



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

UNIVERSITA' DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale DII

Corso di Laurea Magistrale in Ingegneria dell'Energia Elettrica

Losses allocation techniques in unbalanced low voltage networks

Relatore: Prof. Roberto Turri

Correlatore: Ing. Massimiliano Coppo

Candidato: Marco Vanzetto
1132332

Anno Accademico 2017/2018

“To strive, to seek, to find, and not to yield.”

Alfred Tennyson

Abstract

The allocation of system losses to users, both producers and consumers, is a challenging, yet very useful, topic for distribution networks with a high penetration of distributed generation and with a liberalized electricity market. Meaningful losses allocation techniques are much needed in modern distribution networks, which are more and more active and "smart", to plan and manage the grid as well as to create a fair system of economic penalties and rewards for the users. This work analyses the non-linear nature of the issue and different loss allocation methods existing in literature. One new specific circuit-based procedure, Multi-Phase and Neutral conductors Loss Allocation (MPNLA), is implemented in a multi-conductor environment designed for unbalanced low voltage networks and tested on a real grid under different generation conditions. The methodology presented in this thesis is confirmed to be effective and useful in planning and operation of the network.

Contents

1	Introduction	1
2	Losses Allocation	3
2.1	Introduction to losses allocation	3
2.2	Losses allocation methods	5
2.3	Marginal Procedures	6
2.3.1	Marginal Loss Coefficients	6
2.3.2	Direct Loss Coefficients	8
2.4	Circuit-based Procedures	9
2.4.1	Z-bus Loss Allocation	9
2.4.2	Succinct Method for Loss Allocation	10
2.4.3	Branch Current Decomposition method for Loss Allocation	11
2.4.4	Power Summation Method for Loss Allocation	12
2.5	Admittance Branch Current Loss Allocation	14
2.5.1	The method	15
2.5.2	Considerations	19
3	Implementation of Multi-Phase and Neutral conductors Loss Allocation (MPNLA) method	21
3.1	Correction Current Injection method for power flow analysis of multi-conductor networks	21
3.1.1	Method description	21
3.2	Implementation of MPNLA method	25
3.2.1	Input values	26
3.2.2	Algorithm implementation and validation	27
4	Simulation on e-distribuzione network	33
4.1	The network	33
4.2	Passive scenario	35
4.3	Real case scenario	40
4.4	Optimal scenario	44
4.5	Very active scenario	48
5	Discussion on findings	53
5.1	Allocation of neutral losses	53
5.2	Allocation of transformer losses	55
5.3	MPNLA on a multiple voltage levels network	57
6	Conclusions	61

Bibliography	63
List of acronyms	65
A MPNLA Matlab code	67
B Solutions plotter	73

List of Figures

2.1	Single-conductor schematic for a four-node network	3
2.2	Simplified model for a distribution line	4
2.3	π model for a conductor of a generic network branch	11
2.4	Network branch modelization	15
3.1	π -model of an n-conductor branch element [1]	22
3.2	Shunt element representation where the impedance, current and power correction currents are grouped into $\Delta\mathbf{I}$ [2]	24
3.3	Passive 4-node radial distribution network with balanced loads	25
3.4	Passive 4-node radial distribution network with highly unbalanced loads	25
3.5	Active 6-node radial distribution network, diramated, with balanced loads and generators	26
3.6	Position of the wires along the cross section of the line for fig.3.3, 3.4 and 3.5 networks	26
4.1	Schematic of e-distribuzione network	34
4.2	Voltage magnitudes in e-distribuzione network, passive scenario	36
4.3	Voltage magnitudes in the longest feeder of e-distribuzione network, passive scenario	37
4.4	Magnitude of the branch power losses in e-distribuzione network, passive scenario	38
4.5	Allocated losses in e-distribuzione network, passive scenario, through MPNLA	39
4.6	Voltage magnitudes in e-distribuzione network, real case scenario	41
4.7	Magnitude of the branch power losses in e-distribuzione network, real case scenario	42
4.8	Allocated losses in e-distribuzione network, real case scenario	43
4.9	Voltage magnitudes in e-distribuzione network, optimal Distributed Generation (DG) scenario	45
4.10	Magnitude of the branch power losses in e-distribuzione network, optimal DG scenario	46
4.11	Allocated losses in e-distribuzione network, optimal DG scenario	47
4.12	Voltage magnitudes in e-distribuzione network, current inversion scenario	49
4.13	Voltage magnitudes in the longest feeder of e-distribuzione network, very active scenario	50
4.14	Magnitude of the branch power losses in e-distribuzione network, current inversion scenario	51
4.15	Allocated losses on e-distribuzione network, current inversion scenario	52
5.1	Winding configuration for a Dyn11 transformer	56

