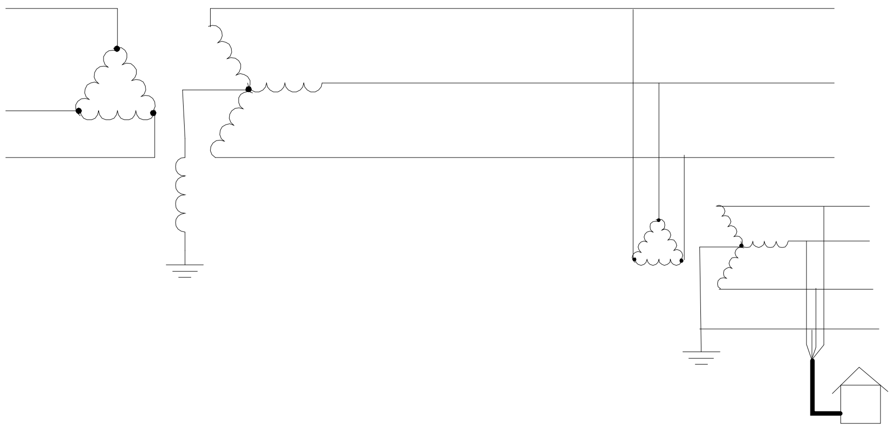
- "Visualize" command
  - visualize What={voltages} element=transformer.SS
  - visualize What={voltages} element=line.lineA-B
  - visualize What={currents} element=line.lineA-B
- "Plot" command
  - plot profile phases=all
  - Note that
    - Requires per unit calculations
    - Only one energy meter
      - Typically at the SS

After "solve"



Distribution LV Network Models

- European distribution networks



3-wire unigrounded substations
Three-phase throughout, including LV

- Particularly, for LV networks, the neutral has to be explicitly modelled. The (explicit) ground, depending on the connection philosophy, can be neglected.
  - For OpenDSS this means connections with 4 phases (3ph+n)

Test LV Network: Snapshot

- New circuit and transformer

```
Clear

set datapath=C:\OpenDSS-Training-Material\

new circuit.LV basekv=11 pu=1.0 angle=0 frequency=50 phases=3

new transformer.LVSS windings=2  buses=(Sourcebus, busbar.1.2.3.4)  conns=(delta, wye)
kvs=(11, 0.415)  kvas=(250, 250) %loadloss=0 xhl=2.5
```

Test LV Network: Snapshot

- New lines

```
new linecode.95mm nphases=4 R1=0.322 X1=0.074 R0=1.282 X0=0.125 units=km
new linecode.35mm nphases=2 R1=0.868 X1=0.077 R0=0.910 X0=0.077 units=km

new line.busbar-A bus1=busbar.1.2.3.4 bus2=A.1.2.3.4 length=0.3 phases=4 units=km
linecode=95mm
new line.A-MPANone bus1=A.1.4 bus2=MPANone.1.4 length=0.01 phases=2 units=km
linecode=35mm
new line.A-MPANtwo bus1=A.2.4 bus2=MPANtwo.2.4 length=0.01 phases=2 units=km
linecode=35mm
new line.A-MPANthree bus1=A.3.4 bus2=MPANthree.3.4 length=0.01 phases=2 units=km
linecode=35mm
```

Test LV Network: Snapshot

- New loads

```
new load.MPANone bus1=MPANone.1.4 phases=1 kV=(0.4 3 sqrt /) kW=30 pf=0.95
model=1 conn=wye status=fixed
new load.MPANtwo bus1=MPANtwo.2.4 phases=1 kV=(0.4 3 sqrt /) kW=20 pf=0.95
model=1 conn=wye status=fixed
new load.MPANthree bus1=MPANthree.3.4 phases=1 kV=(0.4 3 sqrt /) kW=25 pf=0.95
model=1 conn=wye status=fixed
```

Be careful with Vmaxpu and Vminpu

- and monitors

```
new monitor.HVside element=transformer.LVSS terminal=1 mode=1 ppolar=no !active
and reactive power
new monitor.busbar element=transformer.LVSS terminal=2 mode=0 !voltages and
currents
new monitor.A_pq element=line.busbar-A terminal=2 mode=1 ppolar=no
```

Test LV Network: Snapshot

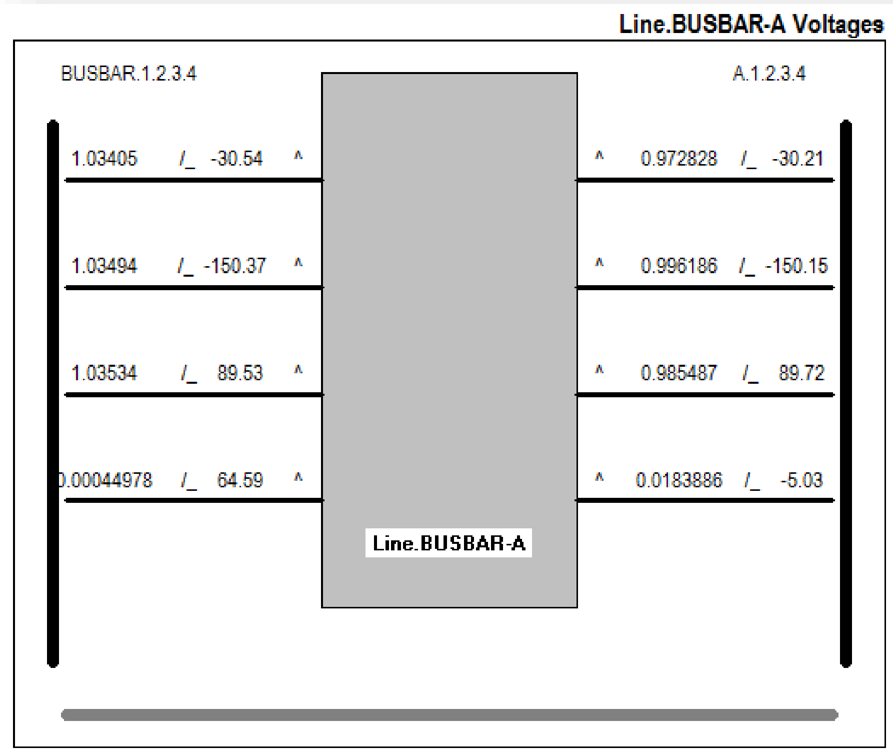
- New energy meter and last commands

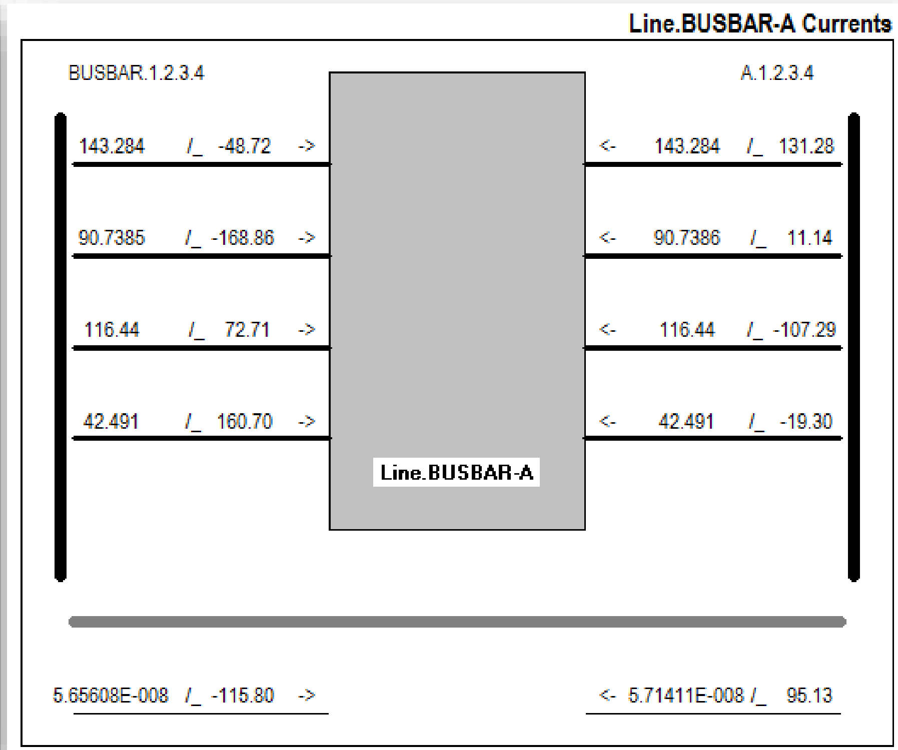
```
new energymeter.busbar element=line.busbar-A terminal=1

set controlmode=STATIC
set mode=snapshot

set voltagebases=[11 0.4]
calcvoltagebases
solve
```

  – ... Plus the typical ‘take’ for monitors and energy meters, exports, visualise (if needed), and plot of profiles

Test LV Network: Snapshot

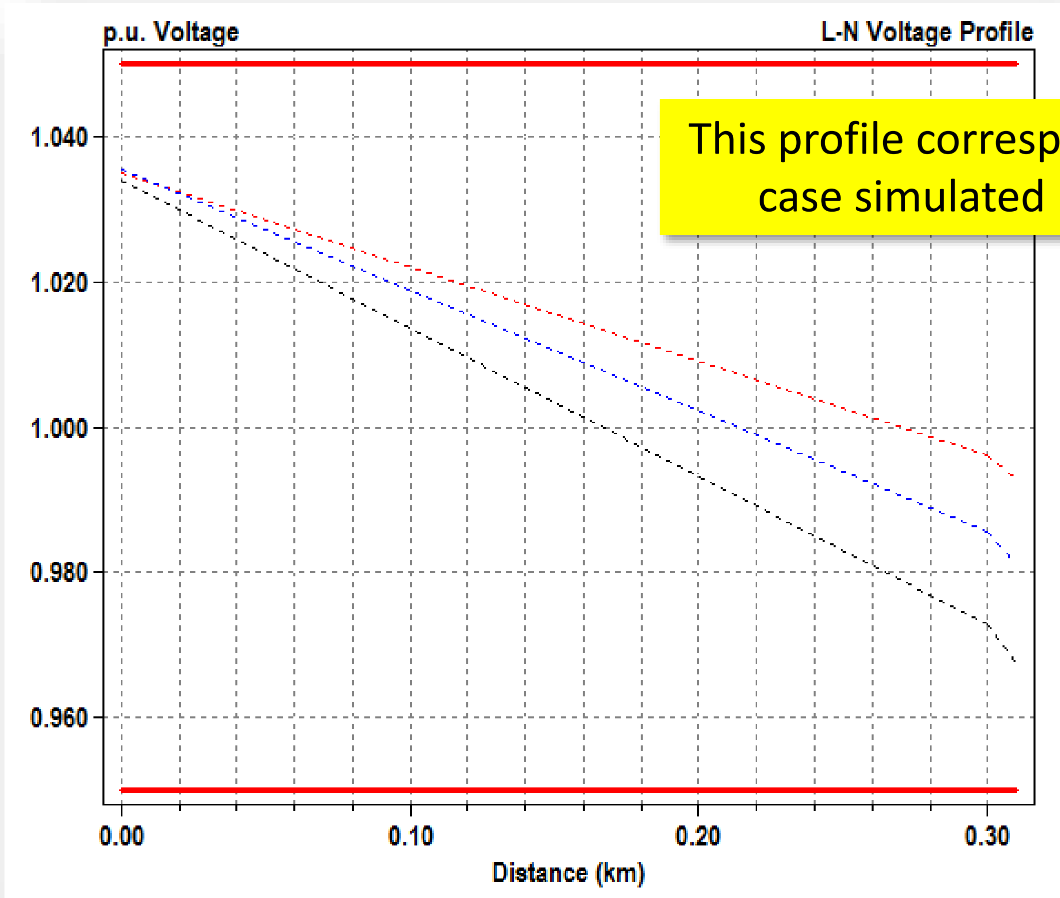
- Busbar-A: voltages (line-to-ground) and currents





Test LV Network: Snapshot

- Voltage profiles (line-to-ground)



This profile corresponds to the last case simulated by OpenDSS

Caution: Three Phases and Neutral

- The neutral has been explicitly modelled (".4" in the load code).
  - It is possible to place monitors (e.g., for the neutral currents).
  - This, however, creates another problem: voltages monitored will be line-to-ground rather than line-to-neutral.
  - So, unless it is necessary, avoid explicitly defining the neutral.
- Note that not defining the neutral does not mean that OpenDSS does not consider it.
  - If positive-sequence impedances (derived from four-wire cables) are provided, OpenDSS then automatically creates the 4x4 matrices and computes the effects of the neutral.
  - The phase voltage results will be line-to-neutral voltages (which is what we want).

Test LV Network: Loadshapes

- New loadshapes

```
new loadshape.demand npts=1440 minterval=1.0 csvfile=house_min_daily.txt action=normalize
```

  - Use normalisation when needed

After “new circuit” and before the “new load”

- Adding loadshapes to loads

```
new load... status=variable daily=demand
```

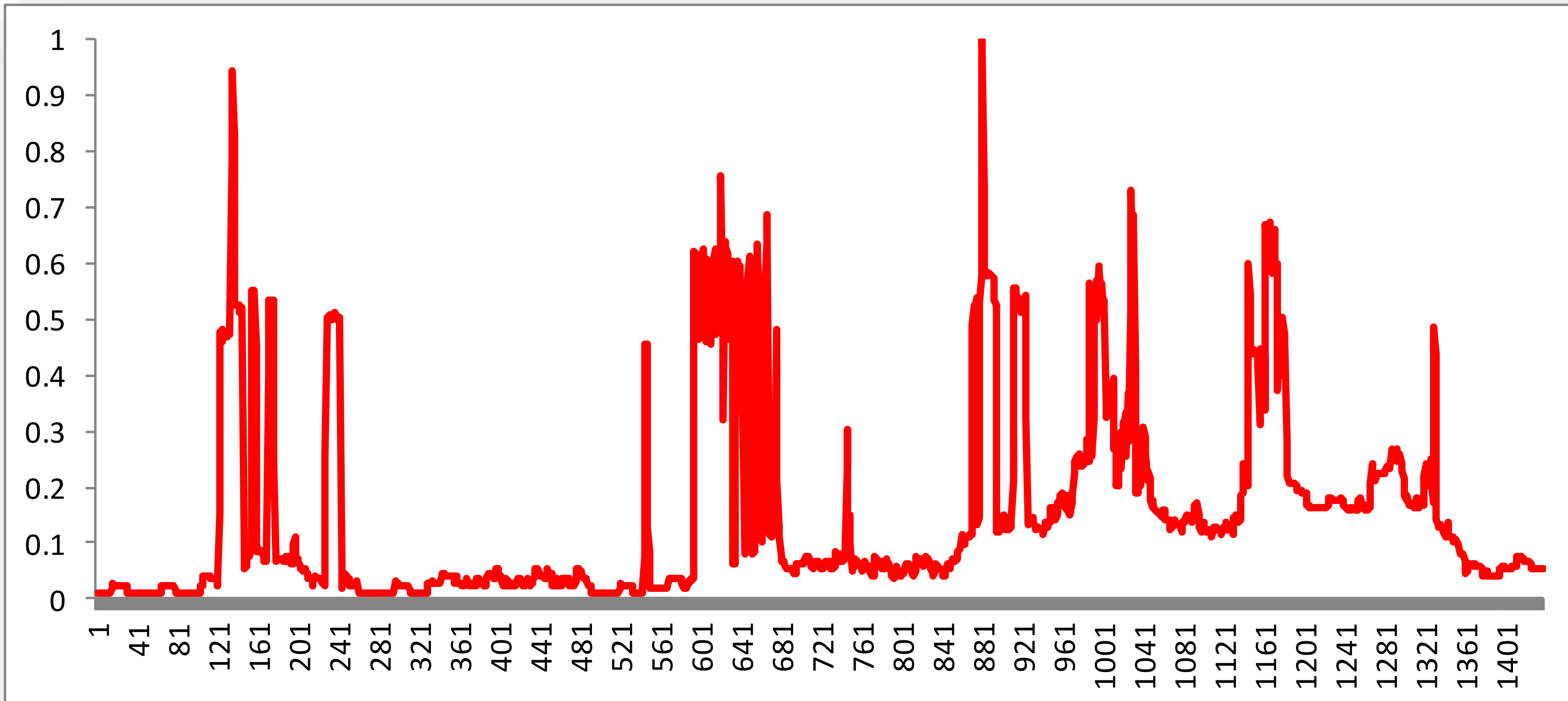
- Set mode for time-series

```
set mode=daily stepsize=1m number=1440
```

- Comment/delete the code to force monitors/energy meters to record data (only for ‘snapshot’)

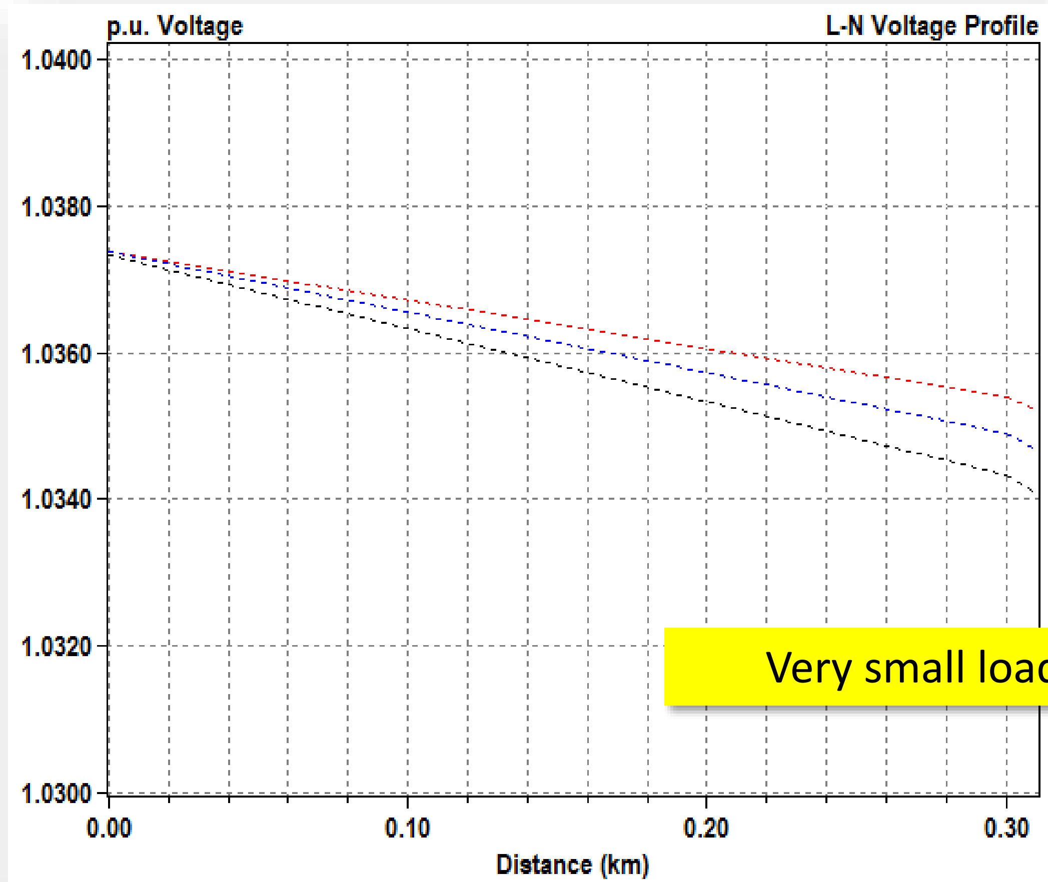
Test LV Network: Loadshapes

- house_min_daily.txt (normalised profile)



Test LV Network: Loadshapes

- Voltage profiles (line-to-ground given the explicit neutral)



Very small load at midnight

Test LV Network: PV Generation

- Simple generator model (not the specific on for PV)
- New loadshapes

```
new loadshape.PVshape npts=1440 minterval=1 csvfile=pv_min_daily.txt
```

- Modelling the PV system

```
new generator.PVone bus1=MPANone.1.4 phases=1 kV=(0.4 3 sqrt /) kW=30 pf=1
model=1 status=variable daily=Pvshape
```

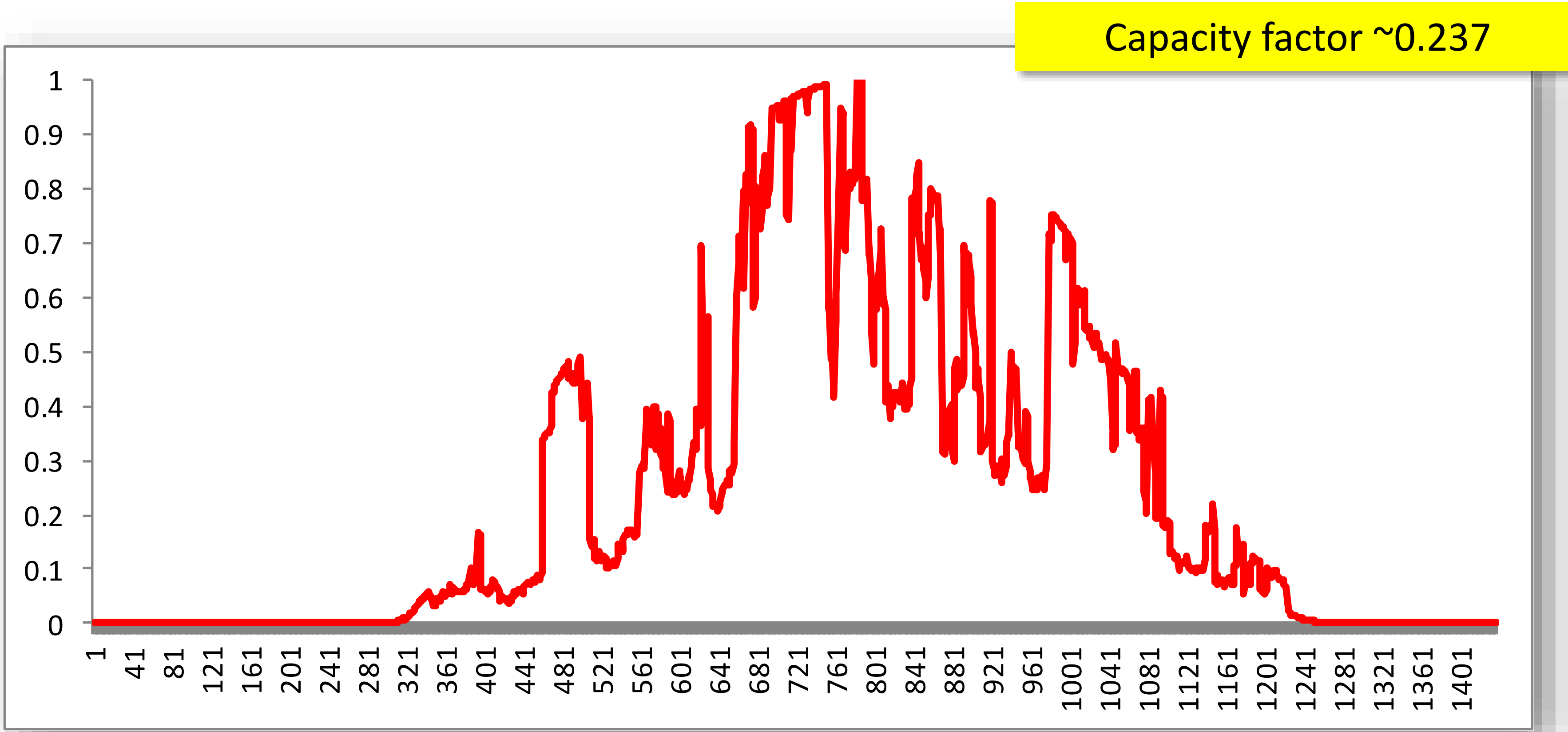
- Set mode for time-series

```
set mode=daily stepsize=1m number=1440
```

PV generation can also be modelled using the new objects in OpenDSS – Homework ☺

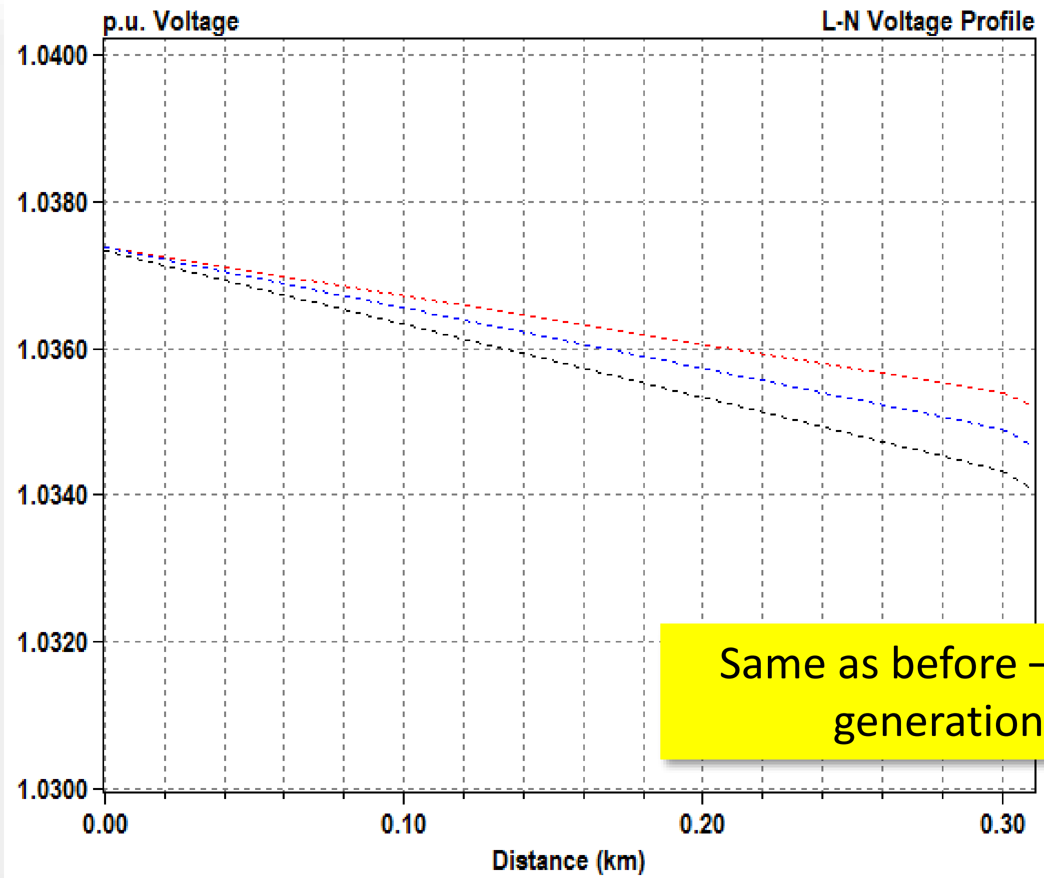
Test LV Network: PV Generation

- pv_min_daily.txt (normalised profile)



Test LV Network: PV Generation

- Voltage profiles (line-to-ground given the explicit neutral)



Same as before – there is no PV generation at night!

Test LV Network: PV Generation

- Total energy consumption (including losses) without PV
  - 280 kWh
- Expected generation from the PV system
  - Capacity factor x Nominal capacity x Number of hours
  - 0.237 x 30 kW x 24 h = ~171 kWh
- Expected total energy consumption (including losses) with PV
  - 280 kWh – 171 kWh = ~ 110 kWh

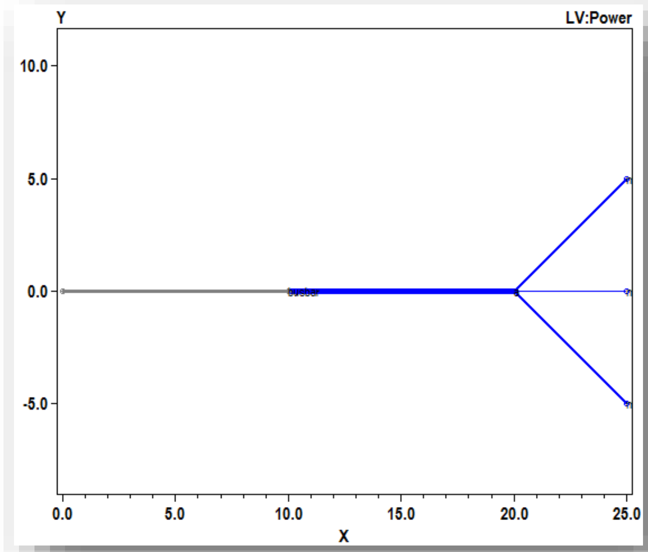
Key Remarks – Basic Modelling

- MV Networks
  - Always set the default base **frequency**
  - **Be organised** – data will end in the same folder
  - Deploy **monitoring** devices (<u>before solving</u>) when possible
    - Advise for large networks
  - **Loadshapes** are defined after circuit but before solve
- LV Networks
  - The **neutral** has to be explicitly modelled
  - Not defining the neutral does not mean that OpenDSS does not consider it.
    - **The phase voltage results will be line-to-neutral voltages**

Explore 1/3

- Plotting the circuit topology
  - Requires a CSV or txt file with coordinates
  - Thickness linked to currents/voltages/power/etc.
- Code
  - BusCoords Simple_LV_network_topology.txt
  - plot circuit power dots=y labels=y C1=blue
  - You will need to create the txt file
    - Busname, X, Y

After "solve"



Explore 2/3

Before "solve"

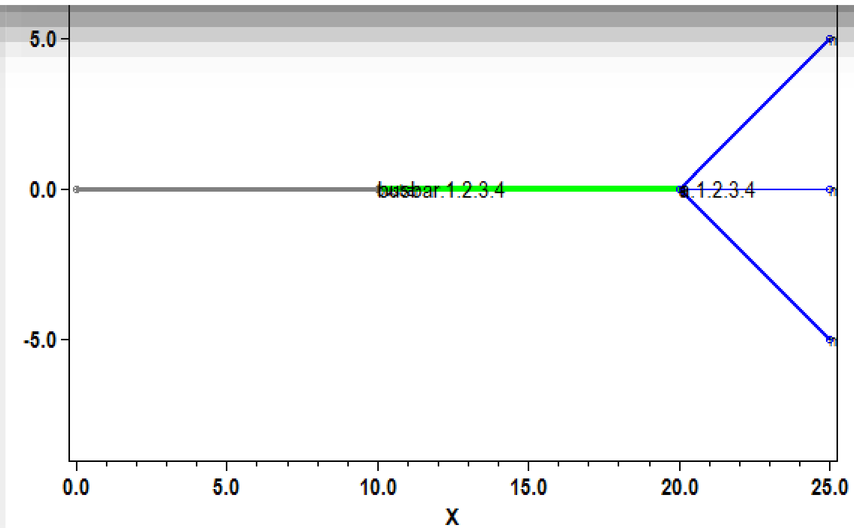
- Load Multiplier
  - Set Loadmult = VALUE
  - Affects all peak values of loads. However, they have to have status=variable (which is the default)
  - Note that some modes do not consider the load multiplier

- Double the demand
  - Set Loadmult = 2.0
- Compare
  - plot circuit power dots=y labels=y C1=red

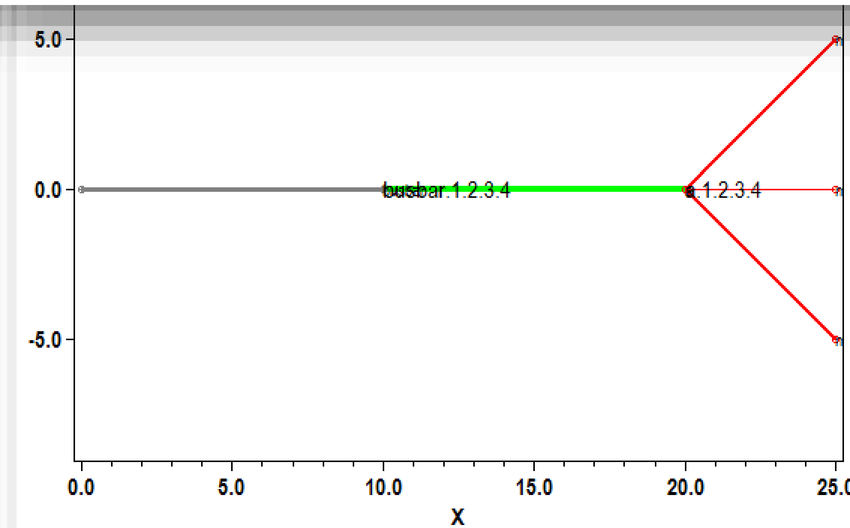
Explore 3/3

- Normal load vs Loadmult=2.0

```
Values for:  LINE.BUSBAR-A

Distance from Meter:    0.000 kM (  0.000 mi)
Power Flow:              3.980 kW + j    1.307 kvar

Phase Currents:          7.002    4.656    5.827    2.025 Amps
Phase Voltages(L-N):     0.240    0.240    0.240    0.000 kV
Per Unit Voltages:       0.240    0.240    0.240    0.000
```




```
Values for:  LINE.BUSBAR-A

Distance from Meter:    0.000 kM (  0.000 mi)
Power Flow:              7.984 kW + j    2.618 kvar

Phase Currents:         14.065    9.327   11.687    4.074 Amps
Phase Voltages(L-N):     0.240    0.240    0.240    0.000 kV
Per Unit Voltages:       0.240    0.240    0.240    0.000
```



15:20 to 15:45

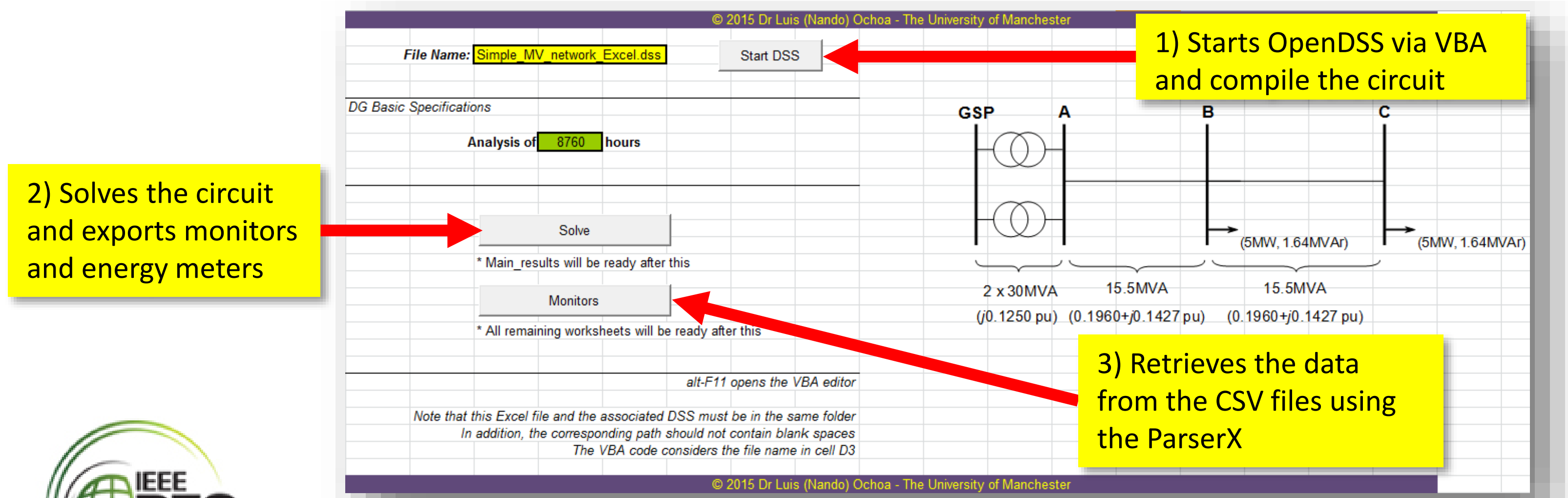
INTERFACING OPENDSS WITH MS EXCEL VBA AND MATLAB

Outline – Interfacing OpenDSS

- Familiarisation with VBA and Matlab and their interaction with OpenDSS
  - Use of simple MV network
  - Time-series simulations
- Adapting circuit data from VBA/Matlab
  - Changing load
  - Incorporation of generators
- Applying basic control on VBA
  - Extracting data in time-series simulations
  - Managing congestion using a var compensation

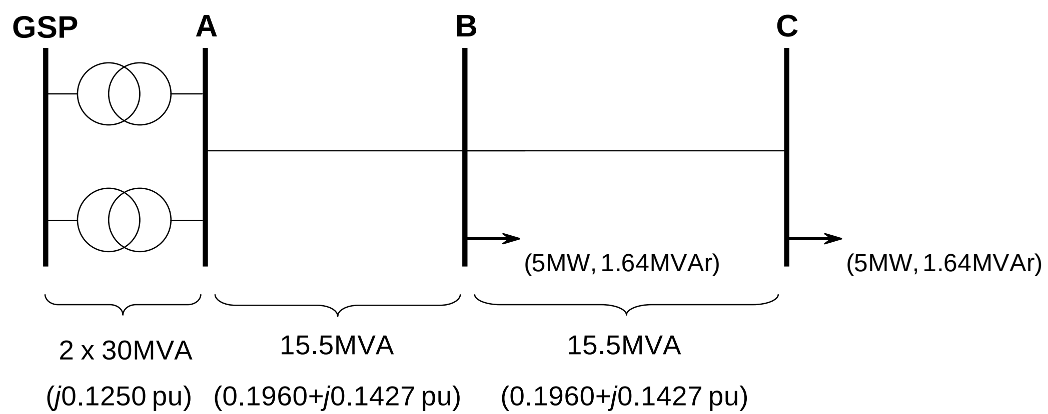
Simple MV network Excel

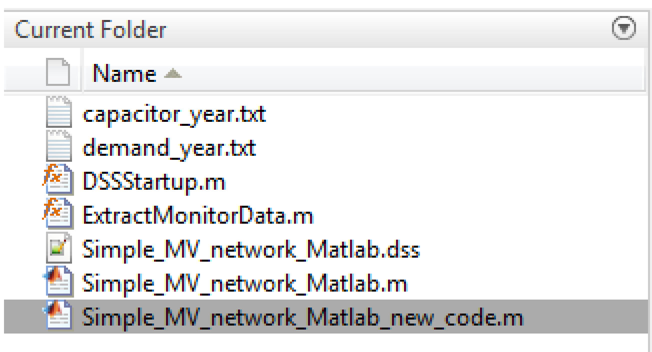
- This Excel file and the associated DSS and txt must be in the same folder
  - To be copied in a folder without blank spaces (e.g., C:\UoM\)
- **Macros must be enabled**
- The VBA code considers the file name in cell D3
  - Any changes requires the code to be updated accordingly

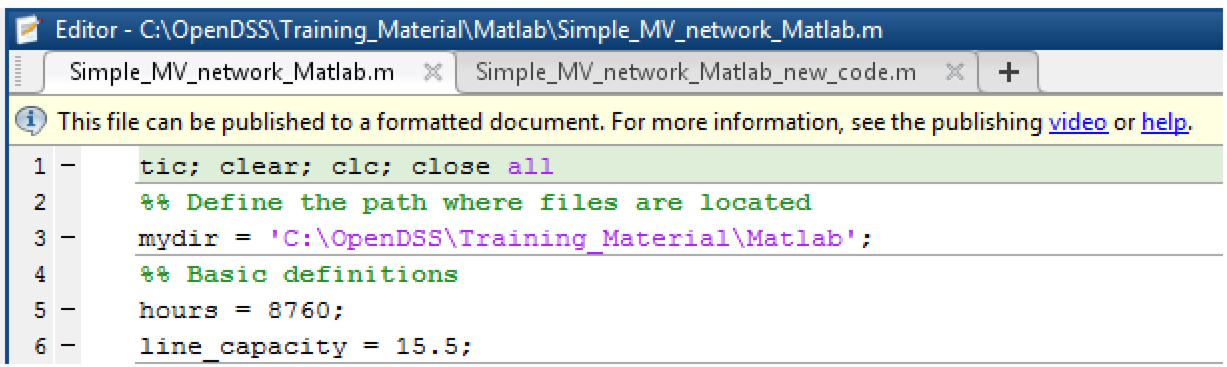


Simple MV network Matlab

- This Matlab file and the associated DSS and txt must be in the same folder







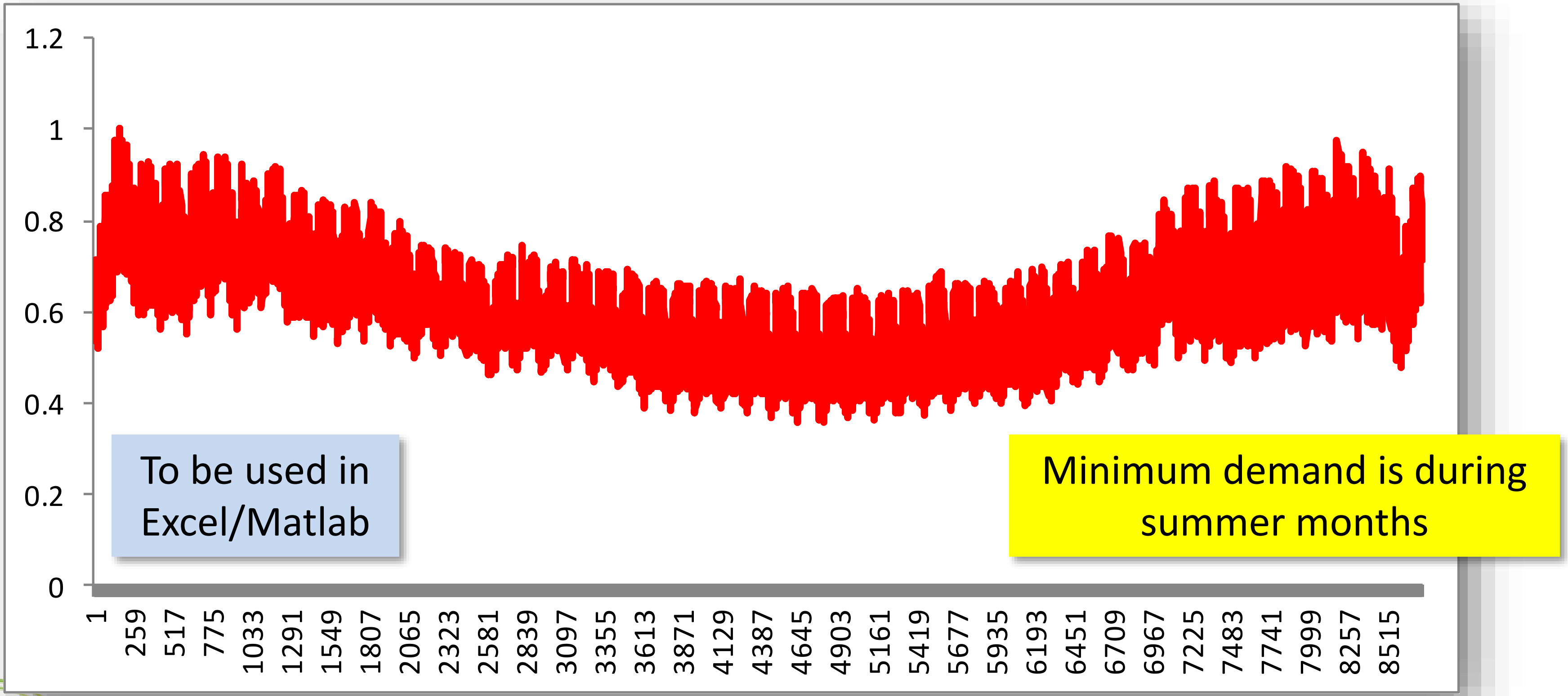
Simple MV network Excel/Matlab.DSS

```
new monitor.LineA-B_A element=line.lineA-B terminal=1 mode=1 ppolar=no !active and reactive power
new monitor.LineA-B_B element=line.lineA-B terminal=2 mode=1 ppolar=no !active and reactive power

  !! meters (to check energy exports/imports, losses)
new energymeter.GSP element=transformer.SS terminal=1
new energymeter.mloadB element=line.conloadB terminal=2
new energymeter.mloadC element=line.conloadC terminal=2

  !! voltage bases to have per unit results when visualising reports
set voltagebases=[33.0 132.0]
calcvoltagebases

!! commands below to be used only with the OpenDSS 'direct' interface - if using Matlab, they should be deleted

!set controlmode=time
!set mode=yearly

!solve

!export monitors SS
!export monitors Vbb
!export monitors Tap
!export meters
```

Controlmode, mode, solve and exports are all controlled by the VBA/Matlab code

“demand_year.txt”

- Normalised, aggregated hourly demand for South Scotland in 2003 (starting from 1st January, 8760 points)



Key Elements of the VBA Code

- Link between VBA and OpenDSS
  - OpenDSSengine.DLL and ParserX.OCX need to be registered

```
Option Explicit

Public DSSobj As OpenDSSengine.DSS
Public DSSText As OpenDSSengine.Text
Public DSSCircuit As OpenDSSengine.Circuit
Public DSSSolution As OpenDSSengine.Solution
Public DSSControlQueue As OpenDSSengine.CtrlQueue
Public Parser As ParserXControl.ParserX
```

If the Parser has not been registered, an error will come up

- Start of OpenDSS

```
Private Sub CommandButton1_Click()

    ' Create a new instance of the DSS
    Set DSSobj = New OpenDSSengine.DSS

    ' Start the DSS
    If Not DSSobj.Start(0) Then
        MsgBox "DSS Failed to Start"
    Else
        MsgBox "DSS Started successfully"
        ' Assign a variable to the Text interface for easier access
        Set DSSText = DSSobj.Text
    End If
```

This is simply to double check the DLL has been registered

Key Elements of the VBA Code

- Reading and compiling the pre-created circuit
- Adopting the circuit through VBA

```
Private Sub CommandButton6_Click()
    Dim stime As Single

    ' Always a good idea to clear the DSS when loading a new circuit
    DSSText.Command = "clear"
    ' Compile the script in the file listed under "fname" cell on the main form
    DSSText.Command = "compile " + Trim(ActiveWorkbook.Path) + "\" + Trim(Range("D3").Value)
    ' OBS. the folder name cannot have blank spaces

    ' The Compile command sets the current directory the that of the file
    ' Thats where all the result files will end up.

    ' Assign a variable to the Circuit interface for easier access
    Set DSSCircuit = DSSobj.ActiveCircuit
    Set DSSSolution = DSSCircuit.Solution
    Set DSSControlQueue = DSSCircuit.CtrlQueue
```

The command line for the compilation (same as "CTRL+D") can be improved to cater for folder names with blank spaces

Key Elements of the VBA Code

- Setting the controlmode and mode
- Solving the circuit and exporting the results

```
    runHours = Range("E8").Value

    stime = Timer

    DSSText.Command = "Set ControlMode=Time"
    DSSText.Command = "Set Mode=yearly number=" + Trim(Str(runHours)) 'yearly analysis
    DSSText.Command = "Reset" 'resetting all energy meters and monitors

    DSSobj.AllowForms = False 'no "solution progress" window
    DSSobj.ActiveCircuit.Solution.Solve

    MsgBox ("Total time " + Trim(Str(Timer - stime)))

    DSSText.Command = "Export monitors SS"
    DSSText.Command = "Export monitors Vbb"
    DSSText.Command = "Export monitors Tap"
    DSSText.Command = "Export monitors LineA-B_A"
    DSSText.Command = "Export monitors LineA-B_B"

    EMeters

End Sub
```

Many commands can either be done with 'typical' text or through the COM interface
Try to use the latter as much as possible so errors can be spotted

Key Elements of the VBA Code

- Using the ParserX

```
' >>>> time series results (P and Q) for GSP
Set WorkingSheet = Worksheets("GSP_PQ")
'using ParserX
Set Parser = Nothing ' destroy old object should it already exist
Set Parser = New ParserXControl.ParserX
Parser.AutoIncrement = True
FileNum = FreeFile
i = 0
Open "example_Mon_ss.csv" For Input As #FileNum
Line Input #FileNum, s  ' skip first line
Do While Not EOF(FileNum)
    Line Input #FileNum, s
    Parser.CmdString = s
    i = i + 1
    iextra = Parser.IntValue 'hours
    iextra = Parser.IntValue 'seconds
    GSP_PQ(i, 1) = Parser.DblValue / 1000 'originally kW
    GSP_PQ(i, 2) = Parser.DblValue / 1000 'originally kvar
    GSP_PQ(i, 3) = Parser.DblValue / 1000 'originally kW
    GSP_PQ(i, 4) = Parser.DblValue / 1000 'originally kvar
    GSP_PQ(i, 5) = Parser.DblValue / 1000 'originally kW
    GSP_PQ(i, 6) = Parser.DblValue / 1000 'originally kvar
    GSP_PQ(i, 7) = GSP_PQ(i, 1) + GSP_PQ(i, 3) + GSP_PQ(i, 5)
    GSP_PQ(i, 8) = GSP_PQ(i, 2) + GSP_PQ(i, 4) + GSP_PQ(i, 6)
Loop

'copying ranges is much faster than cell by cell
Set rangex = WorkingSheet.Range("B3", WorkingSheet.Cells(runHours + 2, 9))
rangex.Value = GSP_PQ
```

1) Parser reads a line in the CSV file
2) An internal matrix stores the data
3) The matrix is copied to the pre-defined area in Excel

Much more time efficient than opening CSV files as ‘spreadsheets’

Key Elements of the Matlab Code

This initializes OpenDSS

- Start of OpenDSS

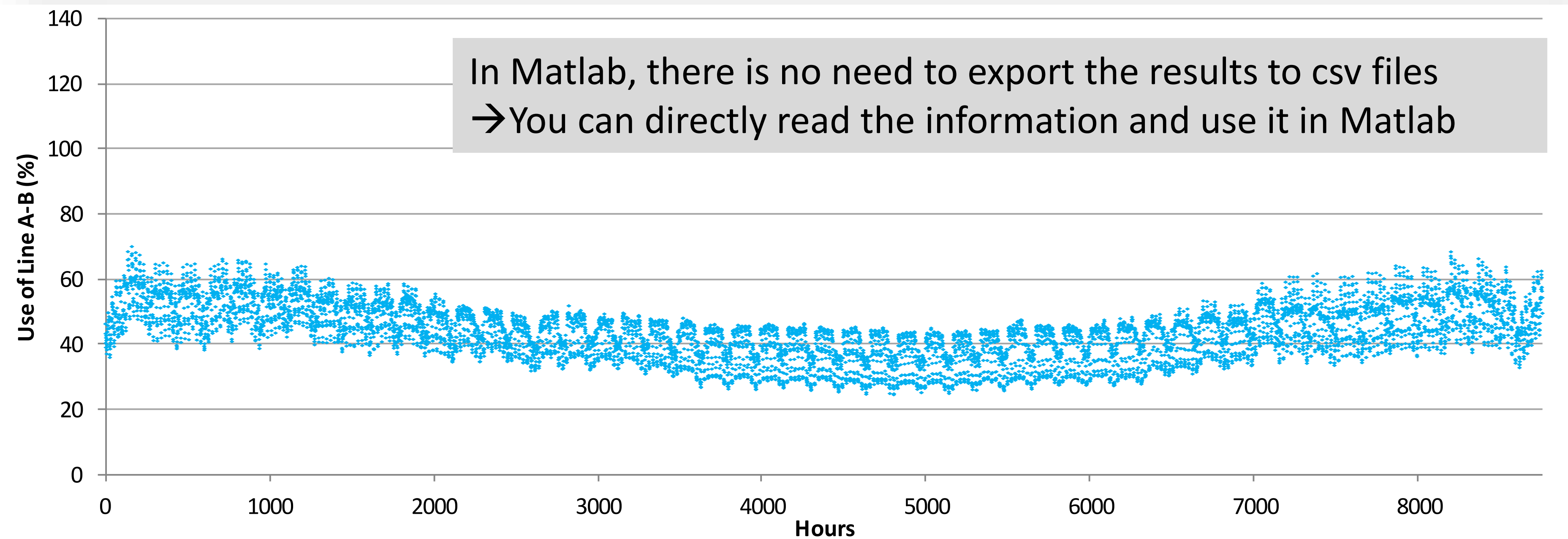
```
%% *****************************************************************************************************
%                                         Initialize OpenDSS
% *****************************************************************************************************
[DSSStartOK, DSSObj, DSSText] = DSSStartup(mydir);
if ~DSSStartOK
    disp('Unable to start the OpenDSS Engine')
    return
end
DSSText = DSSObj.Text;                                                          %   Set up the Text
DSSText.Command = 'clear';                                                      %   Clear text command
DSSText.Command = ...
    sprintf('Compile (%s%sSimple_MV_network_Matlab.dss)',mydir,'\');
DSSCircuit = DSSObj.ActiveCircuit;                                              %   Set up the Circuit
DSSSolution = DSSCircuit.Solution;                                              %   Set up the Solution
ControlQueue = DSSCircuit.CtrlQueue;                                            %   Set up the Control
DSSObj.AllowForms = 0;                                                          %   no "solution progress" window
```

This line scmpiles the DSS file (i.e., Control+D)

These lines set the interface between Matlab and OpenDSS

Use of Line A-B

- Results without changes to the VBA/Matlab code or DSS file
- Data from the exported CSV files was used for this
  - Capacity of line A-B is 15.5 MVA



Increasing the Loads (x2) using VBA/Matlab

There are many others, explore them ☺

- Load Multiplier Option
  - This option multiplies **ALL loads** by a factor defined by the user
  - This only works with loads not specified as 'fixed'

Excel

```
Dim demandgrowth As Double
demandgrowth = Range("H8").Value
```

  - Suboption 1

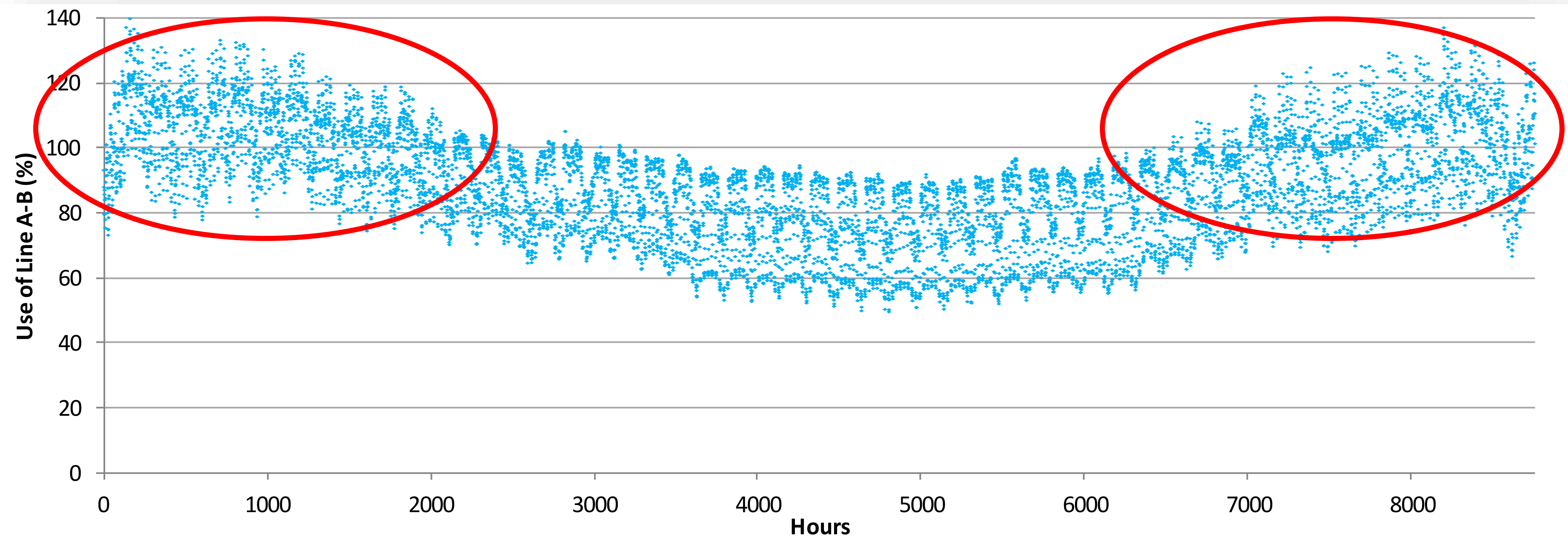
Excel → `DSSText.Command = "Set Loadmult=" + Trim(Str(demandgrowth))`
Matlab → `DSSText.Command = ['Set Loadmult=' num2str(Dgrowth)];`

  - Suboption 2

Excel → `DSSSolution.LoadMult = demandgrowth`
Matlab → `DSSSolution.LoadMult = Dgrowth;`

Use of Line A-B

- demandgrowth equal to 2.0
  - Now we are exceeding the capacity limit of the line
  - Solutions that will be explored: generation and VAr compensation



Inserting a Distributed Generator – Excel

- First, to create even further 'problems' we will increase the reactive power demand of bus C

```
DSSCircuit.Loads.Name = "loadC"
DSSCircuit.Loads.kvar = 5000
```

- Now, let's assume a generator with constant output of 7MW (unity power factor) and a growth factor defined by the user
- Step one, the generator is created

```
DSSText.Command = "new generator.gen1 bus1=C phases=3 kV=33
kW=7000 kvar=0 model=1 status=fixed"
```

This command line is very specific but quicker for this particular case

- Step two, the factor is inserted

```
generationgrowth = Range("H9").Value
DSSCircuit.Generators.Name = "gen1"
DSSCircuit.Generators.kW = DSSCircuit.Generators.kW * generationgrowth
```

Inserting a Distributed Generator – Matlab

- First, to create even further ‘problems’ we will increase the reactive power demand of bus C

```
DSSCircuit.Loads.Name = ‘loadC’;
DSSCircuit.Loads.kvar = 5000;
```

- Now, let’s assume a generator with constant output of 7MW (unity power factor) and a growth factor defined by the user
- Step one, the generator is created

```
DSSText.Command = ‘new generator.gen1 bus1=C phases=3 kV=33
kW=7000 kvar=0 model=1 status=fixed’;
```

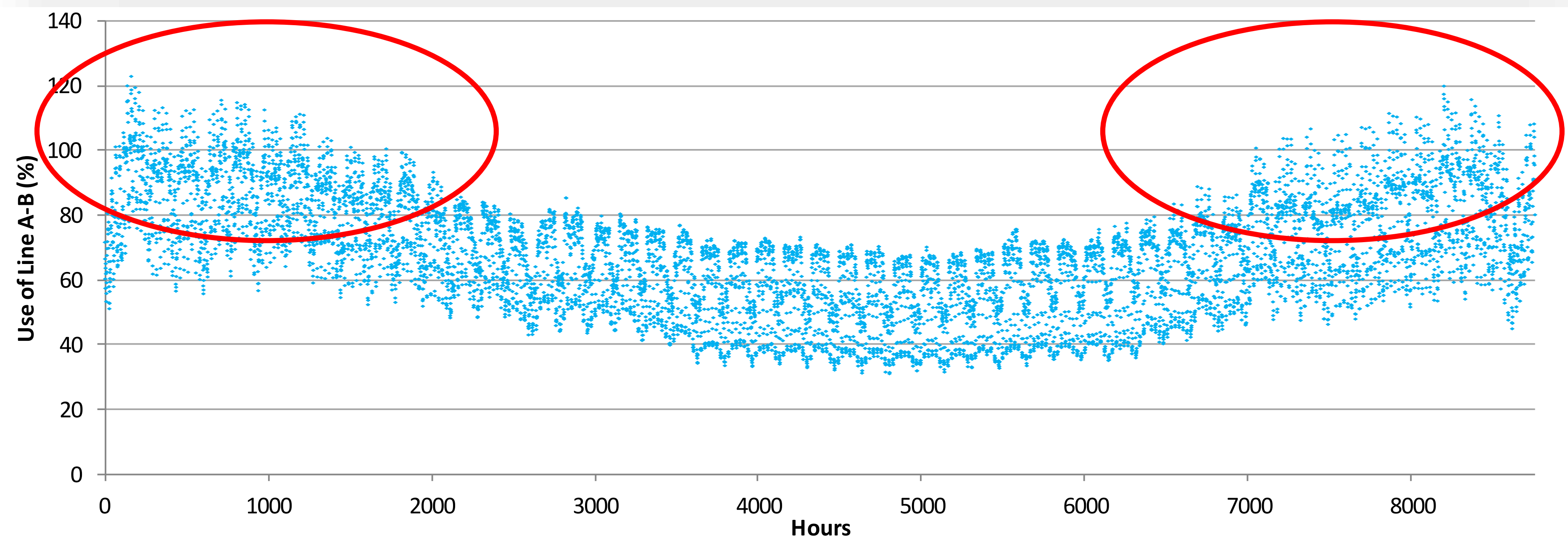
- Step two, the factor is inserted

Very specific command again but quicker for this particular case

```
DSSCircuit.Generators.Name = ‘gen1’;
DSSCircuit.Generators.kW = DSSCircuit.Generators.kW * Ggrowth
```

Use of Line A-B

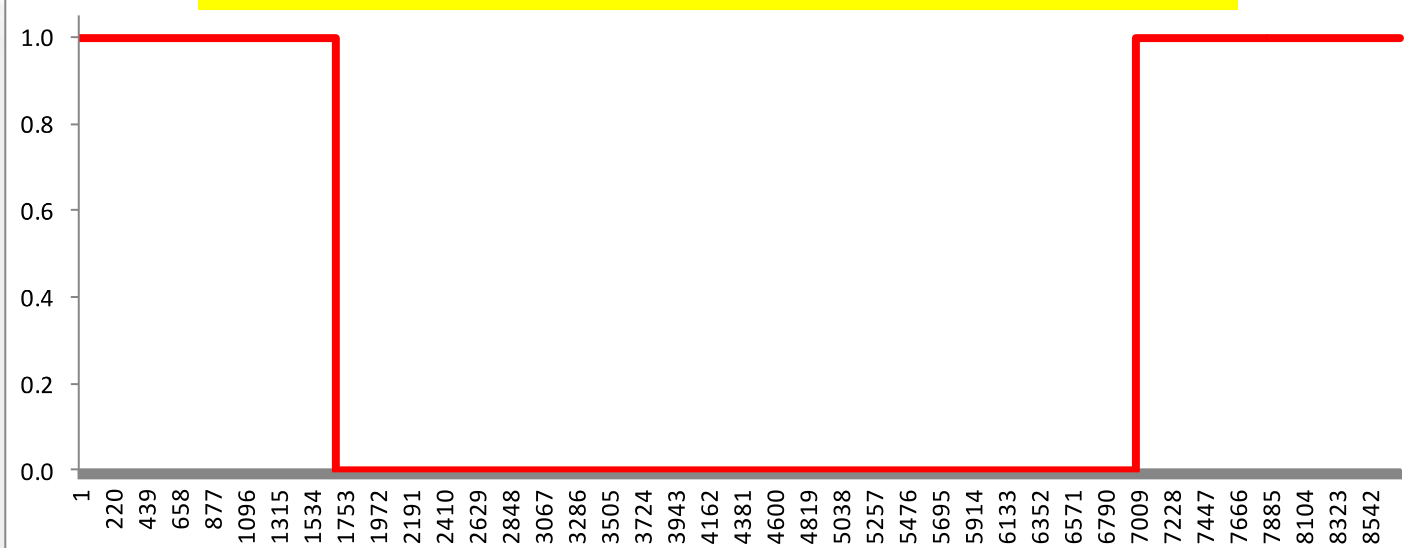
- demandgrowth equal to 2.0, 7MW of distributed generation (unity power factor), generationgrowth equal to 1.0, and modified load at bus C
  - The line still has some congestion issues but much less than before, thanks to the local generator



Managing Congestion using VAr Compensation

- First we can explore the use of ‘time-based’ reactive power compensation
- Given that the problem is during the winter months (0-1700 and 7000-8760 hours), create a load shape that would switch a capacitor on accordingly (on: 1, off: 0)

Create a txt file named “capacitor_year.txt”



Managing Congestion using VAr Compensation

- We create the corresponding load shape

Excel → DSSText.Command = "new loadshape.capacitor npts=8760 interval=1 csvfile=capacitor_year.txt"

Matlab → DSSText.Command = ‘new loadshape.capacitor npts=8760 interval=1 csvfile=capacitor_year.txt’;

–

You may have noticed that " is changed in Matlab for ‘

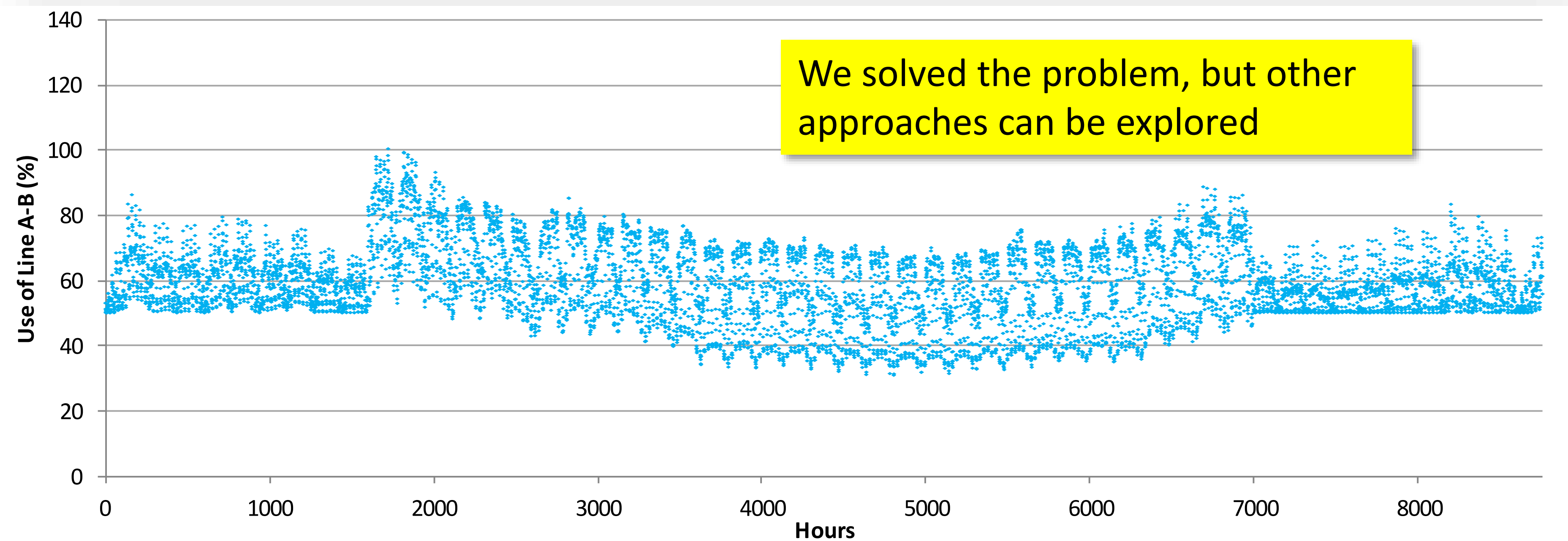
- Then, the capacitor. For simplicity, in this case it is actually a load injecting reactive power (hence, the negative sign!).

Excel → DSSText.Command = "new load.capacitor bus1=C phases=3 kV=33 kW=0 kvar=-7000 model=1 status=variable year=capacitor“

Matlab → DSSText.Command = ‘new load.capacitor bus1=C phases=3 kV=33 kW=0 kvar=-7000 model=1 status=variable year=capacitor’;

Use of Line A-B

- demandgrowth equal to 2.0, 7MW of distributed generation (unity power factor), generationgrowth equal to 1.0, modified load at bus C, and a 7MVAr capacitor with a time-specific load shape
  - The line has no congestion issues!



Managing Congestion using VAr Compensation

- Create the capacitor

Excel → DSSText.Command = "new load.capacitor bus1=C phases=3 kV=33 kW=0 kvar=0 model=1“
Matlab → DSSText.Command = ‘new load.capacitor bus1=C phases=3 kV=33 kW=0 kvar=0 model=1’

kvar=0 since we’ll be controlling it

- Comment the solve and yearly mode lines:

Excel → *' DSSText.Command = "Set Mode=yearly number=" + Trim(Str(runHours))*
Excel → *' DSSobj.ActiveCircuit.Solution.Solve*
Matlab → *%DSSText.Command = ['Set Mode = yearly number = ' num2str(hours)];*
Matlab → *%DSSobj.ActiveCircuit.Solution.Solve*

Managing Congestion using VAr Compensation

- First we need to make OpenDSS run every hour in a way that we can actually get 'inside' each hour

Excel → DSSText.Command = "Set Mode=yearly number=1“
Matlab → DSSText.Command = 'Set Mode=yearly number=1'

Solving the circuit one hour at a time

- Then produce the corresponding loop (**example Excel**)

```
For i = 1 To runHours
    DSSobj.ActiveCircuit.Solution.Solve
Next i
```

This loop allows us now to do things after each hour

These changes should lead to the same results as previously

Managing Congestion using VAr Compensation

- Before we can implement some sort of control we need to have observability, i.e., check the power flows through line A-B

1) Activate element, i.e., line A-B
2) Extract power flows (three-phase, and for each end of the element)

```
DSSCircuit.SetActiveElement ("line.lineA-B")
powerarray = DSSCircuit.ActiveCktElement.Powers
```

3) Only the complex power of one phase and one end is considered (**example Excel**)

```
kVAphaseA = (powerarray(LBound(powerarray)) ^ 2 +
powerarray(LBound(powerarray) + 1) ^ 2) ^ 0.5
```

4) Then we calculate the use of the line

```
useofAB = (3 * kVAphaseA) / 15500
```

Managing Congestion using VAr Compensation

- Now a control action can be implemented
  - This is the actual ‘smart’ part that comes from the researcher ☺
- Here, a simple set of rules will be considered
  - If the line is congested then we switch on the capacitor (7MVAr) for 5 hours
  - If after 5 hours there is still a problem, we consider another 5 hours and so on
  - If after 5 hours there is no problem, then we switch it off

Note that ‘counters’ will need to be defined

The control rule should be within the loop

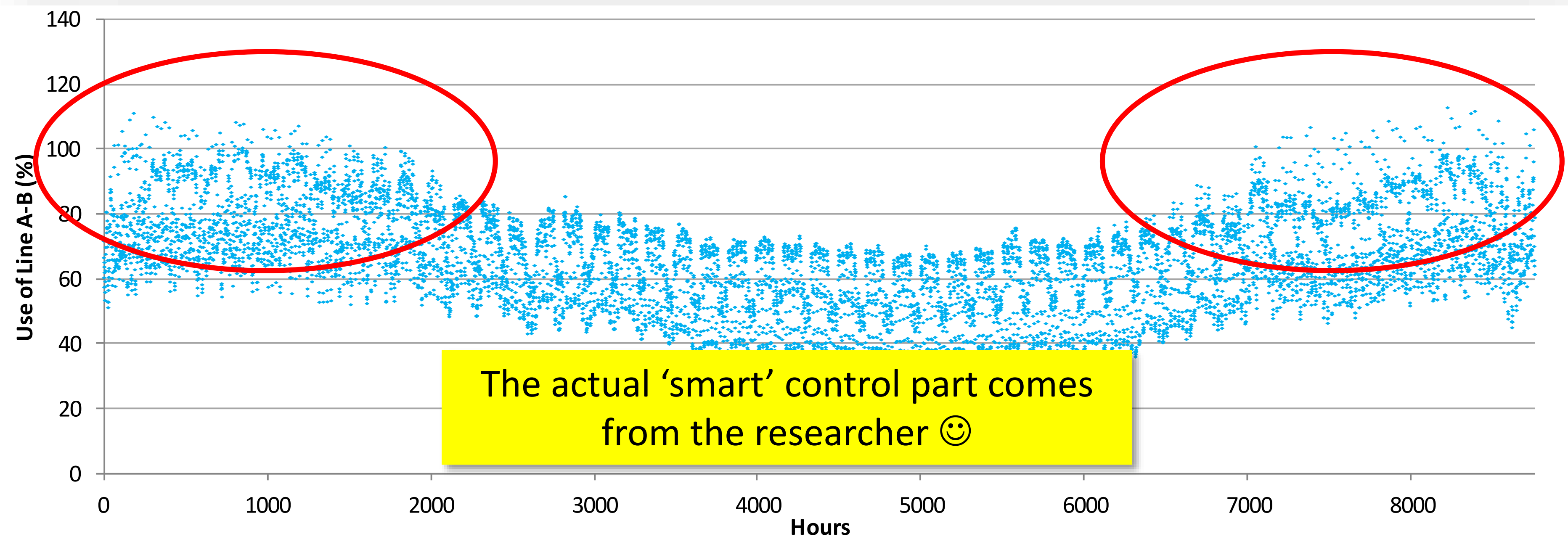
Managing Congestion using VAr Compensation

- Based on these rules we can write the following code (**Excel**)

```
If ijk = 0 Or ijk > 5 Then
  ijk = 0
  If useofAB > 1 Then
    DSSCircuit.Loads.Name = "capacitor"
    DSSCircuit.Loads.kvar = -7000
  ElseIf useofAB < 0.7 Then
    DSSCircuit.Loads.Name = "capacitor"
    DSSCircuit.Loads.kvar = 0
  End If
End If
DSSCircuit.Loads.Name = "capacitor"
If DSSCircuit.Loads.kvar = -7000 Then
  ijk = ijk + 1
End If
```

Use of Line A-B

- demandgrowth equal to 2.0, 7MW of distributed generation (unity power factor), generationgrowth equal to 1.0, modified load at bus C, and a controlled 7MVAr capacitor
  - The line still has some congestion issues → room for improvement



Key Remarks – Interfacing OpenDSS

- When using VBA/Matlab, controlmode, mode, solve and exports are all **controlled via VBA/Matlab**

- VBA/Matlab is extremely helpful for **management of data** from a large number of simulations

- VBA/Matlab is extremely helpful **to adapt circuit data**

- VBA/Matlab allows **controlling elements** in OpenDSS

Explore redirect *filename*

- Redirects the OpenDSS input stream to the designated file that is expected to contain DSS commands.
  - The file is processed as if the commands were entered directly into the command line.
  - Extremely helpful to call a stored **LineCode library**, and to call many **Lines** in a circuit

Excel → DSSText.Command = 'redirect LineCode.txt';
Matlab → DSSText.Command = 'redirect LineCode.txt';

Try it ☺

30 min

16:15 to 16:45

STOCHASTIC IMPACT ANALYSIS OF LOW CARBON TECHNOLOGIES

Mr Alejandro Navarro (PhD Student)

ENWL "LV Network Solutions" Project

Outline – Stochastic Impact Analysis of LCTs

- LV Network Solutions (LVNS) Project
  - Objective, how to achieve this
- Impact Assessment Methodology
- Creation of Low Carbon Technology (LCT) profiles
- How is this stochastic analysis actually done?
  - Impact Assessment Application
  - Single feeder, metrics and multi-feeder
- Remarks

LV Network Solutions (LVNS)

Total Cost
US$2.3m

Project run by
Electricity North West Limited

http://www.enwl.co.uk/lvns

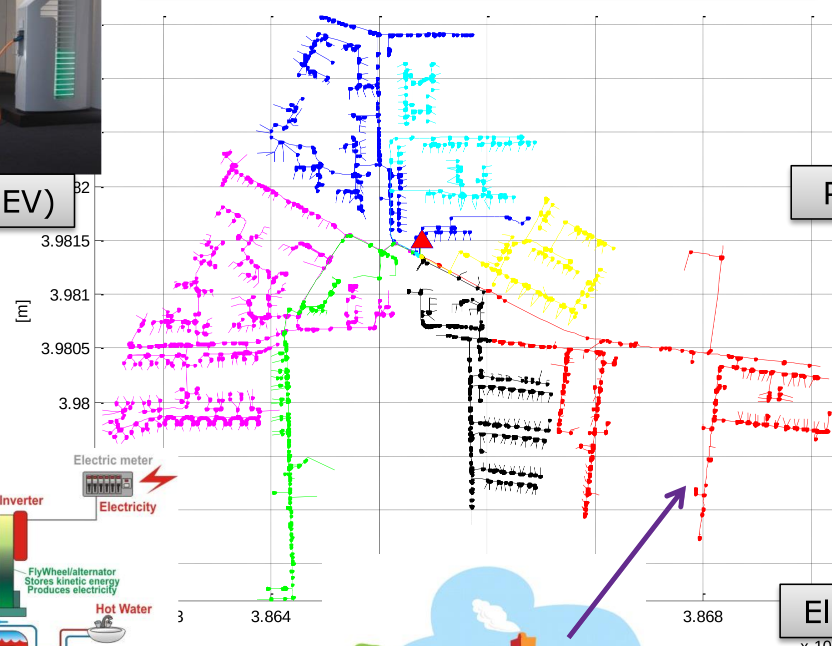
LV Network Solutions (LVNS)

- To understand the behaviour and needs of future LV networks with high penetrations of low carbon technologies (LCTs)

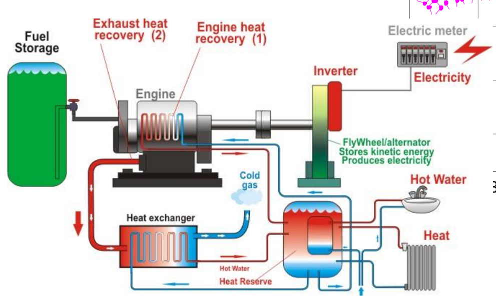
Different behaviour and sizes of loads and LCT along the day

Electric Vehicles (EV)

Photovoltaic Panels (PV)



Electric Heat Pumps (EHP)



Micro combine heat & power (uCHP)

Residential Loads

How to achieve this objective?

- Considerations
  - **Monte Carlo** analysis to cope with the **uncertainty** (LCT size and location, sun profile, heat requirements, EV utilization, load profile, etc.)
  - **Time-Series** Analysis (5-min synthetic data)
  - Three-phase **unbalanced** power flow (OpenDSS)
- Input data
  - Load and **LCT profiles**
  - Real UK **networks** (topology and characteristics)

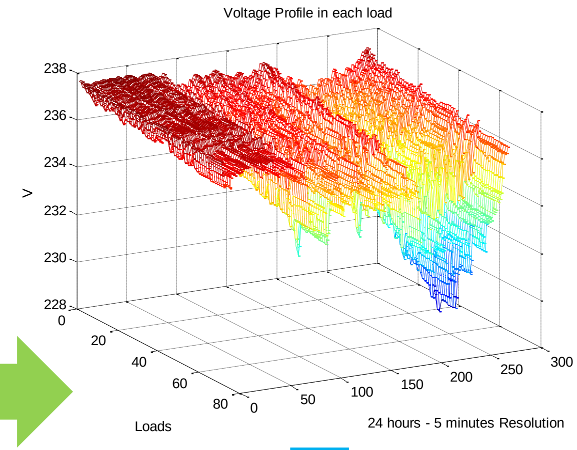
Impact Assessment Methodology

This process is repeated 100 times for each feder and penetration level

Loads
- Random allocation for each customer node

LCT
- Random allocation of sites and sizes

Power Flow
- Time Series Simulation
- 3 Phase four wire power flow



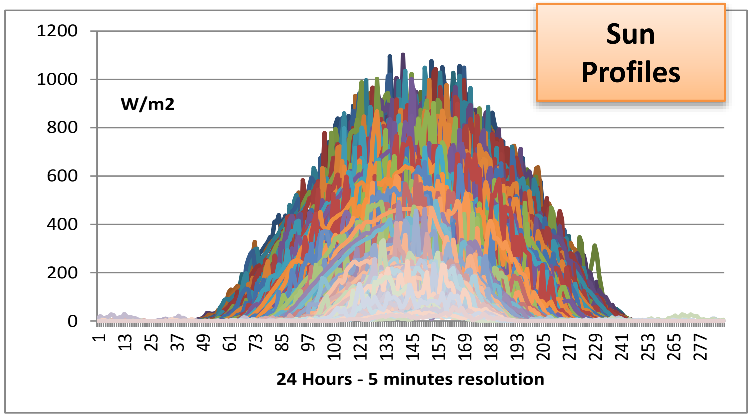
Results Storage

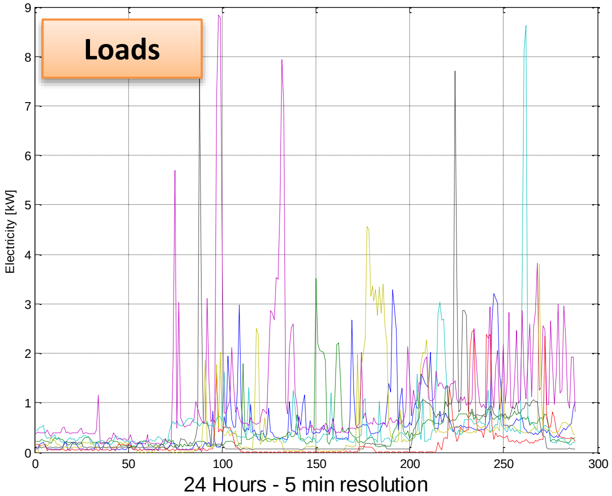
Impact Assessment

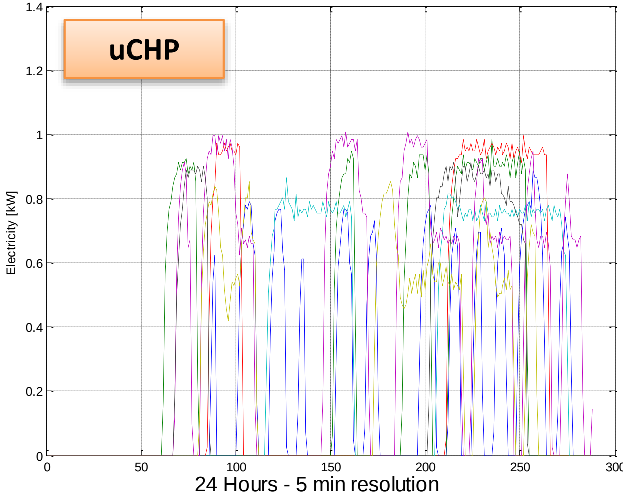
- Impacts metrics:
  - Customers with voltage problems: defined according to the Standard BS EN 50160.
  - Utilization level of the head of the feeder: hourly maximum current divided by the ampacity.

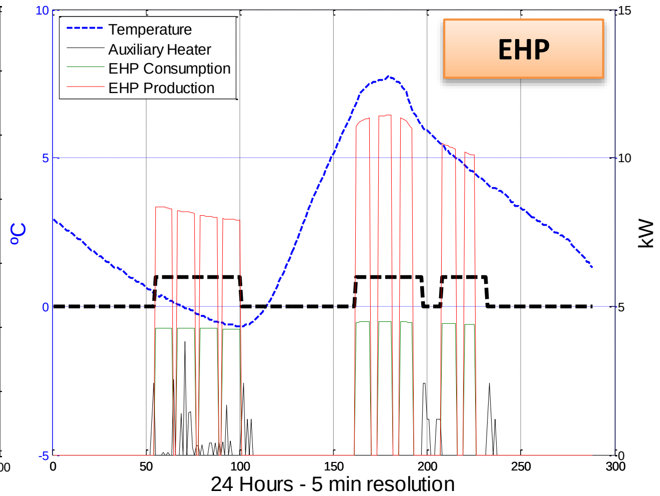
Creation of Realistic Load and LCT Profiles

- Pools of thousands of different individual residential profiles with a granularity of 5 minutes are created for:
  - Loads
  - Photovoltaic Panels
  - Electric Vehicles
  - Micro CHP Units
  - Electric Heat Pumps









How is this stochastic analysis actually done?

MATLAB

OpenDSS driver
Random variables creator
Results Analyser

COM

DSS

Time-Series, three-phase
power flow solver

1-Input Data acquisition

2-Profiles Random Allocation

3-Power flow Simulation

4-Result Visualization

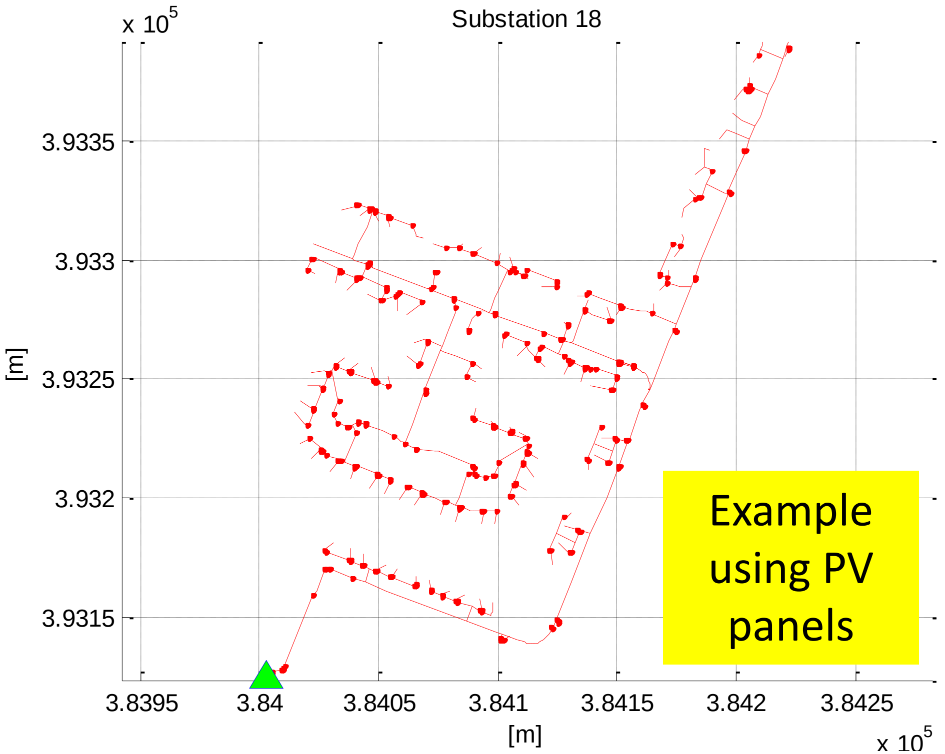
Simple ☺

*Can also be done using VBA

Impact Assessment Application

- The impact metrics are quantified for the real feeder shown in the figure.
- The PV, EV, EHP and uCHP are implemented and studied.
- Voltage reference at the bus bar (secondary side):

Vsec = 241 Vfn (1.05*Vnom)



2.2 km (including services cables)
94 loads

MANCHESTER 1824

Stochastic Impact Analysis of LCTs: Input data

LineCode.txt - Notepad

File Edit Format View Help

```
New LineCode.2c_.007 nphases=3 R1=3.97 X1=0.099 R0=3.97 X0=0.099 C1=0 C0=0 Units=km
New LineCode.2c_.0225 nphases=3 R1=1.257 X1=0.085 R0=1.257 X0=0.085 C1=0 C0=0 Units=km
New LineCode.2c_16 nphases=3 R1=1.15 X1=0.088 R0=1.2 X0=0.088 C1=0 C0=0 Units=km
New LineCode.35_SAC_XSC nphases=3 R1=0.868 X1=0.092 R0=0.76 X0=0.092 C1=0 C0=0 Units=km
New LineCode.4c_.06 nphases=3 R1=0.469 X1=0.075 R0=1.581 X0=0.091 C1=0 C0=0 Units=km
New LineCode.4c_.1 nphases=3 R1=0.274 X1=0.073 R0=0.959 X0=0.079 C1=0 C0=0 Units=km
New LineCode.4c_.35 nphases=3 R1=0.089 X1=0.0675 R0=0.319 X0=0.076 C1=0 C0=0 Units=km
New LineCode.4c_185 nphases=3 R1=0.166 X1=0.068 R0=0.58 X0=0.078 C1=0 C0=0 Units=km
New LineCode.4c_70 nphases=3 R1=0.446 X1=0.071 R0=1.505 X0=0.083 C1=0 C0=0 Units=km
New LineCode.4c_95_SAC_XC nphases=3 R1=0.322 X1=0.074 R0=0.804 X0=0.093 C1=0 C0=0 Units=km
```

+

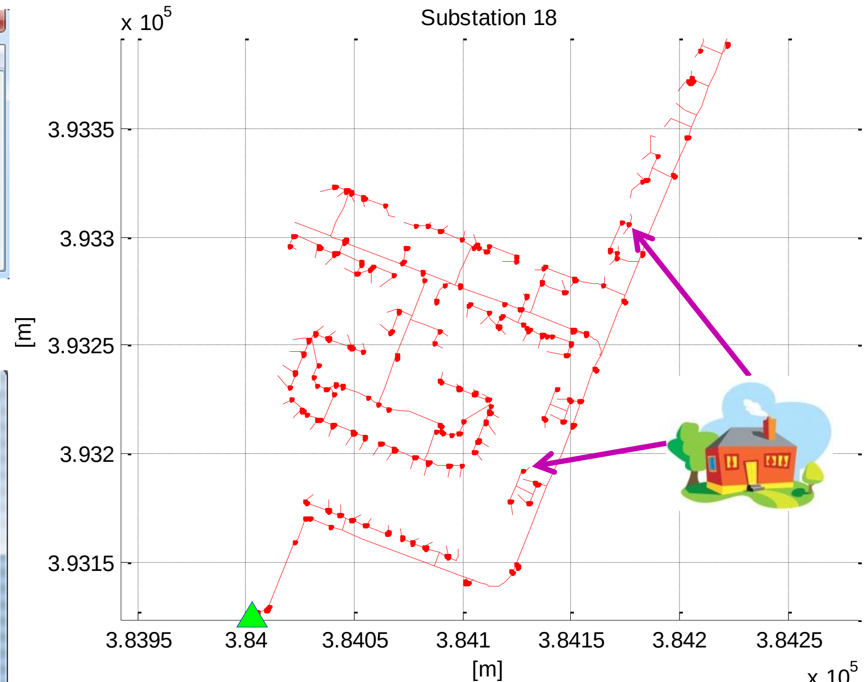
Lines.txt - Notepad

File Edit Format View Help

```
New Line.LINE1 Bus1=1 Bus2=2 phases=3 Linecode=4c_70 Length=1.098 Units=m
New Line.LINE2 Bus1=2 Bus2=3 phases=3 Linecode=4c_70 Length=0.11511 Units=m
New Line.LINE3 Bus1=3 Bus2=4 phases=3 Linecode=4c_70 Length=0.10784 Units=m
New Line.LINE4 Bus1=4 Bus2=5 phases=3 Linecode=4c_70 Length=0.094021 Units=m
New Line.LINE5 Bus1=5 Bus2=6 phases=3 Linecode=4c_70 Length=0.14812 Units=m
New Line.LINE6 Bus1=6 Bus2=7 phases=3 Linecode=4c_70 Length=10.0863 Units=m
New Line.LINE7 Bus1=7 Bus2=8 phases=3 Linecode=4c_70 Length=0.45175 Units=m
New Line.LINE8 Bus1=8 Bus2=9 phases=3 Linecode=4c_70 Length=0.376 Units=m
New Line.LINE9 Bus1=9 Bus2=10 phases=3 Linecode=4c_70 Length=0.32997 Units=m
New Line.LINE10 Bus1=10 Bus2=11 phases=3 Linecode=4c_70 Length=0.24622 Units=m
New Line.LINE11 Bus1=11 Bus2=12 phases=3 Linecode=4c_70 Length=0.23114 Units=m
New Line.LINE12 Bus1=12 Bus2=13 phases=3 Linecode=4c_70 Length=0.14213 Units=m
New Line.LINE13 Bus1=13 Bus2=14 phases=3 Linecode=4c_70 Length=0.35384 Units=m
New Line.LINE14 Bus1=14 Bus2=15 phases=3 Linecode=4c_70 Length=2.8664 Units=m
New Line.LINE15 Bus1=15 Bus2=16 phases=3 Linecode=2c_.007 Length=0.09654 Units=m
New Line.LINE16 Bus1=15 Bus2=17 phases=3 Linecode=4c_70 Length=3.476 Units=m
New Line.LINE17 Bus1=16 Bus2=18 phases=3 Linecode=2c_.007 Length=0.044045 Units=m
New Line.LINE18 Bus1=17 Bus2=19 phases=3 Linecode=4c_70 Length=0.23917 Units=m
New Line.LINE19 Bus1=18 Bus2=20 phases=3 Linecode=2c_.007 Length=0.046174 Units=m
New Line.LINE20 Bus1=19 Bus2=21 phases=3 Linecode=4c_70 Length=0.26856 Units=m
New L ... phases=3 Linecode=2c_.007 Length=0.041976 Units=m
New L ... phases=3 Linecode=4c_70 Length=0.43727 Units=m
New L ... phases=3 Linecode=2c_.007 Length=0.051245 Units=m
New L ... phases=3 Linecode=4c_70 Length=0.7955 Units=m
New L ... phases=3 Linecode=2c_.007 Length=0.055902 Units=m
New L ... phases=3 Linecode=2c_16 Length=5.9049 Units=m
New L ... phases=3 Linecode=4c_70 Length=9.9686 Units=m
New L ... phases=3 Linecode=2c_.007 Length=0.054562 Units=m
New L ... phases=3 Linecode=2c_16 Length=1.6265 Units=m
New L ... phases=3 Linecode=2c_16 Length=1.5847 Units=m
New L ... phases=3 Linecode=4c_70 Length=2.6538 Units=m
New L ... phases=3 Linecode=2c_.007 Length=0.14629 Units=m
New Line.LINE33 Bus1=30 Bus2=34 phases=3 Linecode=2c
New Line.LINE34 Bus1=31 Bus2=35 phases=3 Linecode=2c
```

Don't forget the monitors

Information sent using the command redirect



Typical definitions (preferably before LineCode and Lines):

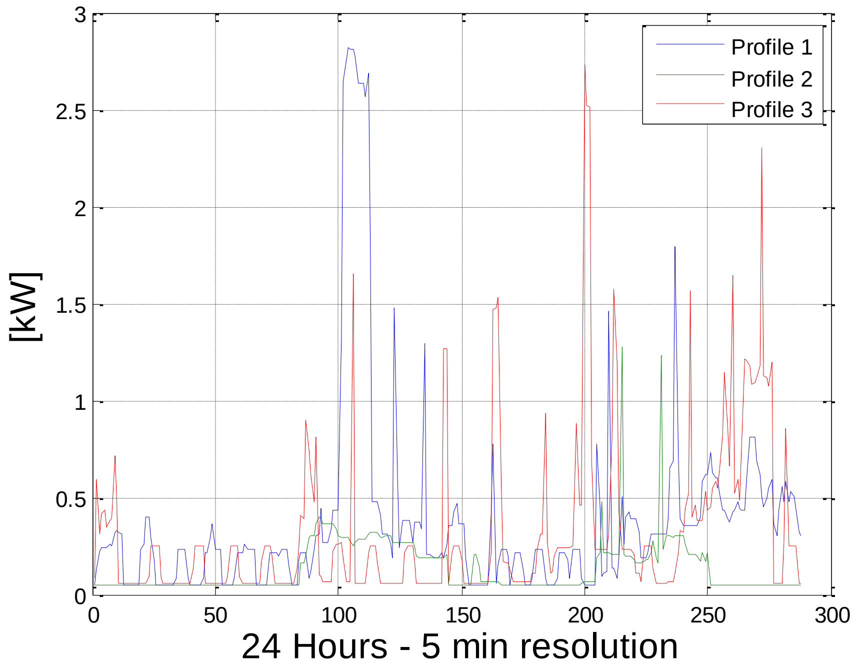
set datapath=C:\......
new circuit.LV ....
new transformer.LVSS .....

IEEE PES Power & Energy Society

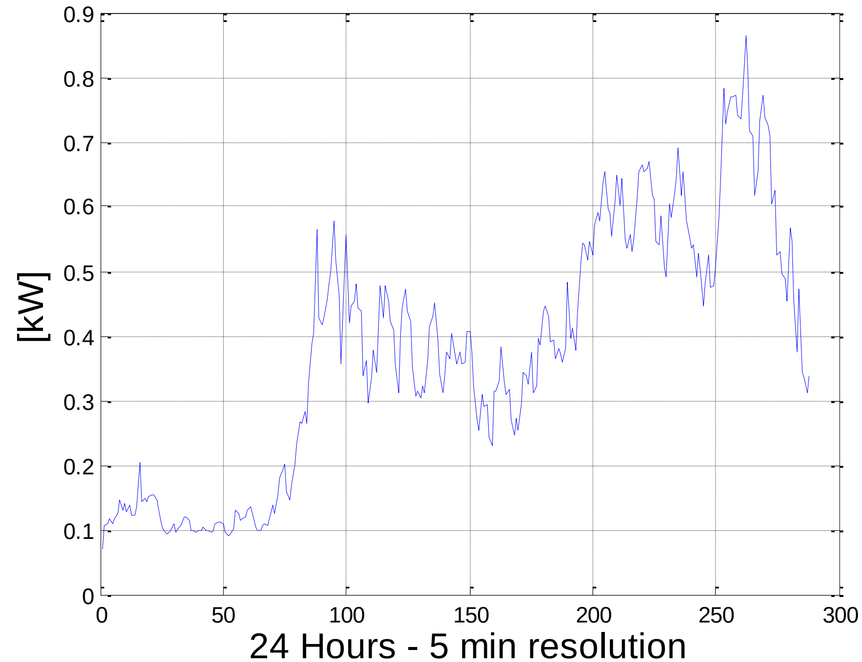
IEEE

Stochastic Impact Analysis of LCTs: Input data

Realistic Load Profiles*



Individual profiles



Average profile

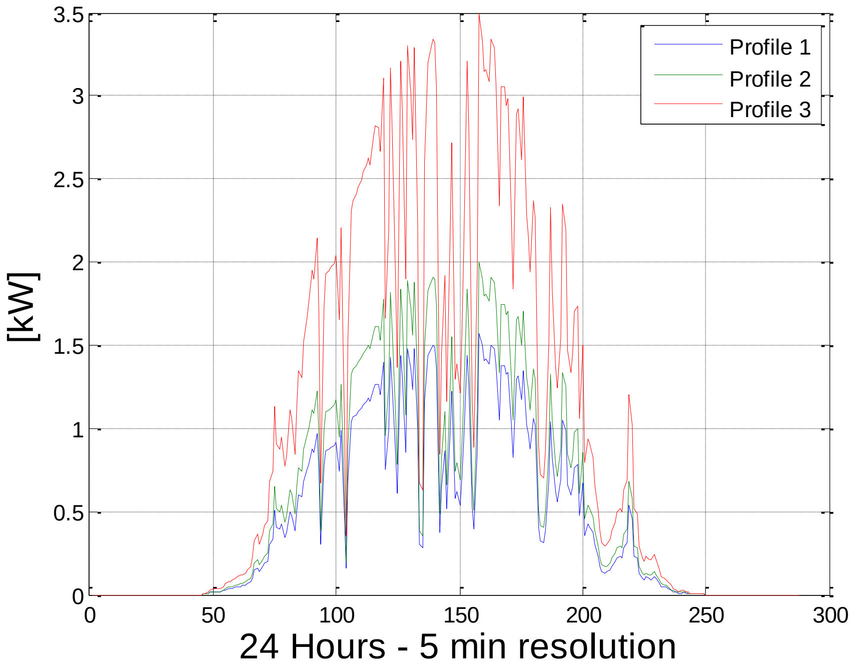
Information sent using the command redirect

This average is calculated among the 100 profiles provided

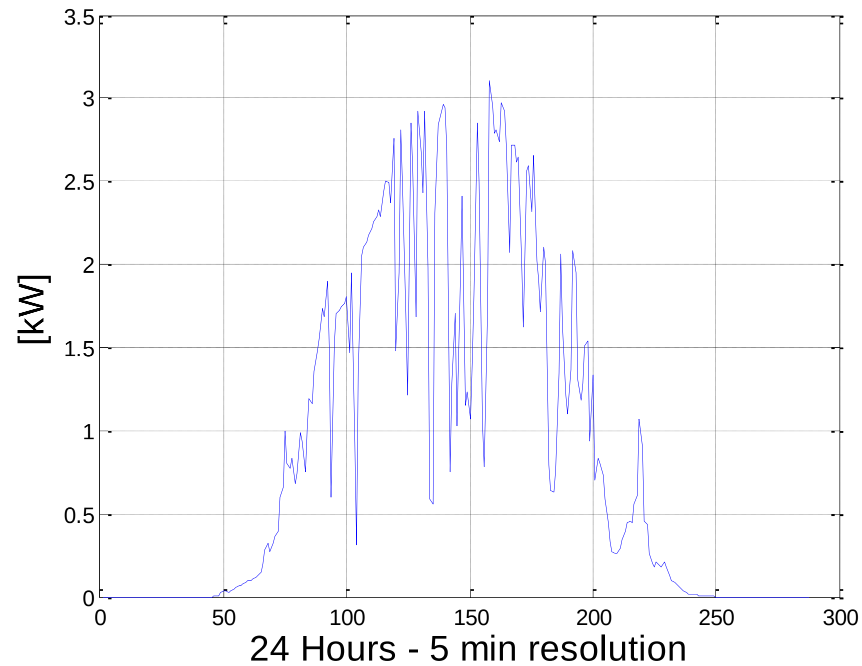
* I. Richardson, "Integrated High-resolution Modelling of Domestic Electricity Demand and Low Voltage Electricity Distribution Networks", PhD Thesis, University of Loughborough, 2011

Stochastic Impact Analysis of LCTs: Input data

Realistic PV Profiles*



Individual profiles



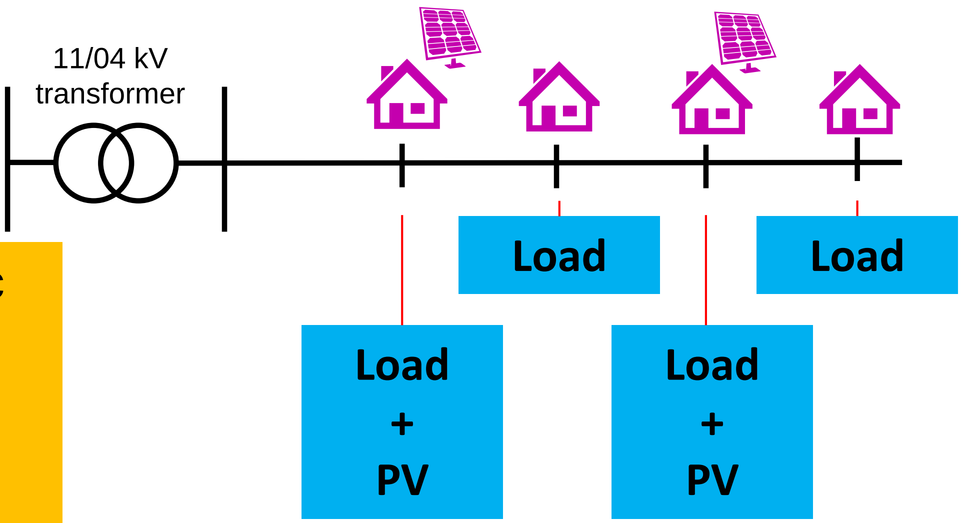
Average profile

Information sent using the command redirect

This average is calculated among the 100 profiles provided

* The University of Manchester, "The Whitworth Meteorological Observatory." [Online]. Available: http://www.cas.manchester.ac.uk/restools/whitworth/.

Stochastic Impact Analysis of LCTs: MATLAB



- MATLAB randomly selects a domestic profile for each house and sends it through the COM server to OpenDSS
  - ✓ Repeated for each house
- MATLAB randomly selects a house to allocated the LCT, its size, etc., and sends the corresponding LCT profile through the COM server to OpenDSS
  - ✓ Repeated for each house with LCT

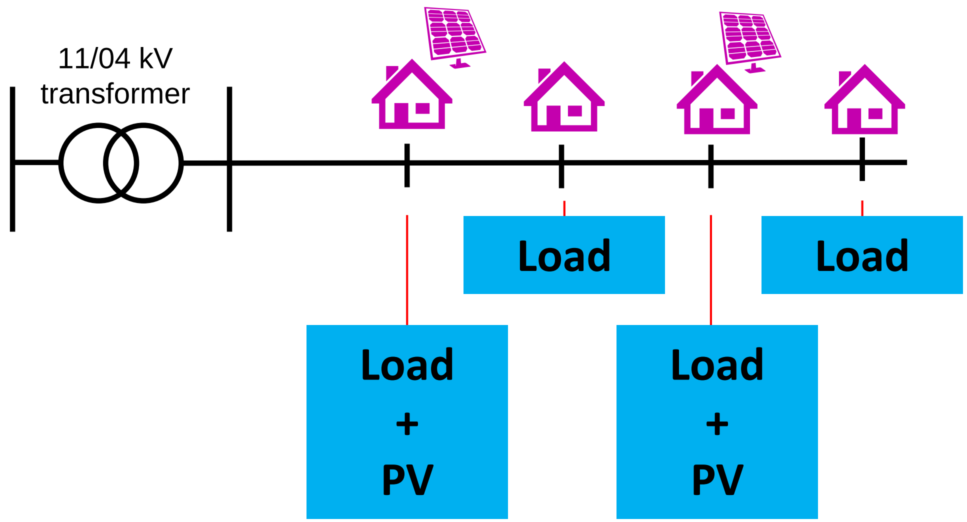
Random assignation of variables

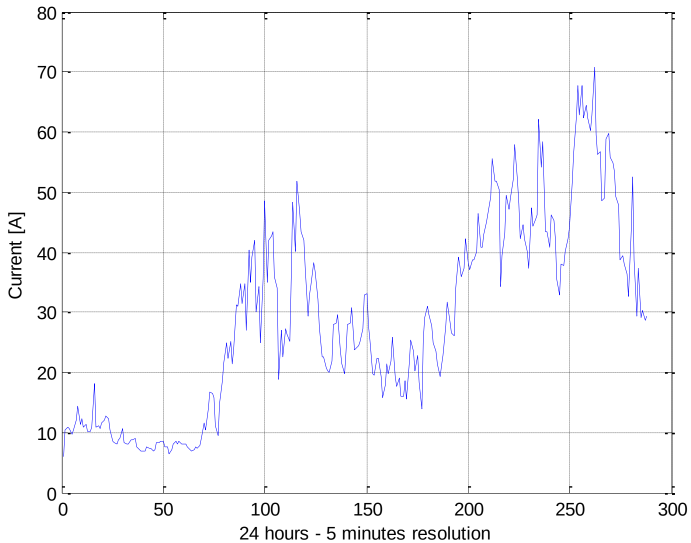
```
%% Initial Data===========================================================
close all;
main_path='H:\2015\OpenDSS_Revamp';
feeder_folder=[main_path,'\Example_Feeder'];                %OpenDSS file location
input_path=[main_path,'\Input_Profiles\'];                  %Data Base Profiles
load_path=[input_path,'Individual_Profiles\'];              %Creation of indicidual profiles

%Data from excel==========================================================
Load_Selection_name=[input_path,'Summer_Load_profiles.xlsx'];   %Location of load profiles
DER_name='Summer_PV_profiles.xlsx';
Load_Base=xlsread(Load_Selection_name);                      %Reading load profiles
DER_f
DER_B
conne
conne
```

```
%Initialization of OpenDSS================================================
DSSObj = actxserver('OpenDSSEngine.DSS');                    %initialization the DSS Object
DSSText = DSSObj.Text;                                       %Define the text interface
DSSText.Command = 'clear';                                   %Clear previous session
DSSText.Command = 'New Circuit.Simple';                      %Definition of the new session
master_file=[feeder_folder,'\Master.txt'];                   %Location of Master OPENDSS file
DSSText.Command = ['Compile ' master_file];                  %Compile the DSS circuit script

%Solve time simulation problem============================================
DSSText.Command = 'Set mode=Daily';                          %Time Simulation
                                                             %Number of step 1min => 24*60/5=288
DSSText.Command='Set stepsize=300s';                         %Flow each 5min*60s = 300s Resolution
DSSText.Command = 'Solve';                                   %Solve the simulation
```

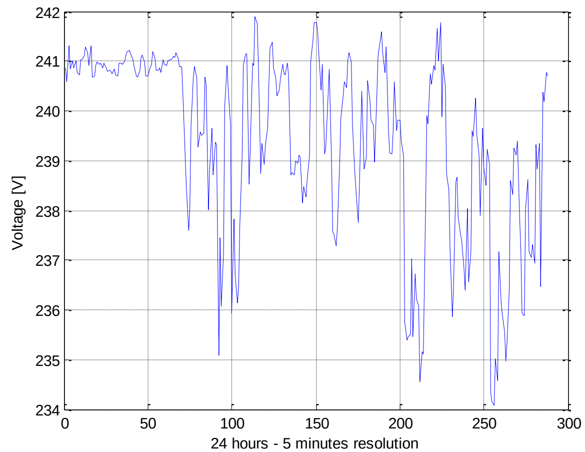
Stochastic Impact Analysis of LCTs: OpenDSS

Time-Series Unbalanced Power Flows





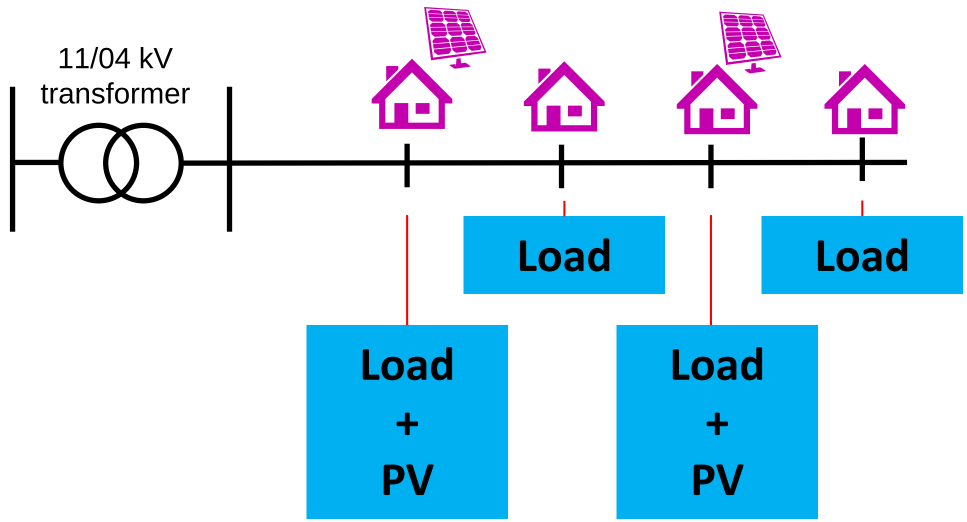
Current at the head of the feeder

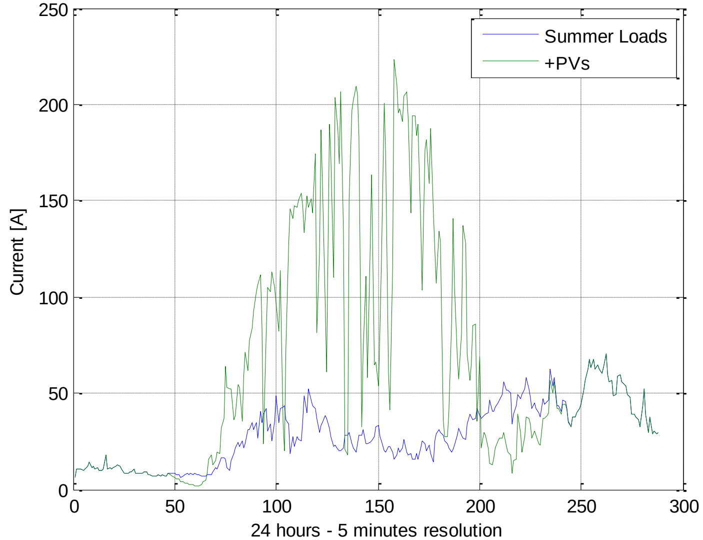


Voltage at the last customer

Stochastic Impact Analysis of LCTs: MATLAB

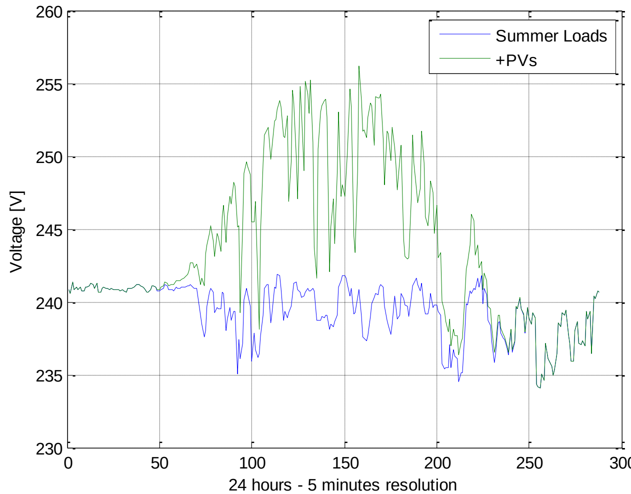
Extract results & Analysis





Current at the head of the feeder

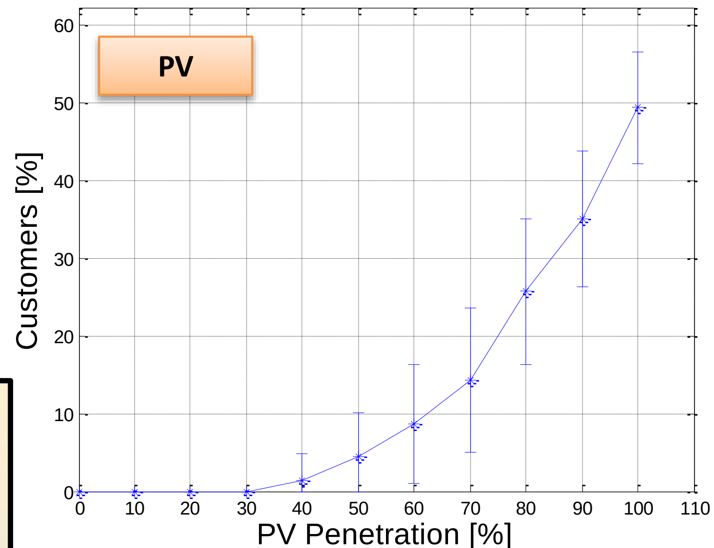
Assessment Visualization

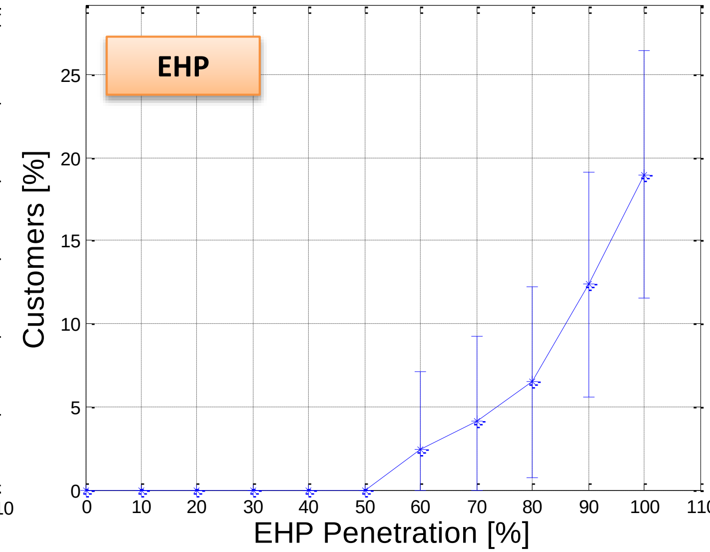


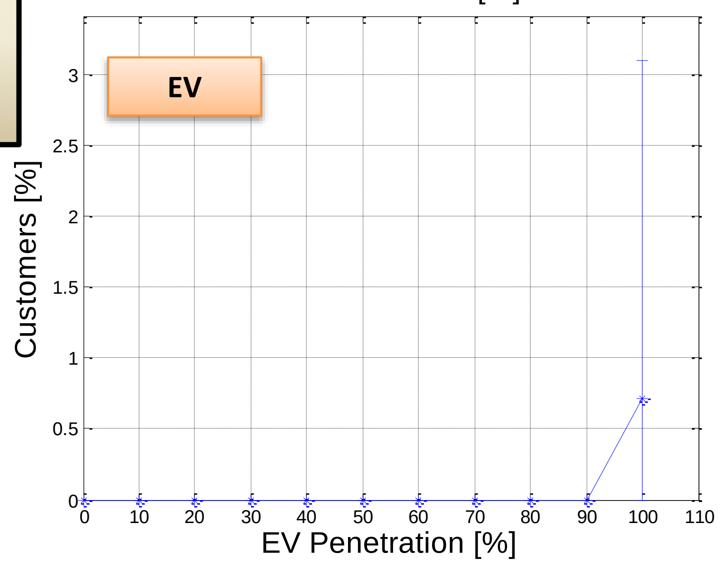
Voltage at the last customer

Metric 1: Voltage Problems

% of Customers with Voltage Problems – BS EN 50160





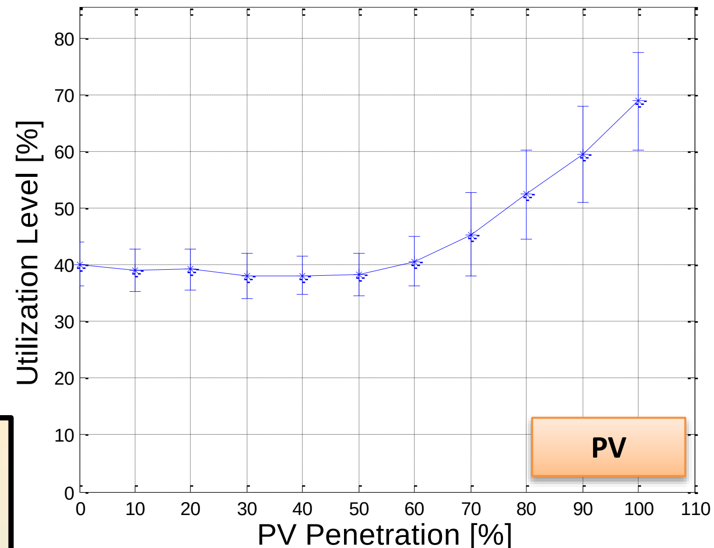


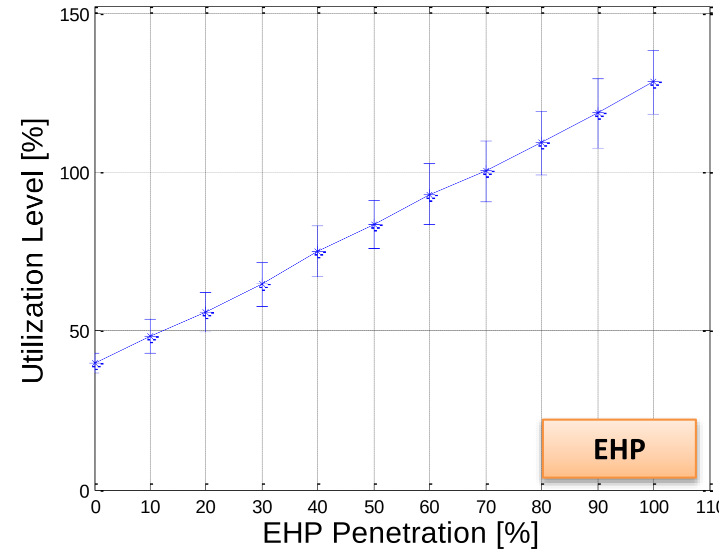
uCHP

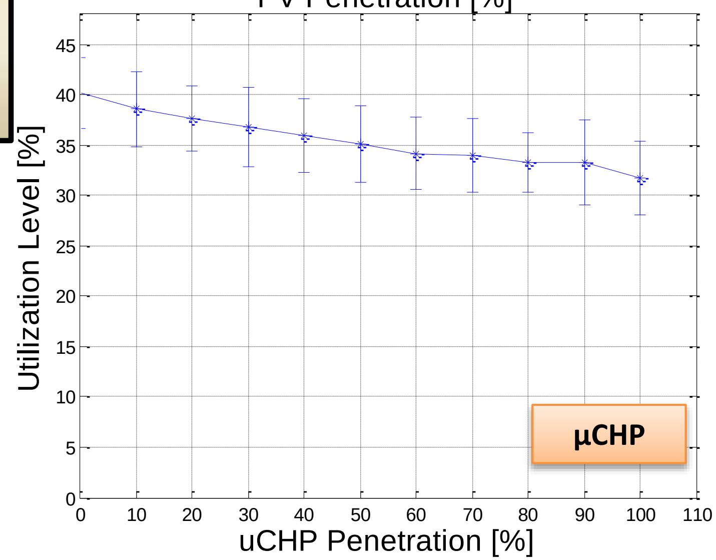
No voltage problems in this feeder with uCHP

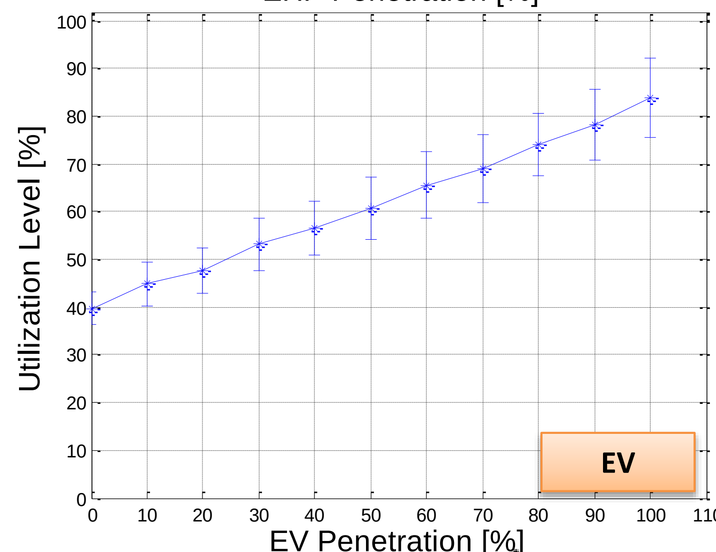
Metric 2: Thermal Problems

Utilization Level of the Head of the Feeder

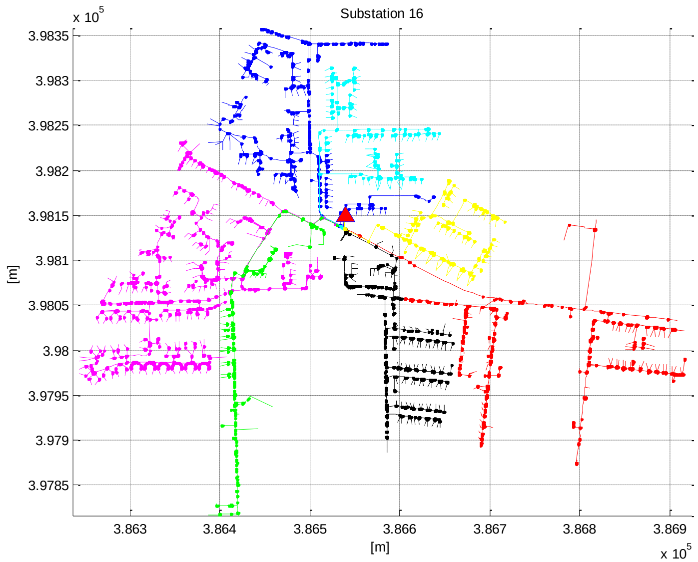


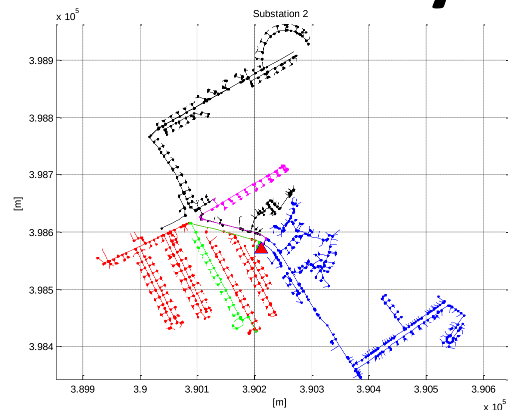


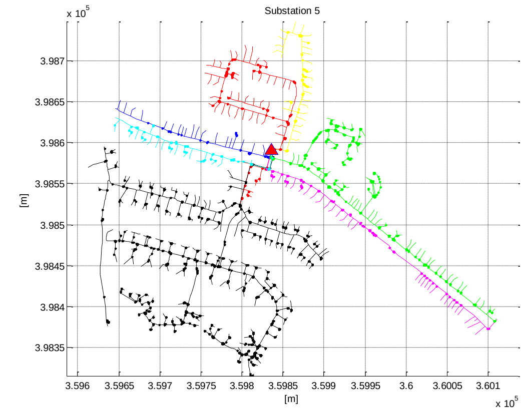




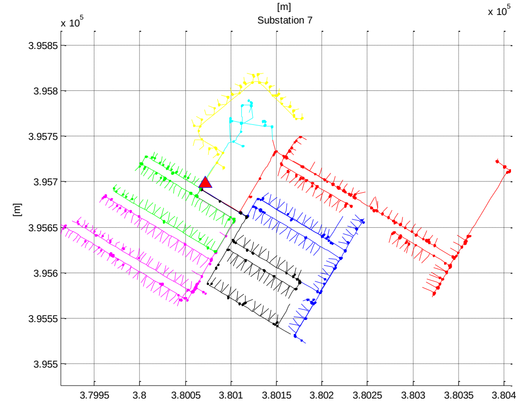
Multi-Feeder Analysis

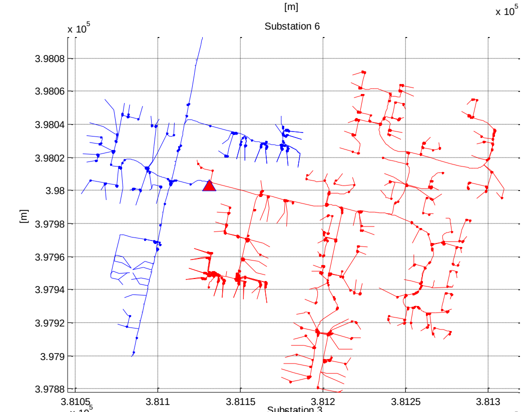


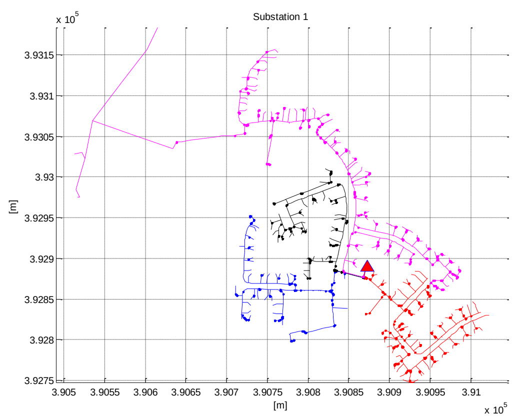


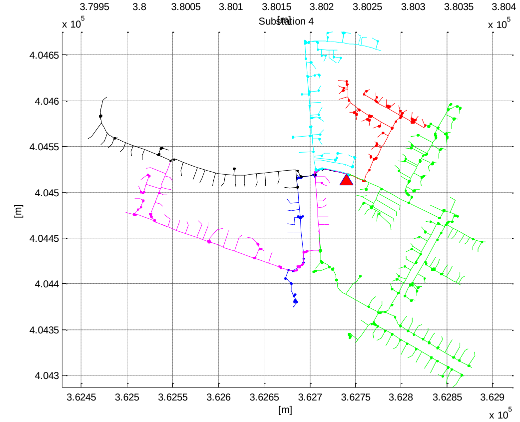


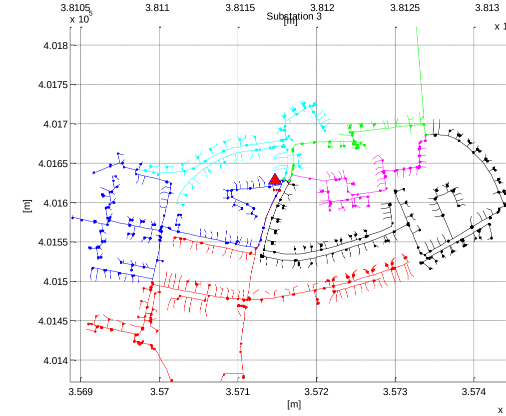
Network Examples





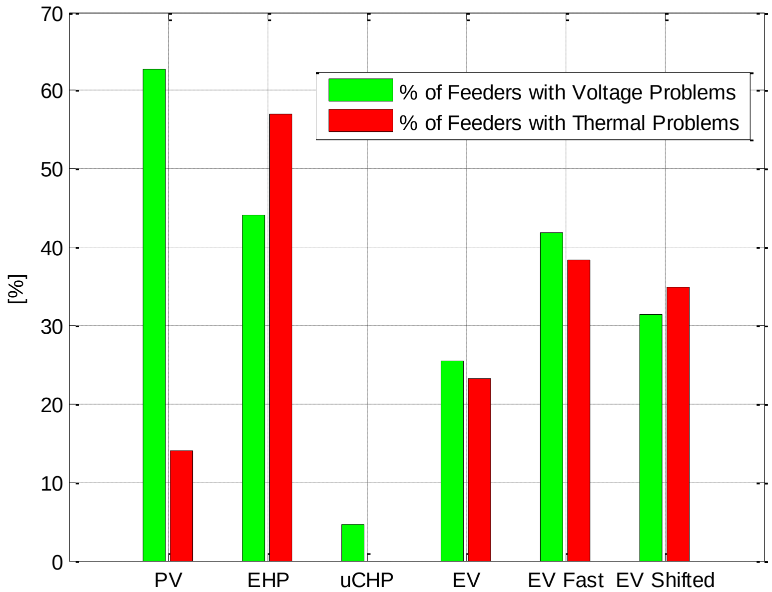




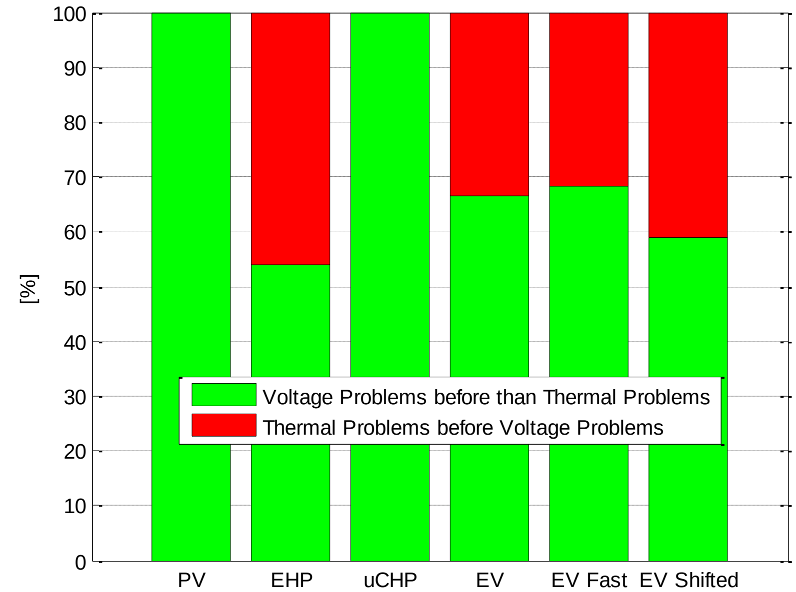


Multi-Feeder Analysis (128)

- Feeders with less than 25 customers (30%) do not present any technical problem for any of the technologies analysed
- Below are the results for the feeders with a technical problem at some penetration level:



% of feeders with problems per technology



% of "Bottleneck" cases per technology

Key Remarks – Stochastic Imp. Analysis of LCTs

- **Uncertainties** of LCT → **Probabilistic** impact assessment
  - Identifying the likelihood of impacts
- **True** understanding of **impacts** → **Realistic models**
  - Networks, demand, LCTs
- **Monitoring** → When, where, what, how often?
  - Who keeps an eye on the data (flagging issues)?
- **Industry needs to adopt this learning**
  - ENWL is now integrating the findings of LVNS into their rules for monitoring LV networks with LCTs

16:45 to 17:15

MANAGEMENT OF ELECTRIC VEHICLE CHARGING POINTS

Dr Jairo Quiros-Tortos (PDRA)

EATL-SSEPD "My Electric Avenue" Project

Outline – Management of EV Charging Points

- My Electric Avenue (MEA) Project
  - EVs in the UK, EV Challenges, Aims of MEA
  - Introduction and Infrastructure of the trials
- EV Charging Behaviour Modelling
- Solution and Customer Effect
  - ESPRIT-based control algorithm
- Remarks

My Electric Avenue (MEA)

Total Cost
US$15m

MY ELECTRIC AVENUE

Project run by
EA Technology
Funding DNO
SSE PD

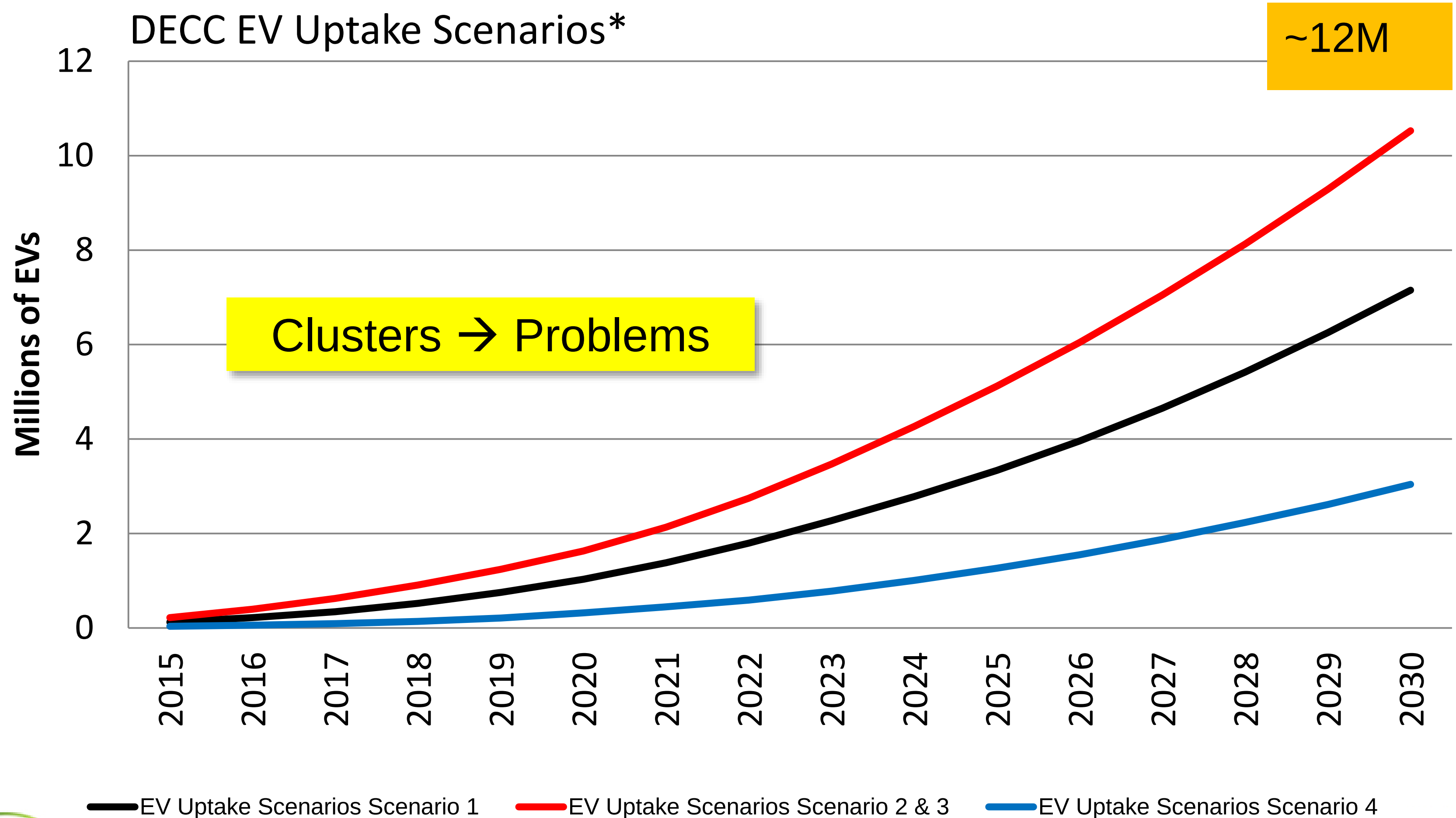
ea technology
Scottish and Southern Energy Power Distribution

myelectricavenue.info

IEEE PES Power & Energy Society®

IEEE

Electric Vehicles (EVs) in the UK



* Department of Energy and Climate Change (DECC)
https://www.ofgem.gov.uk/ofgem-publications/56824/ws3-ph2-report.pdf

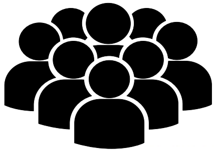
EV Challenges

- EV Clusters
  - Can affect the infrastructure close to customers (LV networks)
  - Thermal overloads, voltage drops

MY ELECTRIC AVENUE

Control of EV Charging Points

- EV Management
  - Cost-effective infrastructure
  - Fair criteria to control EVs
  - Customer acceptance



IEEE PES Power & Energy Society

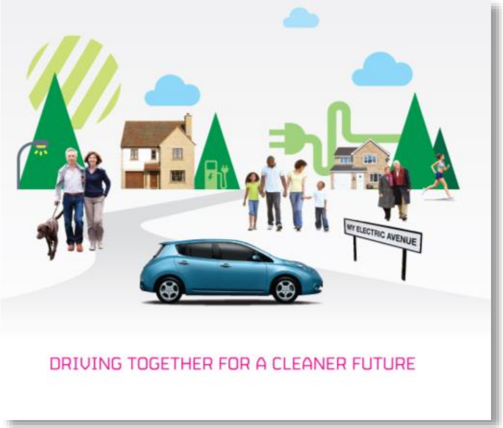
IEEE

My Electric Avenue (MEA)

Aims

- To understand charging behaviour of (200+) EV users
- To investigate the impacts of EVs on 9 real LV networks
- To trial a cost-effective and practical solution to control EV charging points (ESPRIT Technology*)

MY ELECTRIC AVENUE



* Patent app. no. PCT/GB2012/050702

Geographical Extent of the Trial

112 Social Trials

109 Technical Trials

<u>221 in total</u>

Infrastructure Overview

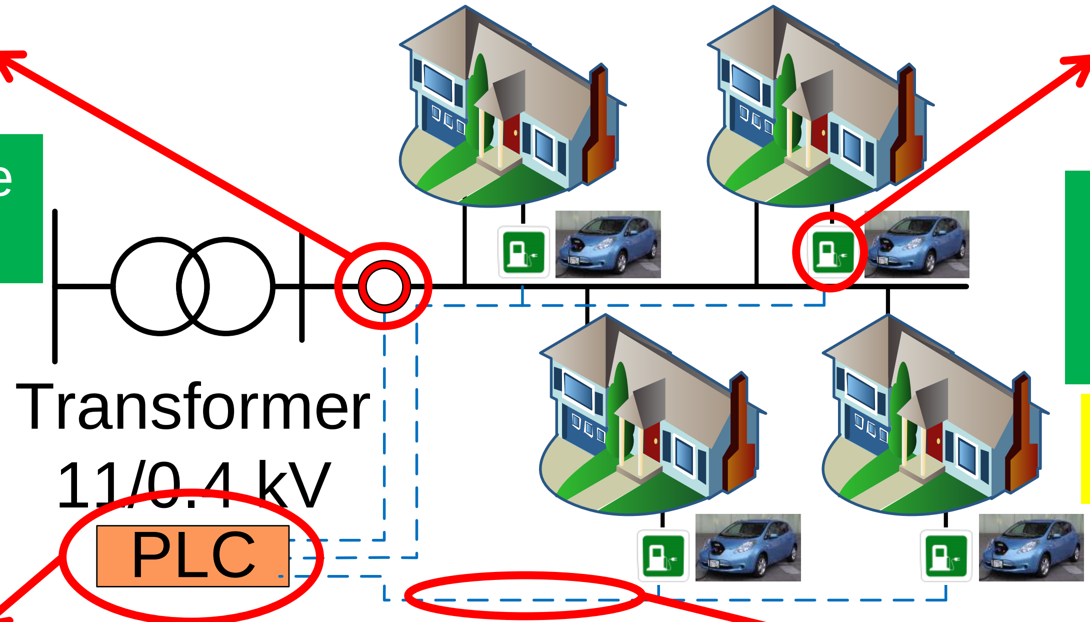
Sensors (V, I) head of feeders

Violations in the thermal limits

Sensors and actuators at EV charging points

Significant voltage drops

State of Charge: Unknown

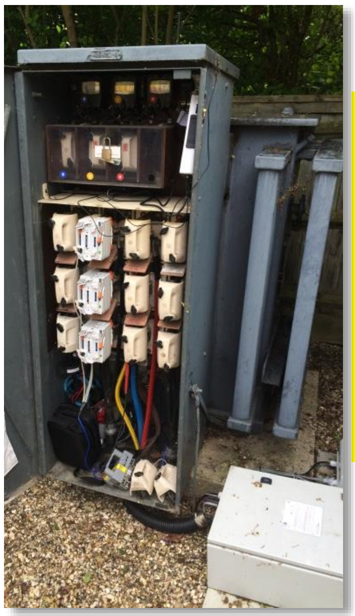


PLC-like device at substations **(control hub)**

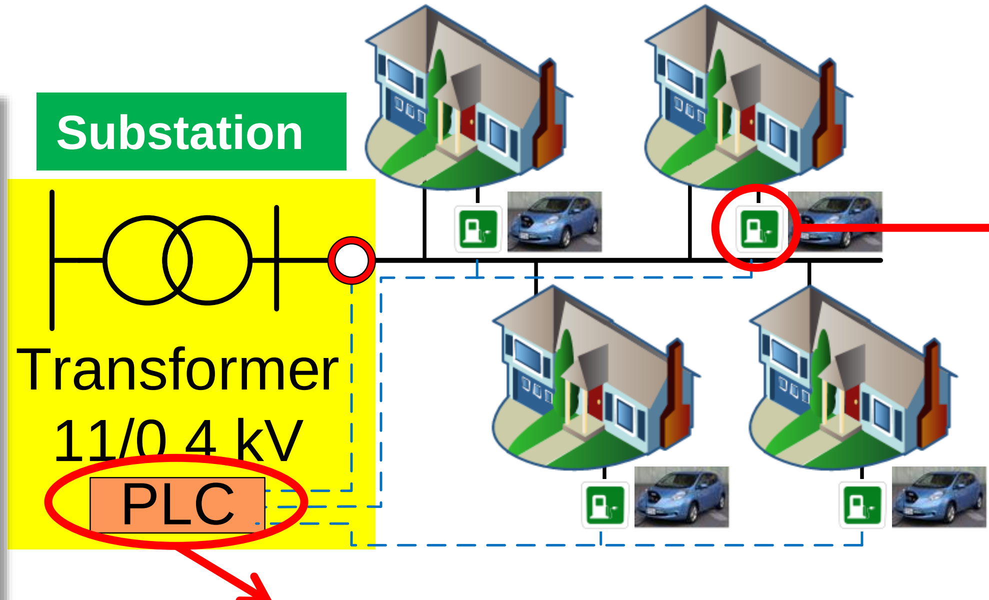
Power Line Carrier-based communications (**bi-directional**)

MEA makes the most of available infrastructure

Infrastructure Overview



Real 500 kVA Transformer

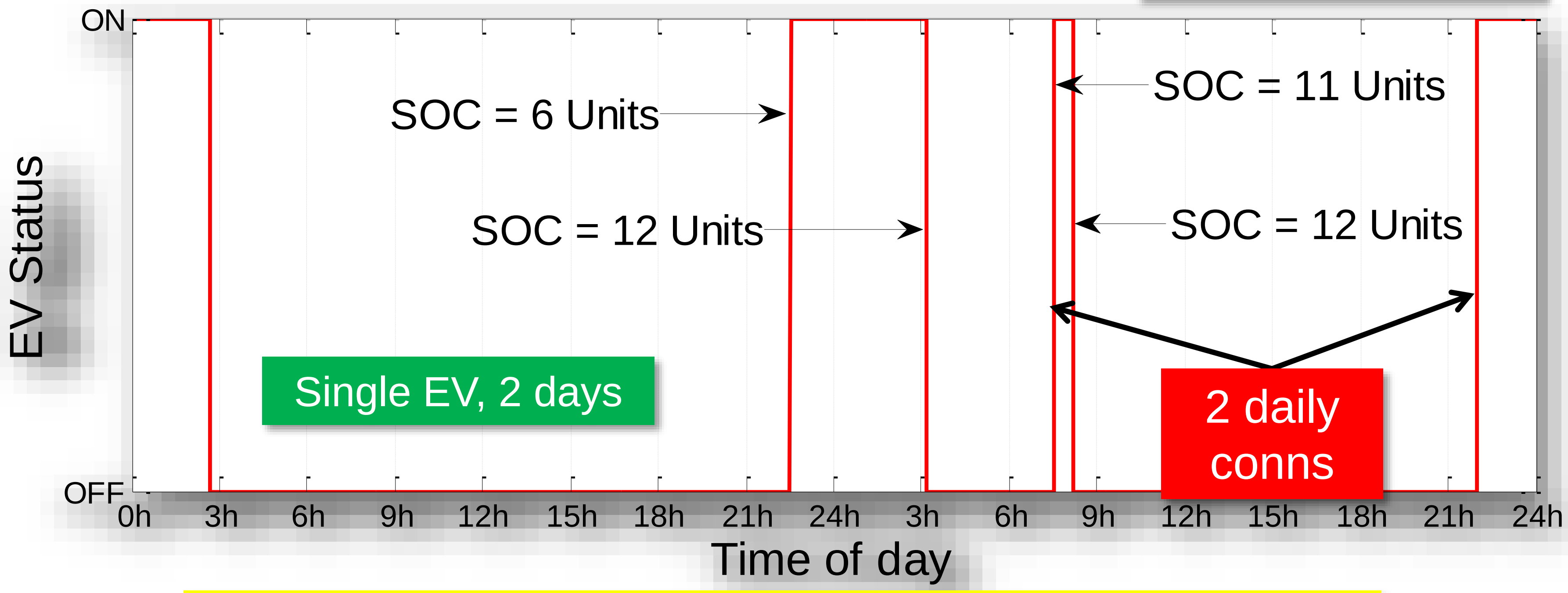


ROLEC* charging point
+
EA Technology **Intelligent Control Box**

* http://www.rolecserv.com/

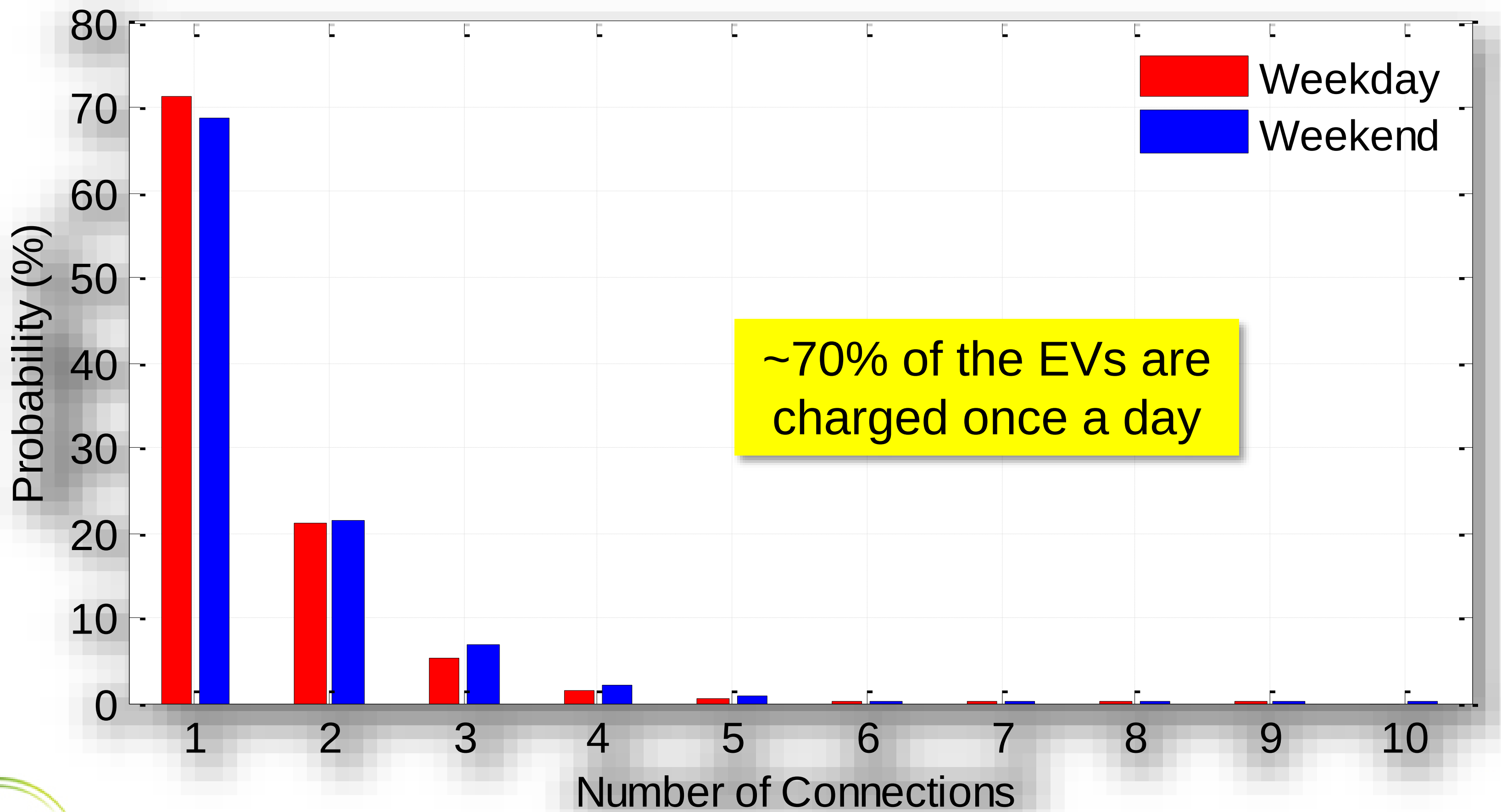
EV Charging Behaviour

More than 75,000 charging samples
(without control)

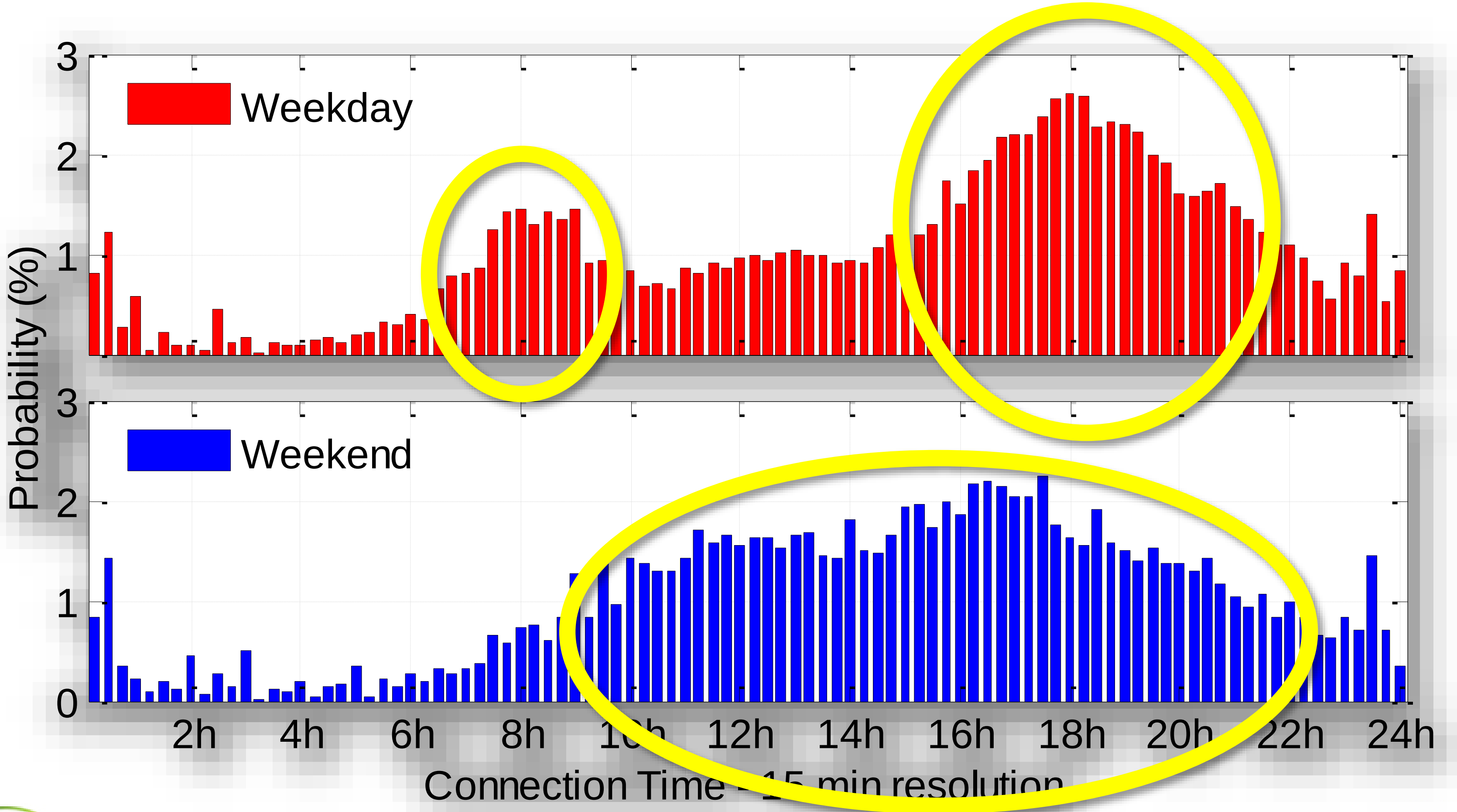


Crucial to understand EV users charging behaviour

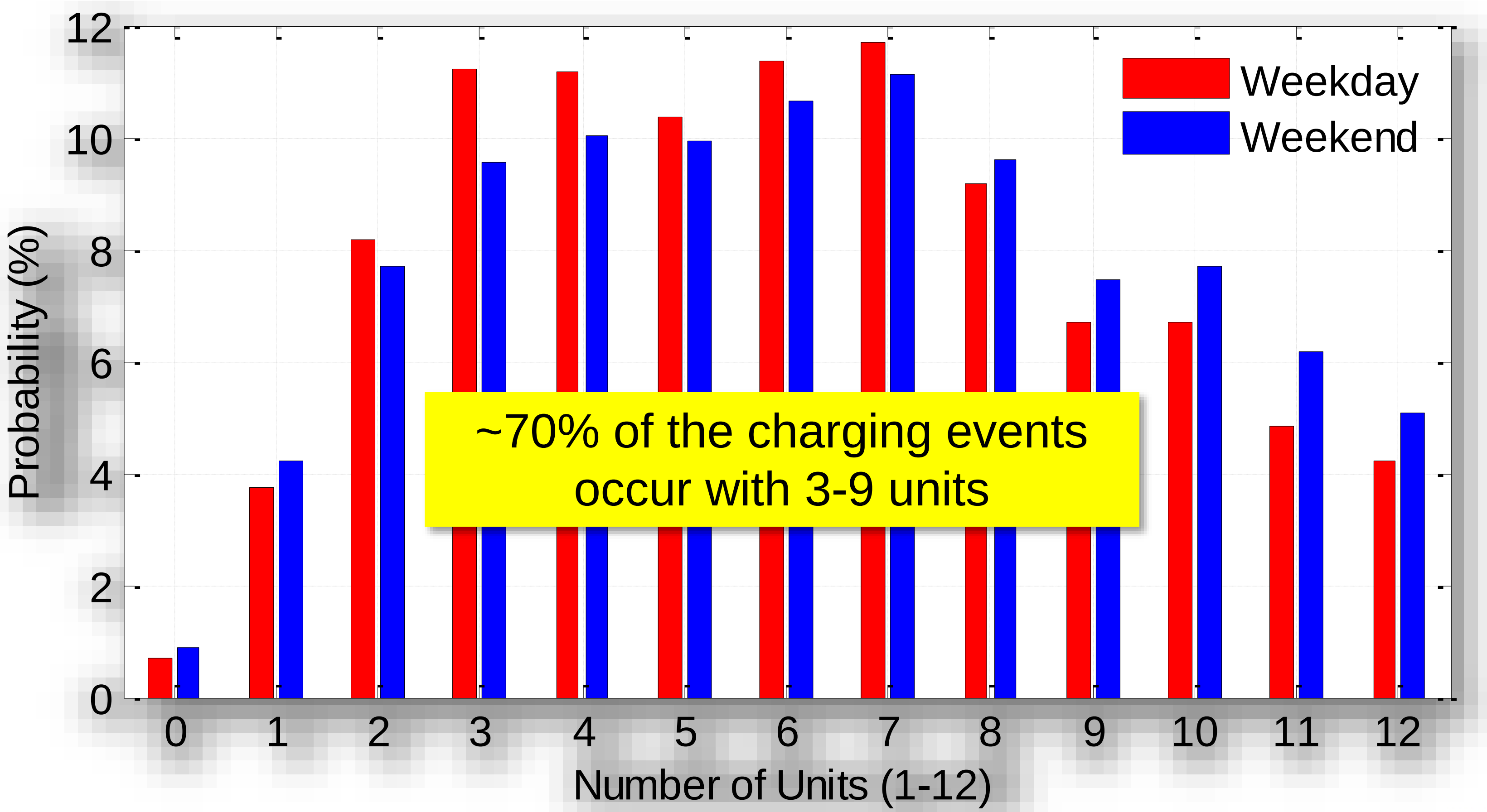
Number of Connections per Day



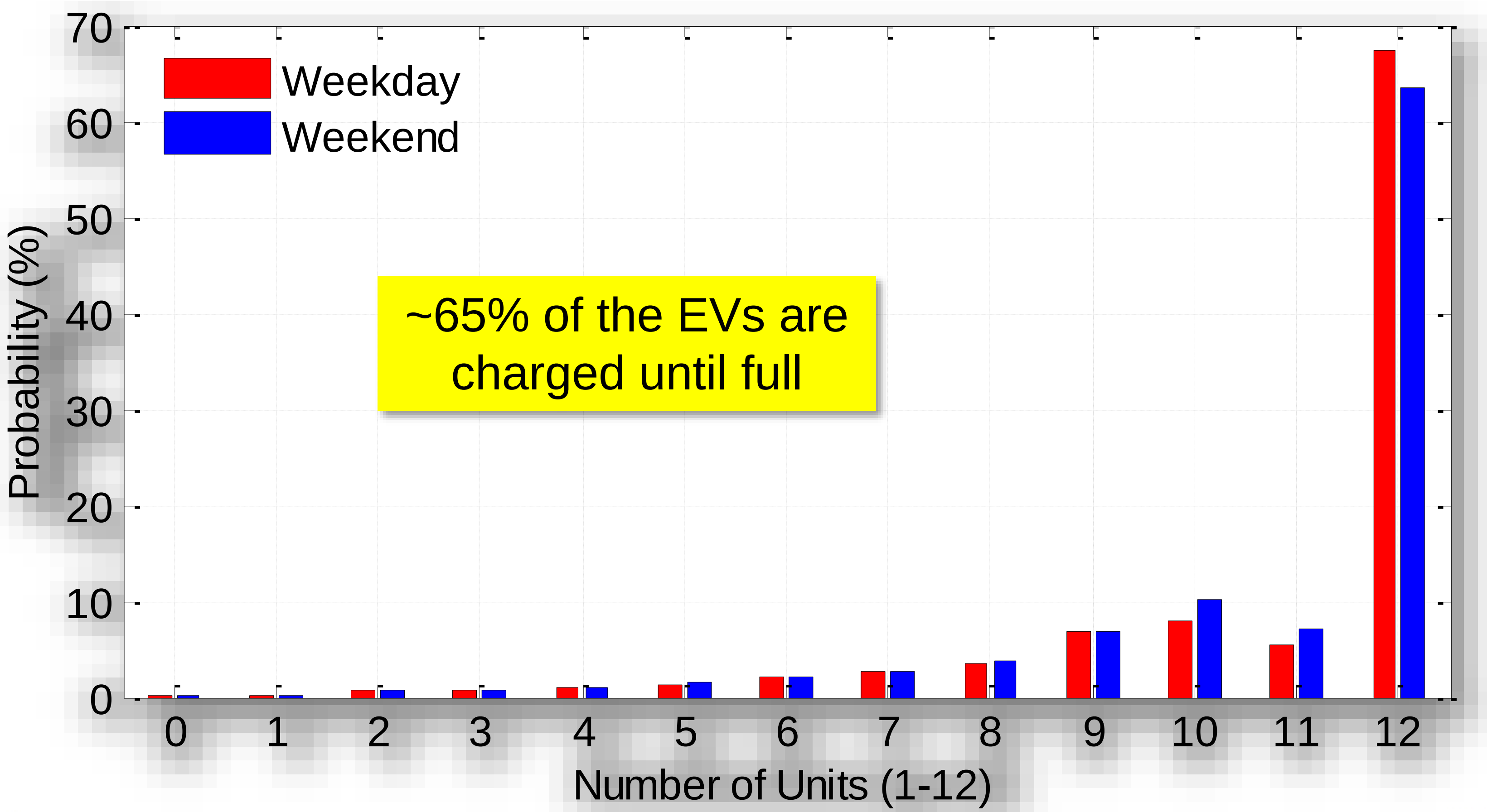
Start Charging Time



Initial Charging Level

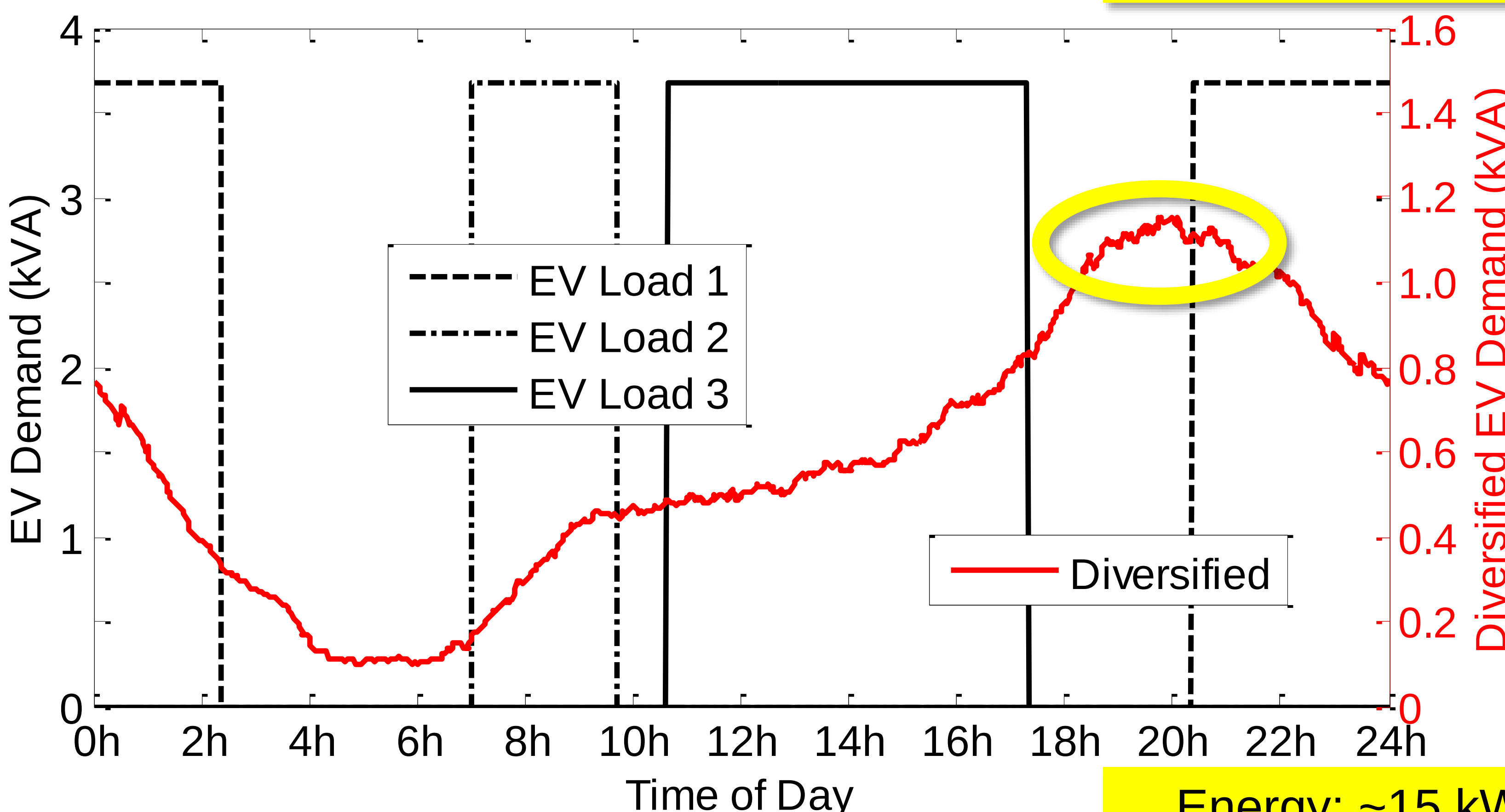


Final Charging Level



From Statistical Analysis to Realistic EV Models

EV Demand: ~1.2kW

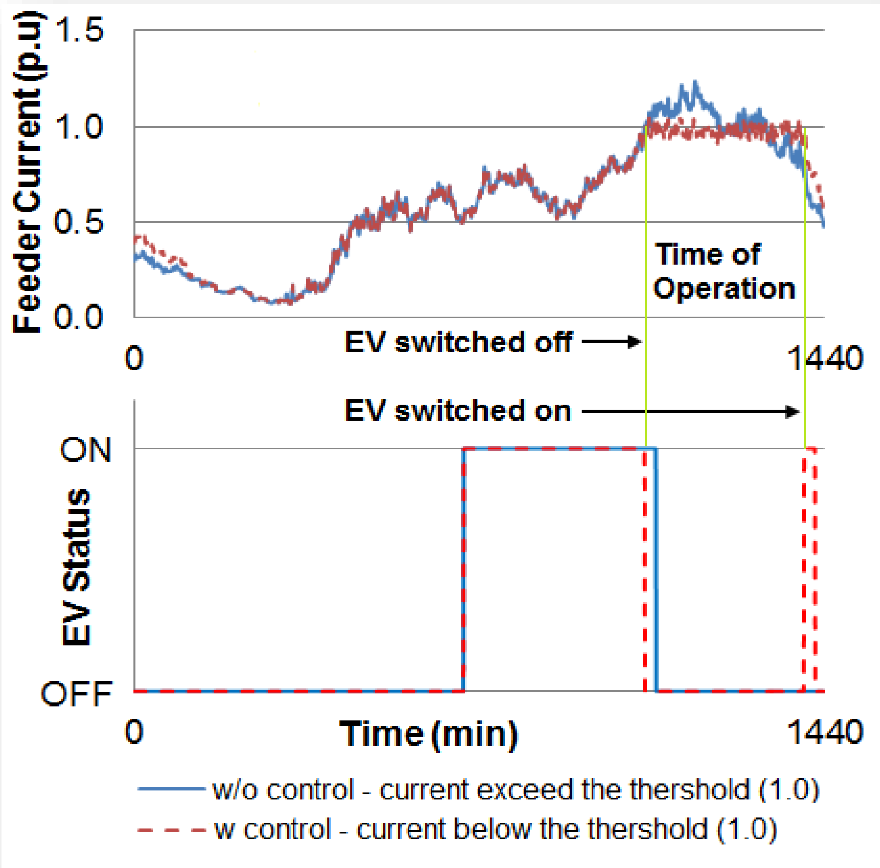


Energy: ~15 kWh

ESPRIT-Based Control: Conceptual approach

1. Disconnect loads when problems are detected

2. Reconnect loads when no problems are detected (considering security margins)



ESPRIT-Based Control: Design Challenges

- Hierarchical (corrective) disconnection

Feeder Level (per phase per feeder) → Transformer Level

- Hierarchical (preventive) reconnection

Transformer Level → Feeder Level (per phase per feeder)

- The number and which EVs will be managed
- Effects on customers – charging delays

More details: J. Quirós-Tortós, L. F. Ochoa, S. Alnaser, and T. Butler, "Control of EV charging points for thermal and voltage management of LV networks," IEEE Transactions on Power Systems, In Press

How is this EV management actually done?

MATLAB

- OpenDSS driver
- Random variables creator
- **Control Implementation**
- Results Analyser

COM

1-Input Data acquisition

2-Profiles Random Allocation

3-Power flow Simulation

Adopting the control algorithm

4-Result Visualization

Simple ☺

DSS

Time-Series, three-phase power flow solver

*Can also be done using VBA

EV management : MATLAB + OpenDSS

Initialization (Similar to the Impact Assessment)

MATLAB ⟷ COM ⟷ DSS

Random assignation of variables

Time-Series Unbalanced

Control

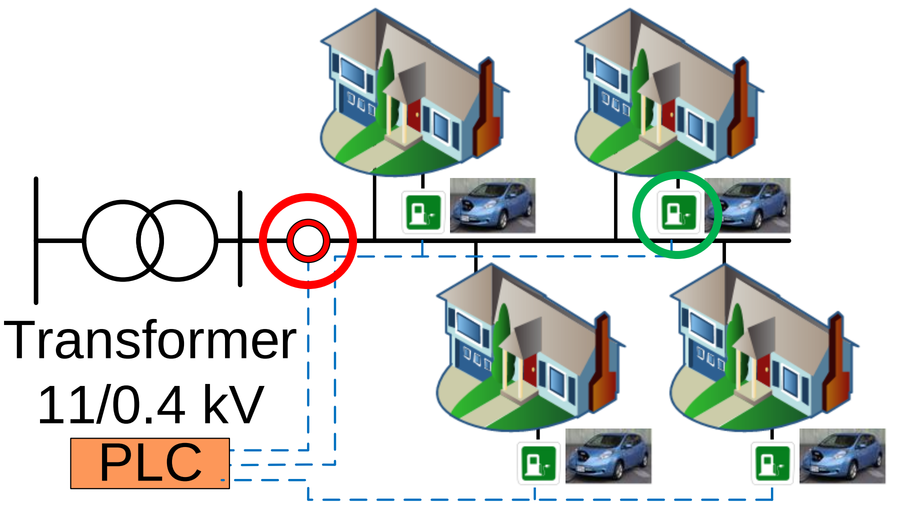
MATLAB ⟷ COM ⟷ DSS

Calculations Control settings ⟷ Every control cycle ⟷ Time-Series Unbalanced

MATLAB is finally used to extract monitors and assess, visualise the results

EV management : MATLAB + OpenDSS

- Data collection
  - ✓ Phase current (head of feeder)
  - ✓ Busbar phase voltages
  - ✓ Charging point phase voltages



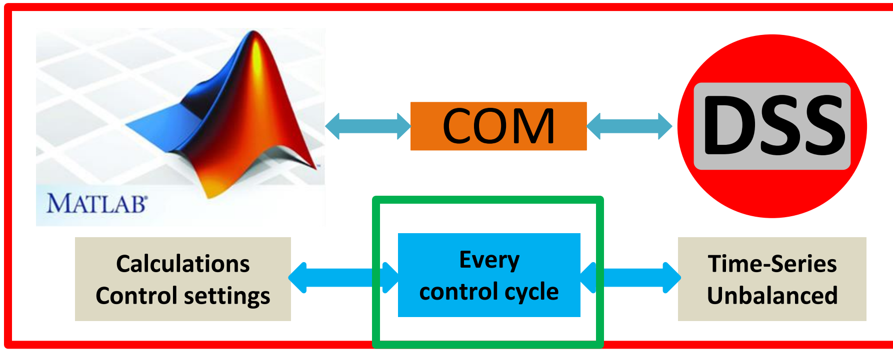
Charging time is updated for each EV
*To select EVs to be managed
**SOC is unknown

Hierarchical (corrective) disconnection

- Disconnection (*Feeder*)
  - ✓ Required to mitigate overload
  - ✓ Number of charging points with voltages below the limit
  - ✓ The maximum of both
- Disconnection (*TX*)
  - ✓ Required to mitigate overload

Problems?
Y

Which ones?



EV management : MATLAB + OpenDSS

EV1 will be disconnected fist

✓ Its charging time is longer than EV2

The longer the charging time, the more likely it is to be disconnected

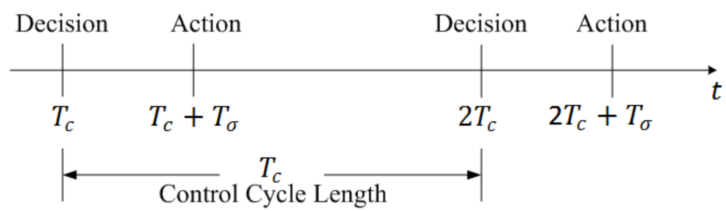
Feeder Level: Each phase is treat independently

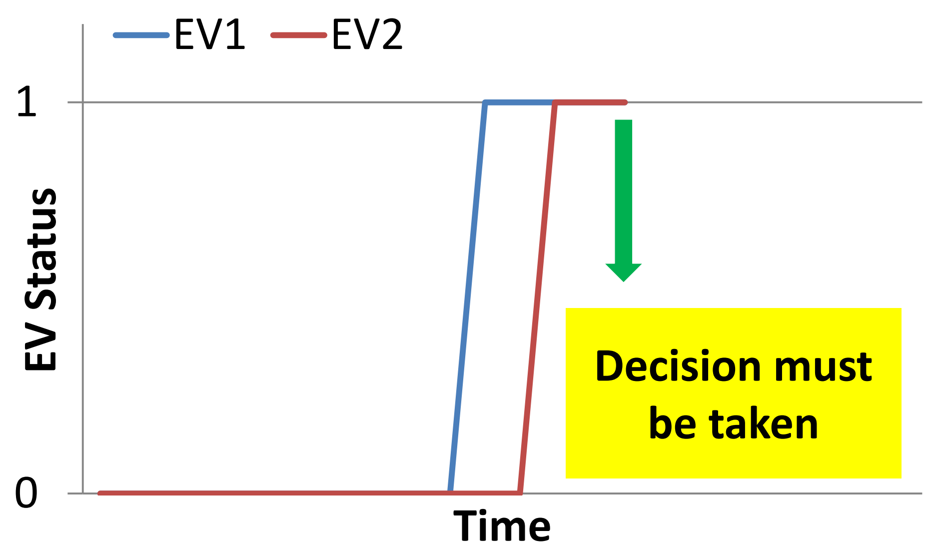
✓ EVs in a phase without problems will not be affected

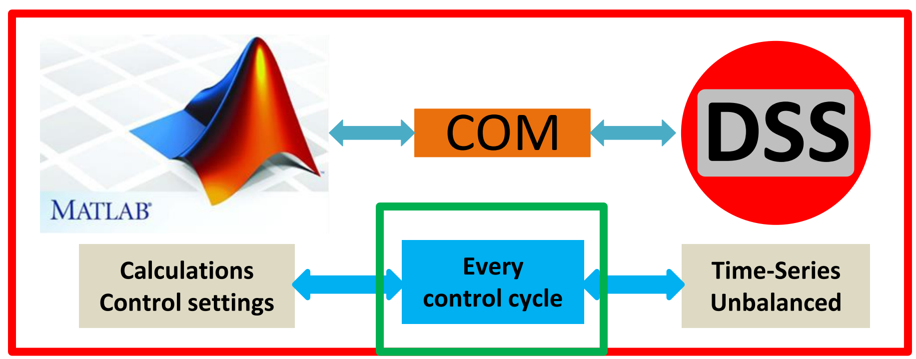
TX Level: Three-phase analysis

✓ Every EV may be disconnected

Problems are fairly shared







EV management : MATLAB + OpenDSS

Hierarchical (preventive) reconnection

- Reconnection (*TX*)
  - ✓ Max # to keep *TX* loading below a security margin

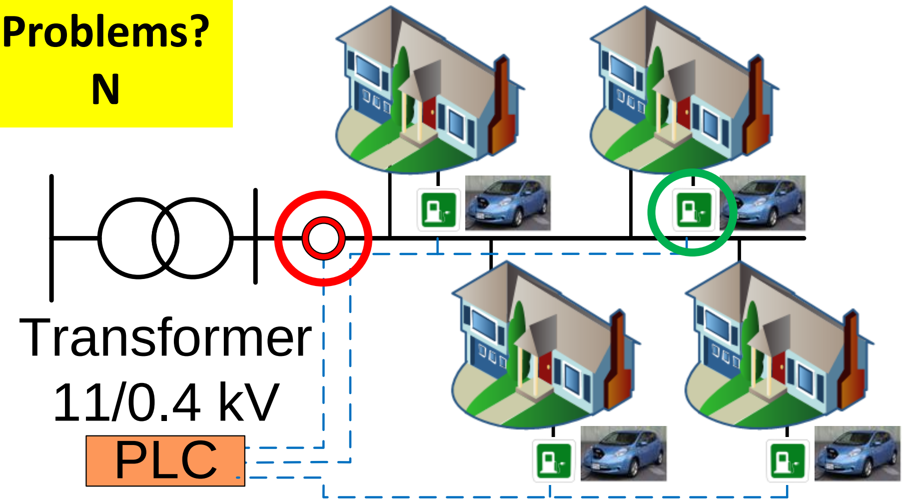
Which ones?

The shorter the charging time, the more likely it is to be reconnected

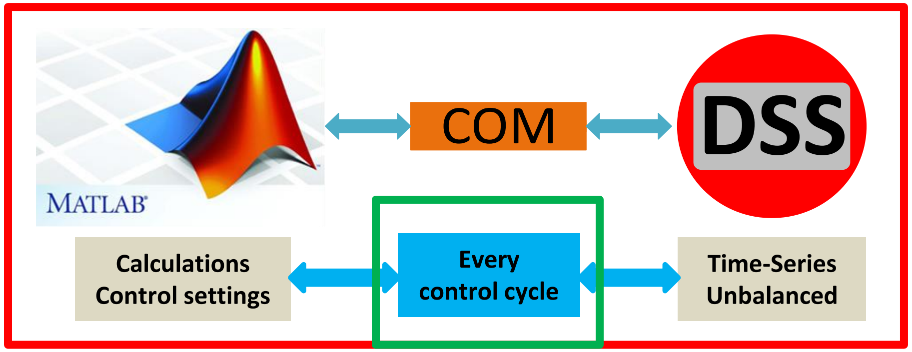
An EV will be reconnected if its reconnection does not violate feeder constraints

- Reconnection (*Feeder*)
  - ✓ If phase current after reconnection will be below a security margin
  - ✓ If charging point phase voltages higher than a security margin

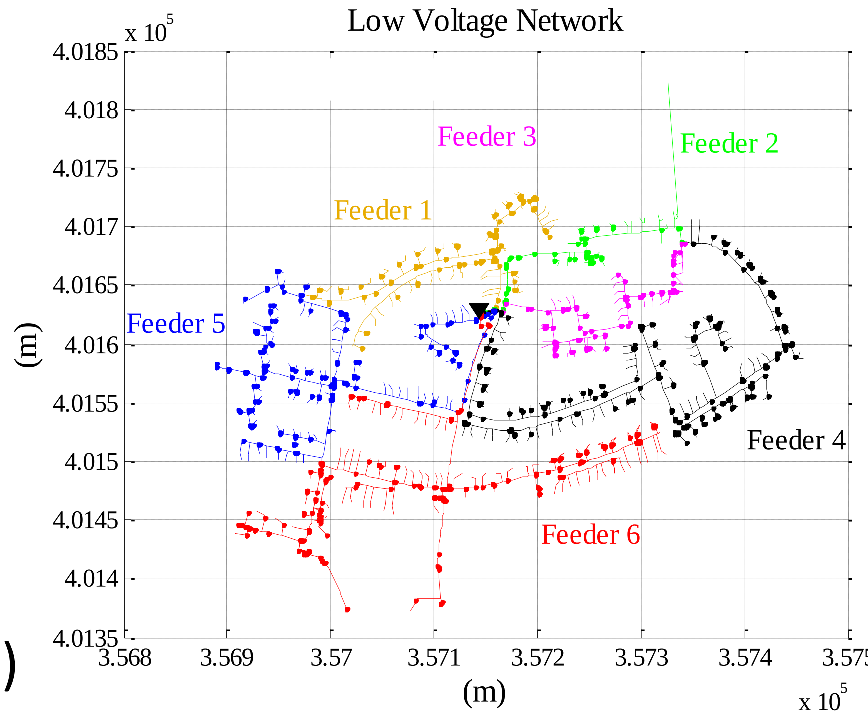
Problems?
N



Available capacity is given to EVs with lowest charging time



ESPRIT-Based Control: Assessment



Inputs

- Real LV **networks**
- Realistic **domestic*** and **EV** load **profiles**

Probabilistic Assessment

- **Monte Carlo** approach (uncertainty)
- **Time-series** analysis (unbalanced)
- **Metrics**
  - ✓ Thermal overloads
  - ✓ Voltage issues (BS EN 50160)

<u>9 Real UK LV Networks</u>

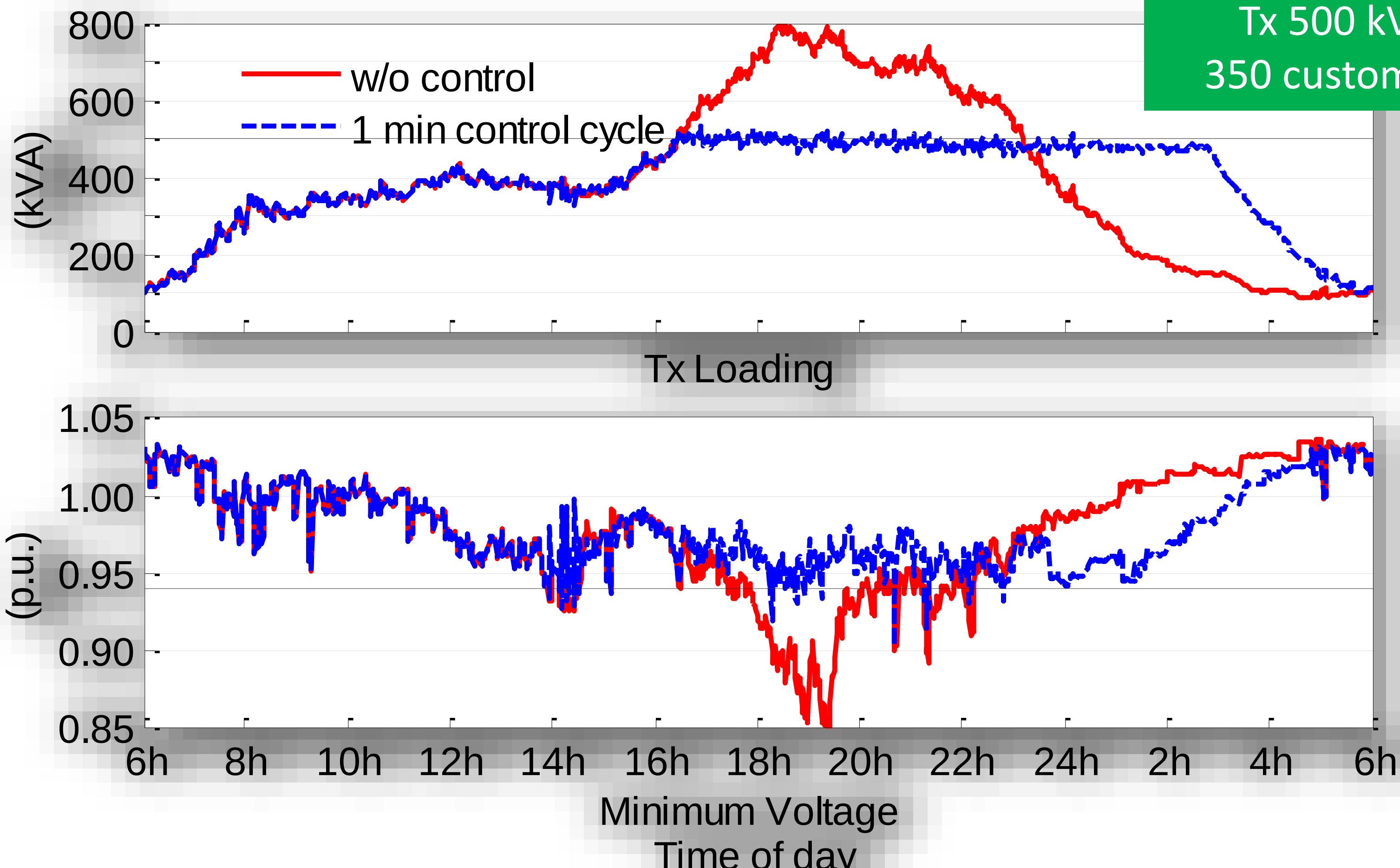
- 11kV/433V, three-phase
- Single-phase customers
- 31 LV feeders
- Main cable: 220–750m
- 2,000+ customers

* CREST Tool

More details: https://www.youtube.com/watch?v=Ox2bQ4vpLNg

Network Performance (100% EVs)

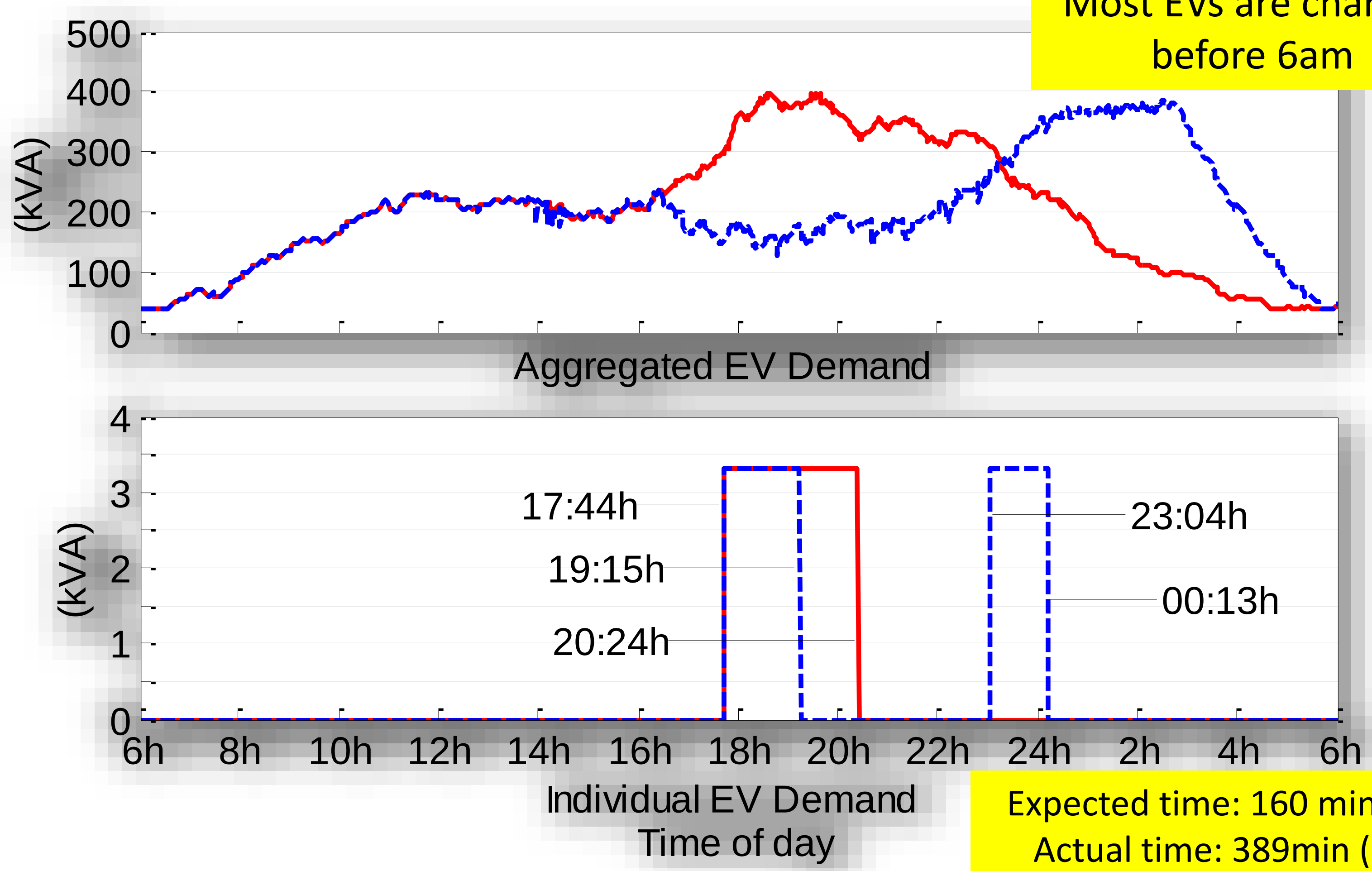
Tx 500 kVA
350 customers



1-min control cycle → Problems solved! (<u>in theory</u>)

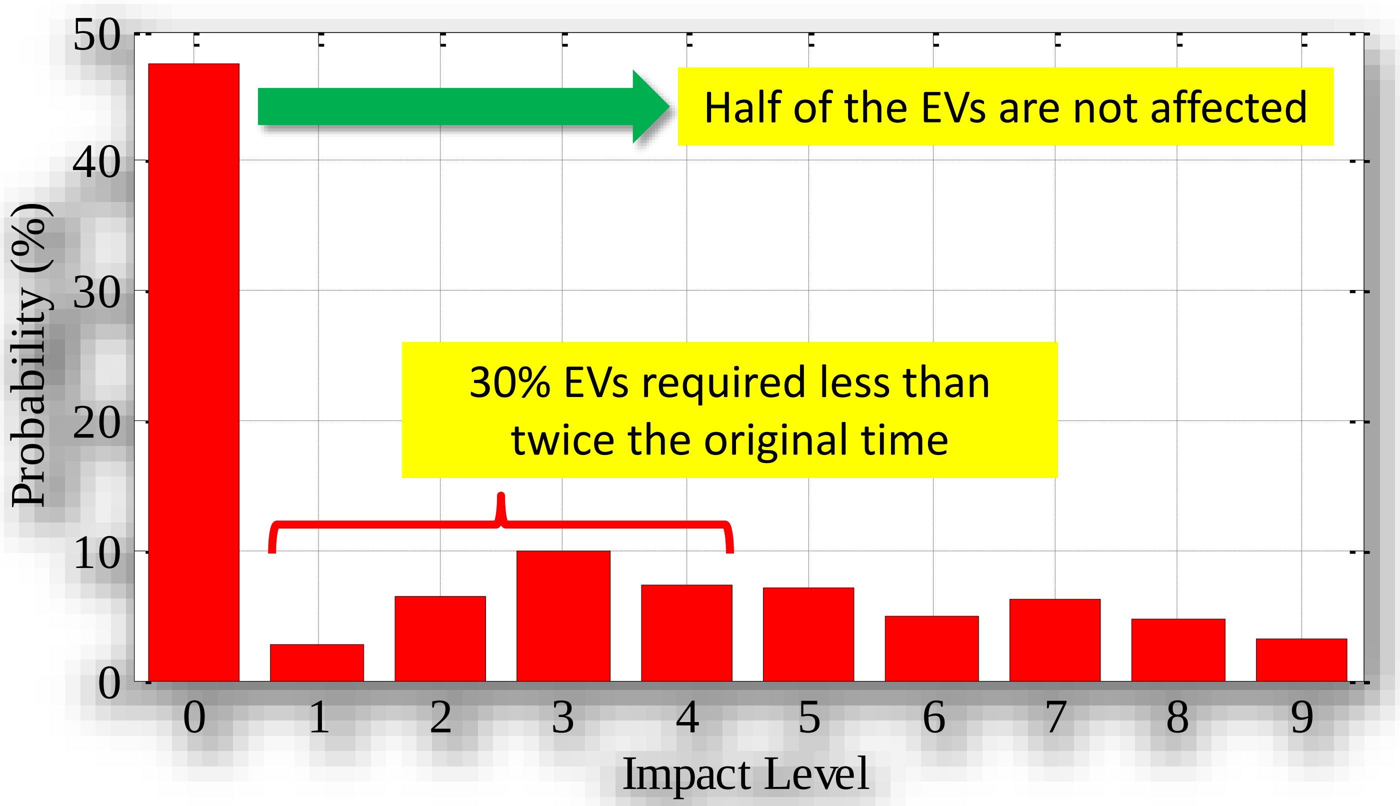
Effects on EV Demand

Most EVs are charged before 6am



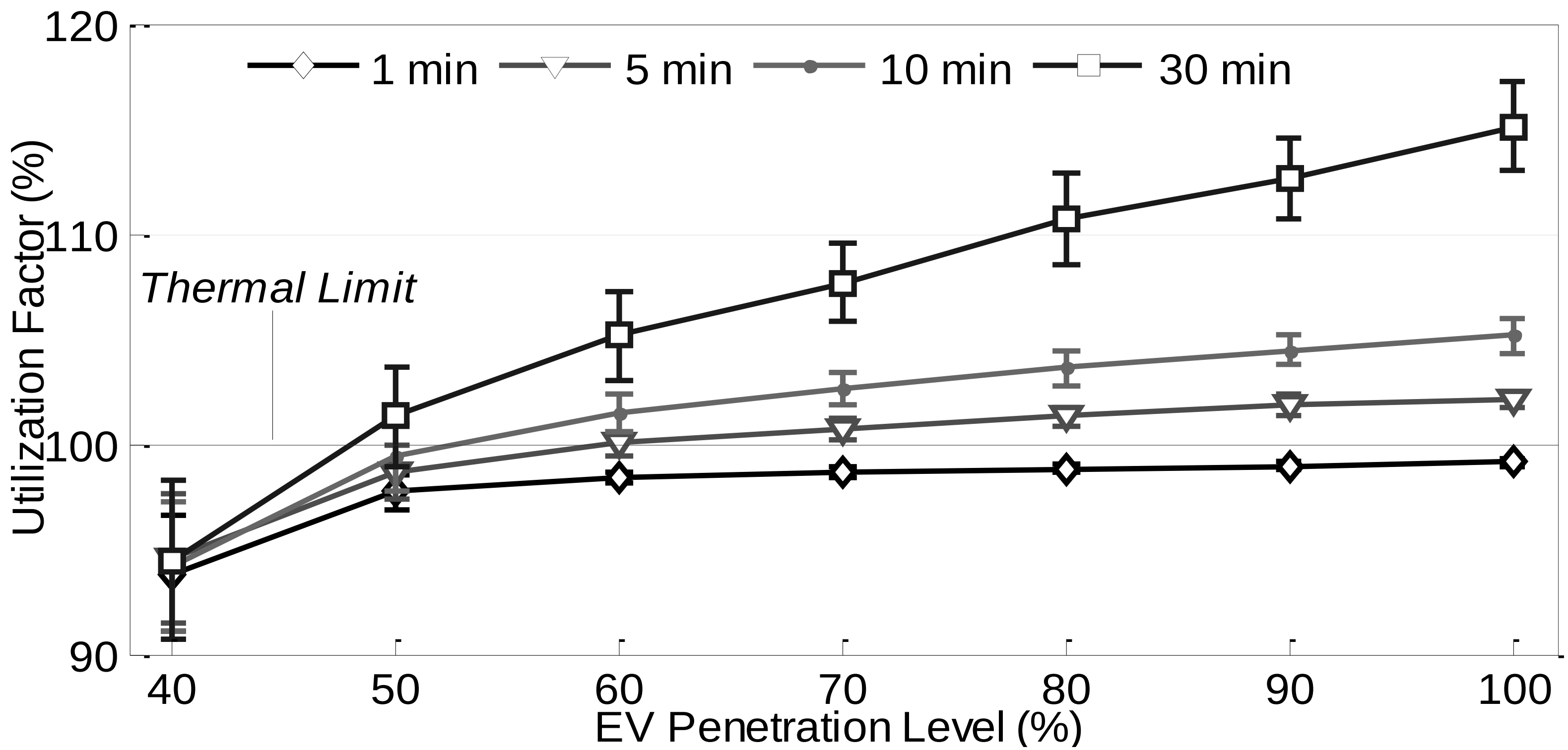
Expected time: 160 min(2:40h)
Actual time: 389min (6:29h)
Charging Delay: 143.13%

Customer Impact Level (CIL)



Customer Impact Level	**0**	**1**	**2**	**3**	**4**
Additional Charging Time (%)	0	1-25	26-50	51-75	76-100
Customer Impact Level	**5**	**6**	**7**	**8**	**9**
Additional Charging Time (%)	101-125	126-150	151-175	176-200	> 200

Probabilistic Assessment: Tx Loading



10-min control cycle can be as effective

Probabilistic Assessment: CIL

Percentage of EV users
w/o delay
CIL = 0

Control Cycle	EV Penetration Level (%)						
	40%	50%	60%	70%	80%	90%	100%
1 min	99	87	72	59	50	44	**40**
5 min	99	89	77	67	60	56	**51**
10 min	100	91	80	71	63	59	**54**
30 min	100	95	85	76	70	64	59

and it improves customer acceptance

Key Remarks – Management of EVs

- **Trials** are crucial to **capture** the actual EV behaviour and customer acceptance
  - Significant changes from weekday to weekend but little across seasons
  - 30% of EV users charge more than once a day
  - Customers are willing to have EVs being managed

- **ESPRIT-Based EV Management**
  - Actual trial proves the infrastructure works
  - Studies also show it can solve network problems

10 min

17:25 to 17:45

INTEGRATED OPTIMAL VOLT-VAR CONTROL OF MV AND LV NETWORKS

Mr Luis Gutierrez (PhD Student)

ENWL "Smart Street" Project

Outline – Optimal Control of MV/LV Networks

- Smart Street Project
  - Geographical extent, stats, scope
- LV and MV Active Voltage Control
  - Normal Operation → Optimal Operation
- Energy Reduction (CVR)
- Modelling Real Network Control
- Optimal Control (Examples)
- Remarks

Smart Street (aka eta)

Total Cost
US$13.1m

SMART STREET

Project run by
Electricity North West Limited

electricity north west

http://www.enwl.co.uk/smartstreet

Smart Street (aka eta)

- 6 Primary Substations
  - 11 MV feeders
  - 7 MV capacitors
- 38 Secondary Substations
  - 163 LV feeders
  - 84 LV capacitors
  - 5 **LV OLTCs**
  - 80x3 **LYNXs**
  - 163x3 **WEEZAPs**
- **~67.500 customers**

First fully centralised MV/LV network management and automation system in GB

http://www.kelvatek.com/

Smart Street (aka eta)

Scope

- Innovative **reconfiguration** (including meshing) of **MV** and **LV** networks to increase the hosting capacity of LCTs (PV, EVs, EHPs).

6 primary substations
11 HV circuits

38 distribution substations
163 LV circuits

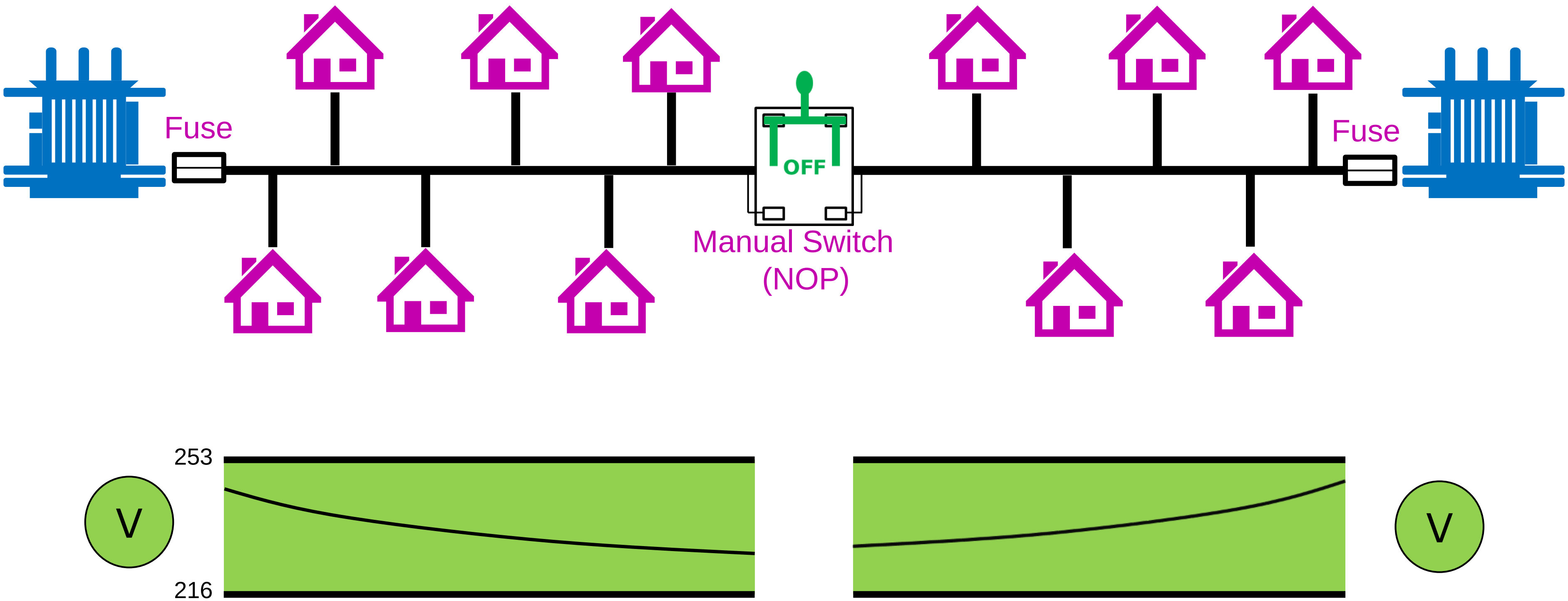
Around 62,000 customers

3 selected primary substations in CLASS

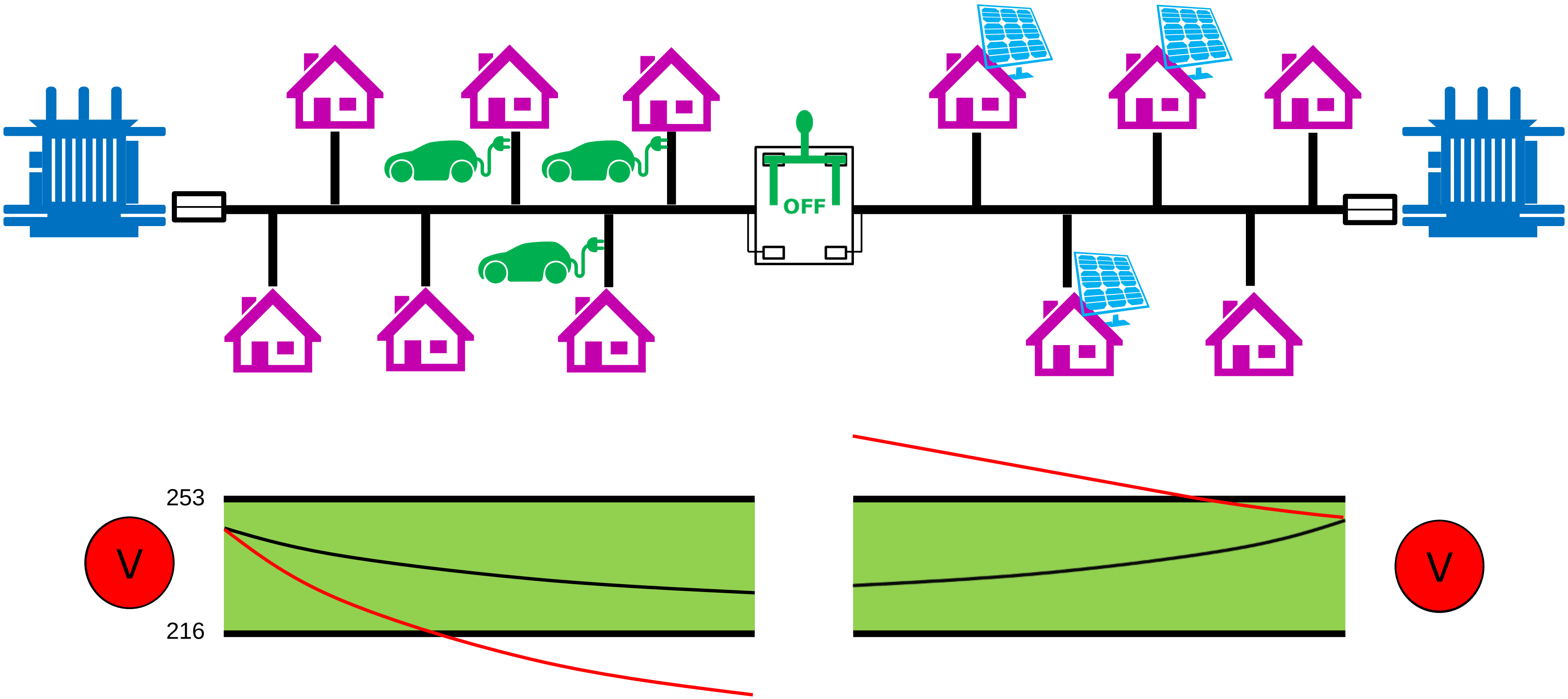
- Optimal **management** of **MV** and **LV** voltages to provide flat profiles at customer connection points throughout the day → **energy reduction**

SMART STREET

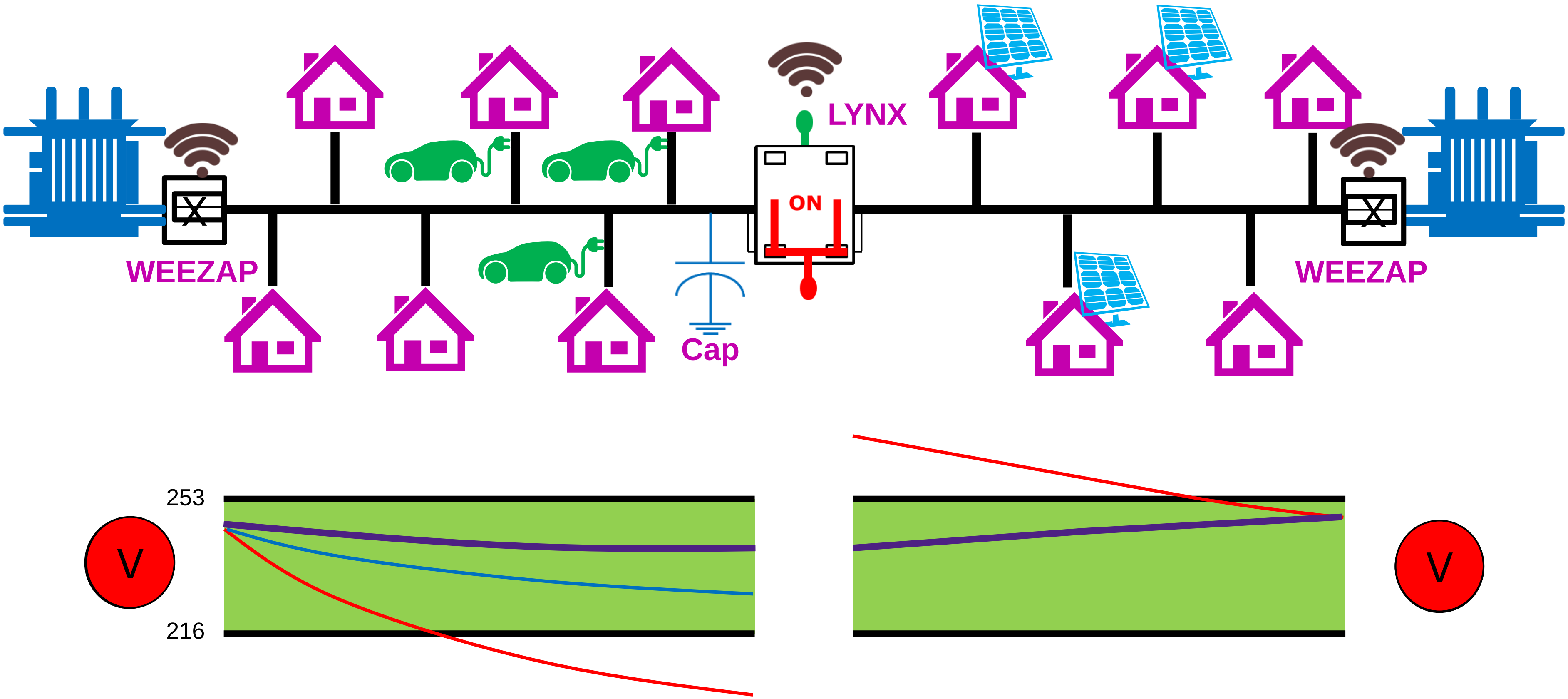
Normal Operation of LV Networks



Effects of LCTs in LV Networks



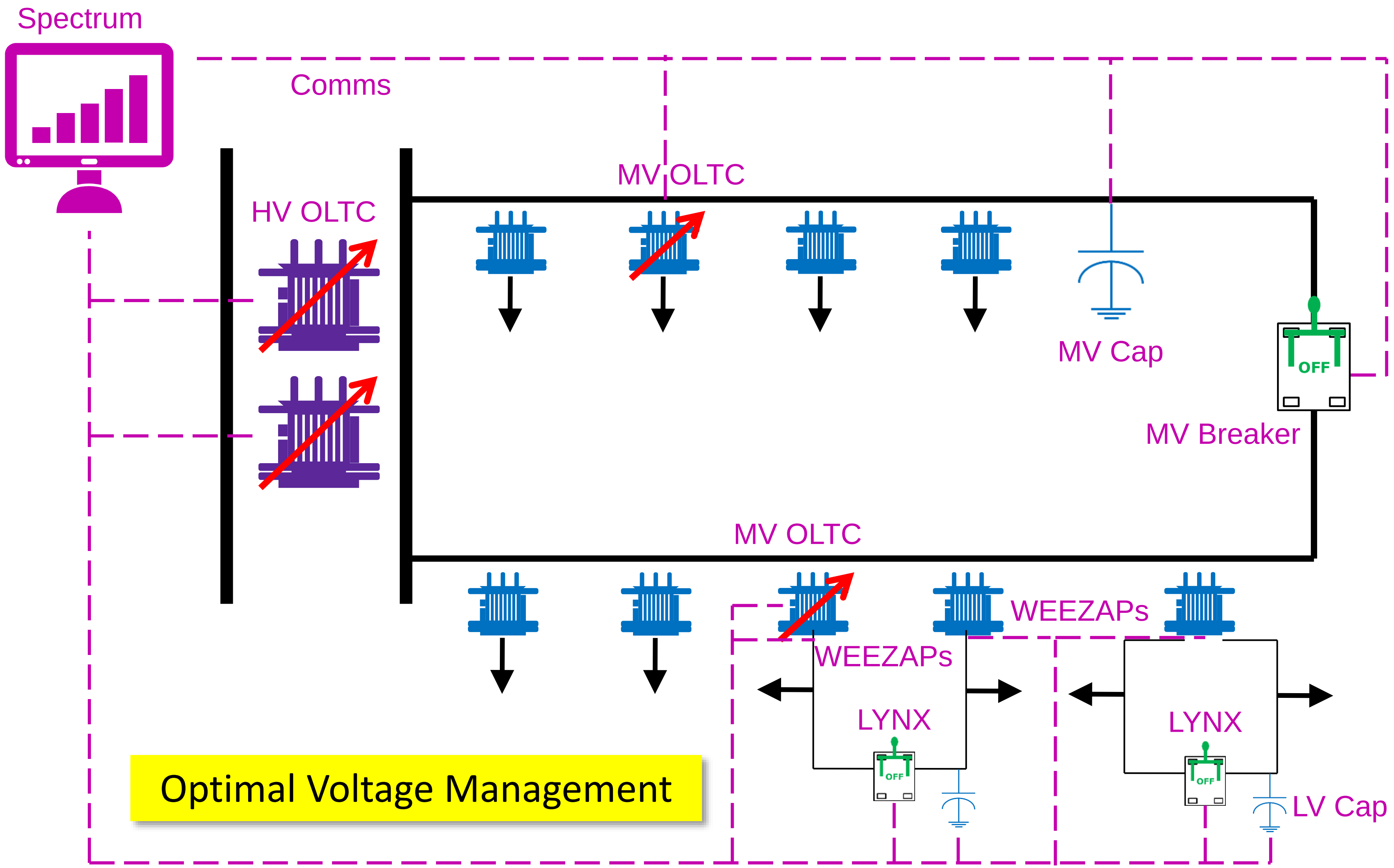
LV Active Voltage Control



Capacitors help to bring back V in highly loaded feeders

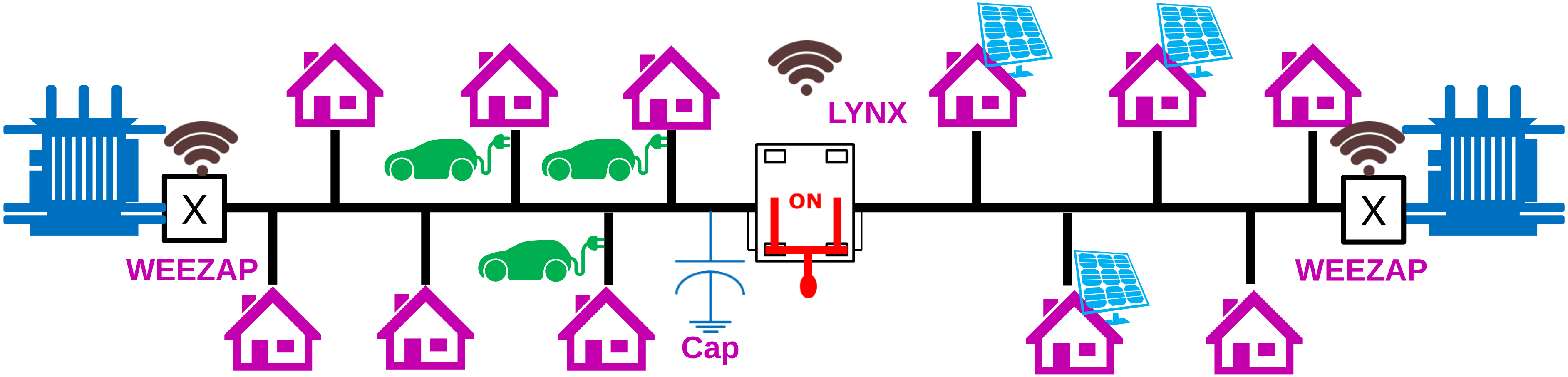
Interconnection helps flattening voltages

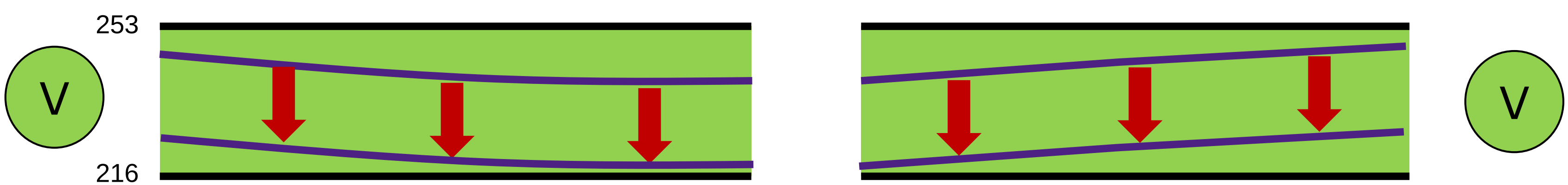
Voltage Control on MV and LV networks



Optimal Voltage Management

Energy Reduction (CVR)





Lower voltages at customer sites

- **Lower energy bills**
- **More LCTs**

Modelling Real Network Control

Real world

SCADA

- Measurements
- Current set points

Interface

- New set points
- Control actions

- Measurements
- Forecast (Loads+DG)

NMS Optimisation Engine

- Set points
- Control actions

Modelling/Simulations

(SCADA)

- 1 or 5 min res. Power flow

- Measurements
- Current set points

(Interface)

- Calculates impact metrics
- Produce forecast

- New set points
- Control actions

- Measurements
- Forecast (Loads+DG)

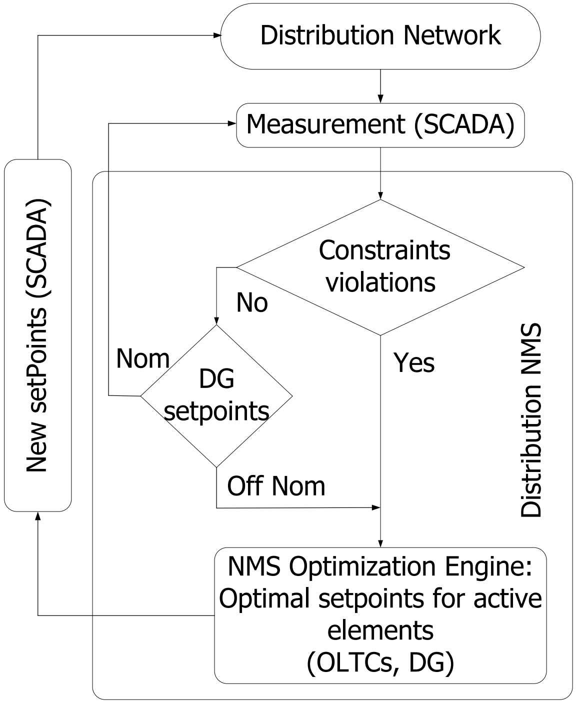
(NMS Opt Engine)

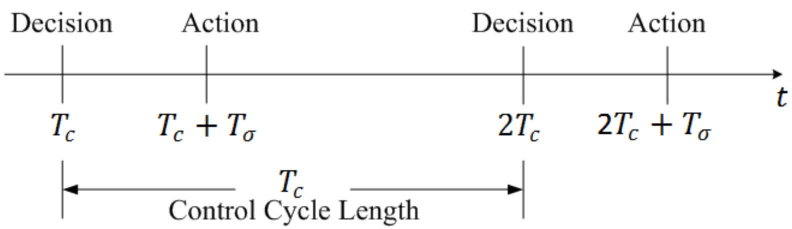
- Solve optimisation problem for control purposes in next control cycle

- Set points
- Control actions

Optimal Control: Example 1

- **OPF-Based Centralised** close to real-time NMS optimisation engine to minimise DG curtailment by actively managing voltages and congestion issues
  - Measurements collected each control cycle (e.g., 5 min)
  - Decisions to solve the seen network issues (Deterministic)
  - Control Action finds the optimum set points
  - OpenDSS-VBA-AIMMS

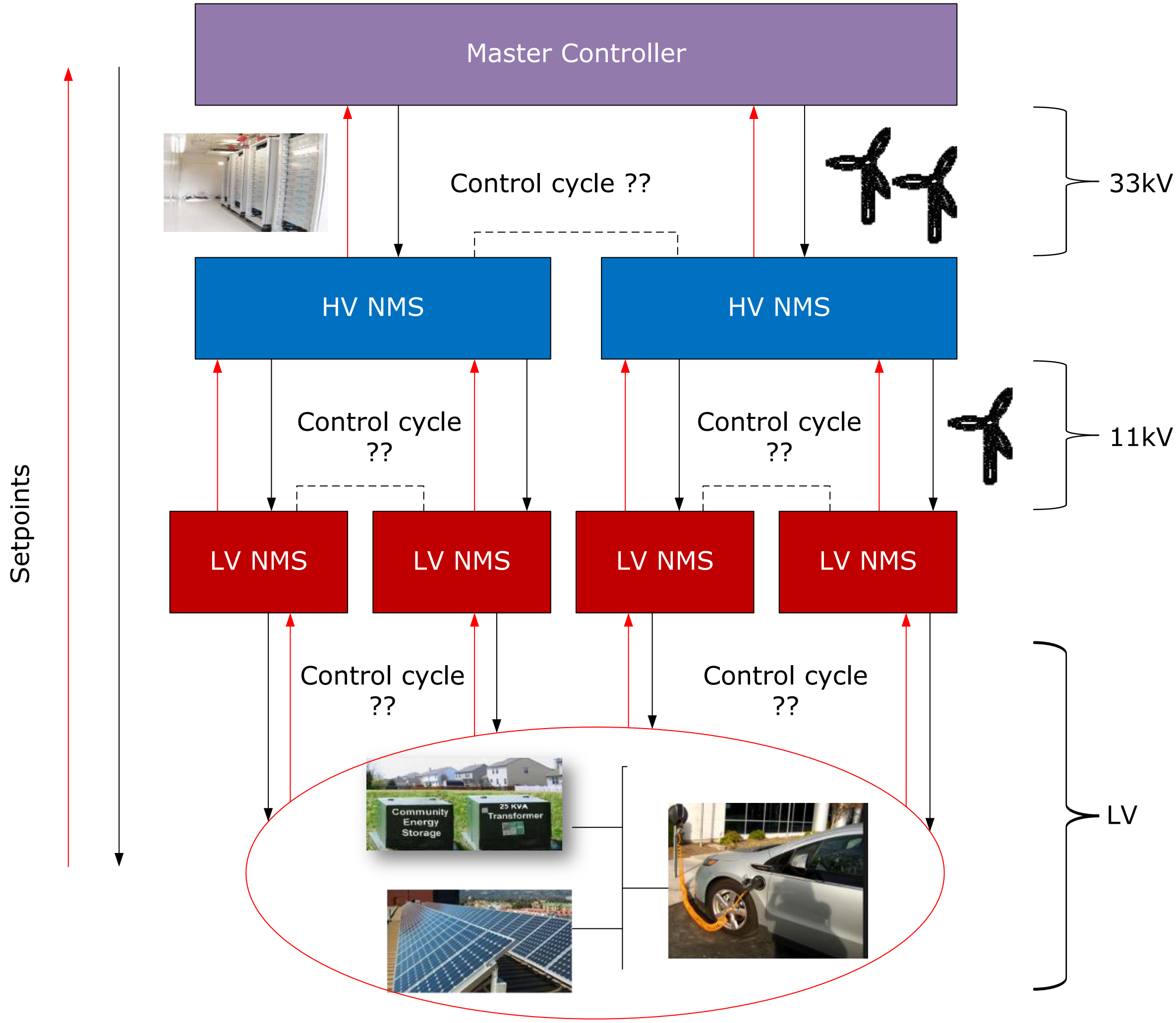




Optimal Control: Example 2

Coordinated, Hierarchical Optimal Control

- Coordination to achieve system wide objective
  - Network areas?
  - Hybrid optimisation?
    - AC OPF
    - DMPC
  - Control cycles?
  - Interactions among voltage levels?
  - Communication networks?



Key Remarks – Optimal Control of Networks

- **Observability** is a currently barrier but soon to be overcome
  - Cost, ICT aspects, data management
- **Complexity** of solutions will increase with more flexibility
  - The extent to which simple rules can be used is unknown (but it is preferred by the industry)
- **Coordination** among solutions is key in BAU implementation

17:45 to 18:00

WRAP UP, CONCLUSIONS AND Q&A

General Conclusions

- **OpenDSS** is a very **flexible** and **comprehensive** power flow engine
  - Interfaces via COM server with Matlab, MS Excel VBA, Python, etc.
  - Time-series three-phase power flows
  - Models for network devices (OLTCs, switches), load models
  - Models for new devices (DG units, storage, etc.)
- **OpenDSS** can be used for **sophisticated Smart Grid studies**
  - Minute by minute simulations, large number of nodes
  - New technologies (e.g., PVs, EVs, wind, storage, etc.)
  - Probabilistic studies (e.g., Monte Carlo)
  - Optimisation studies (e.g., AIMMS-OpenDSS)

ISGT LA Tutorial

Advanced Modelling of Smart Distribution Networks Using OpenDSS

Dr Luis (Nando) Ochoa
luis_ochoa@ieee.org
&
Dr Jairo Quiros-Tortos
jairoquirostortos@ieee.org
The University of Manchester, UK

5th October 2015

Further Reading 1/4

- L.F. Ochoa, P. Mancarella, “Low-carbon LV networks: Challenges for planning and operation,” in Proc. of IEEE/PES General Meeting 2012 (10.1109/PESGM.2012.6344760).
- S.W. Alnaser, L.F. Ochoa, “Hybrid controller of energy storage and renewable DG for congestion management,” in Proc. of IEEE/PES General Meeting 2012 (10.1109/PESGM.2012.6345393).
- A. Navarro, L.F. Ochoa, P. Mancarella, D. Randles , “Impacts of photovoltaics on low voltage networks: A case study for the North West of England,” in Proc. of 22nd International Conference on Electricity Distribution CIRED 2013 (10.1049/cp.2013.1229).
- S.W. Alnaser, L.F. Ochoa, “Distribution network management system: An ac OPF approach,” in Proc. of IEEE/PES General Meeting 2013 (10.1109/PESMG.2013.6672895).

Further Reading 2/4

- A. Navarro, L.F. Ochoa, D. Randles , “Monte Carlo-based assessment of PV Impacts on real UK low voltage networks ,” in Proc. of IEEE/PES General Meeting 2013 (10.1109/PESMG.2013.6672620).
- A. Ballanti, A. Navarro, F. Pilo, L.F. Ochoa “Assessing the Benefits of PV Reactive Power Absorption on a Real UK Low Voltage Network,” in Proc. of IEEE/PES ISGT Europe 2013 (10.1109/ISGTEurope.2013.6695423).
- K Petrou, J. Quirós-Tortós, and L.F. Ochoa, “Controlling EV charging points for congestion management of UK LV networks,” in Proc. of IEEE PES ISGT-America 2015 (10.1109/ISGT.2015.7131843).
- J. Quirós-Tortós, A. Navarro Espinosa, L. Ochoa, M. Gillie and R. Hartshorn, “Probabilistic Impact Assessment of EV Charging on Residential UK LV Networks”, in Proc. of 23rd International Conference on Electricity Distribution CIRED 2015.

Further Reading 3/4

- A. Navarro, T. Gozel, L.F. Ochoa, R. Shaw, D. Randles, “Data analysis of LV networks: Determination of key parameters from one year of monitoring over hundreds of UK LV feeders,” in Proc. of 23rd International Conference on Electricity Distribution CIRED 2015.
- E. Saunders, T. Butler, J. Quiros, L.F. Ochoa, R. Hartshorn, “Direct control of EV charging on feeders with EV clusters,” in Proc. of 23rd International Conference on Electricity Distribution CIRED 2015.
- A. Navarro, L.F. Ochoa, R. Shaw, D. Randles, “Reconstruction of low voltage networks: From GIS data to power flow models,” in Proc. of 23rd International Conference on Electricity Distribution CIRED 2015.
- J. Quiros, L.F. Ochoa, B. Lees, “A statistical analysis of EV charging behavior in the UK,” in Proc. of IEEE PES ISGT Latin-America 2015.

Further Reading 4/4

- S.W. Alnaser, L.F. Ochoa, “Advanced network management systems: A risk-based AC OPF approach” IEEE Transactions on Power Systems, vol. 30, no 1, p 409-418, Jan. 2015 (10.1109/TPWRS.2014.2320901).
- V. Rigoni, L.F. Ochoa, G. Chicco, A. Navarro-Espinosa, T. Gozel, “Representative Residential LV Feeders: A case study for the North West of England”, IEEE Transactions on Power Systems, In Press (10.1109/TPWRS.2015.2403252).
- A. Navarro, L.F. Ochoa, “Probabilistic impact assessment of low carbon technologies in LV distribution systems,” IEEE Transactions on Power Systems, In Press (10.1109/TPWRS.2015.2448663).
- S.W. Alnaser, L.F. Ochoa, “Optimal sizing and control of energy storage in wind power-rich distribution networks ,” IEEE Transactions on Power Systems, In Press (10.1109/TPWRS.2015.2465181).
- J. Quiros, L.F. Ochoa, S.W. Alnaser, T. Butler, “Control of EV charging points for thermal and voltage management of LV networks ,” IEEE Transactions on Power Systems, In Press (10.1109/TPWRS.2015.2468062).