5.2 Example of a multiple transformers network 58

Chapter 1

Introduction

Electrification, and therefore the creation of widespread and interconnected electrical grids, has been acknowledged to be the greatest engineering achievement of the XX century [3]. Since its creation though, the electrical system is now changing at a very high pace. The rapid increase in Distributed Generation (DG) - generation plants with nominal power under 10MVA connected to the distribution network (thus concerning Low Voltage (LV) and Medium Voltage (MV) levels) - has dictated a new conception of the grid itself: not a passive radial network anymore, but a much more complex system, where consumers are also producers (hence the term "prosumer") of electricity.

Higher efficiencies and smaller carbon footprint are requirements that are constantly gaining more and more importance. Paris agreement in 2015 stated a world wide effort towards the reduction of CO_2 emissions and sea level rise. Since electricity is the best energetic vector to reach this goal and its penetration is constantly raising, the effort to make it "greener" is substantial. This means to generate it in the most efficient way and transport it to the user with the least losses possible, using renewable energies and carefully managing the power flows in the networks to reduce waste.

Such a new approach enables many innovations to be implemented and more awareness for the user as well, that can participate in a more open and democratic electricity market. It has to be noted however that this rethinking of the grid, which is commonly referred to as "smart grid", come at a price. Electric networks have a big inertia and changes take many years to be planned and commissioned. In the last couple of decades distribution grids have been, and still are, widely and deeply evolving, driven by the need to adapt to the new technologies introduces over time: power-system protections, management systems, regulations and controls needs to be redesigned.

In fact, distribution networks used to be planned to distribute the power from a big centralized power plant towards the loads, all passive and radially placed. Nowadays, generation is pointing towards a decentralize arrangement of small, renewable plants connected to the grid through power electronics. This brings challenges on at least a couple of levels.

Frequency, that used to take advantage of the inertia of the system and its big rotating generators, relies now on electronics and what it is called "digital" or "virtual" inertia. The current flows, once directed solely from the root node (High Voltage (HV) connection point) towards the furthest terminal nodes, now is much more complicated and can even results in a net power transfer from distribution to transmission networks.

Point to point measurements, signals communications, remote control and eventually Internet of Things (IoT) are unavoidable tools to manage the grid of the future. The knowledge on how to understand this huge amount of data and get useful information out of it is essential to reach the aforementioned goals of higher efficiency and smaller

carbon footprint.

Losses Allocation (LA) processes allow for the definition of parameters for each user (producer or consumer) representing its contribution to total system losses. These parameters are important for a number of reasons, including to define economical taxations (or rewards) for each user accordingly to their very own impact on the system losses and to give the Distribution System Operator (DSO) important signals. These can be used for both managing the network in real time, regulating generated power at each node (or even, in a demand response scenario, absorbed power) and planning the renovation and expansion of the grid. These tools come in handy especially since DG is often times synonym of higher imbalances in the grid (the connection of single-phase generators is mostly random amongst the phases and non-regulated), meaning more uncertainties and approximations in understanding and optimizing the network operation.

The economical value of losses allocation in distribution has gain importance for various reasons. Firstly, the liberalization of electricity markets that took place in many countries has changed the way generators are remunerated for their production and the interaction between suppliers and consumers. A fair competition in an open electricity market means that the losses on the path between the generator and the load have to be properly determined and assigned. Secondly, if generators are now spread and mixed to the loads, before rewarding them for the energy transferred to the grid, the DSO should be able to know if they are actually lowering the losses in the system or, by inverting the currents flow, raising them.

Therefore the process of a proper losses allocation is an important step towards a grid which is efficient, that minimize the losses and allows for a big penetration of renewable energy locally produced, where DG is fairly treated and remunerate in an open market.

This work proposes to explain underlying concepts to losses allocation, to analyse different methods present in scientific literature and to implement the most promising one in a multi-conductor software environment for unbalanced low voltage networks. Then, the operation of said process will be tested on a real Italian network. The losses allocation in the grid under different scenarios will be analysed. The objective is to obtain a working algorithm for losses allocation in unbalanced low voltage networks.

Chapter 2

Losses Allocation

The determination of electrical losses on the network and their allocation are key information in both the operation and the optimization of a grid, especially in a smart grid, where power flows are more complex and users (either consumers or generators) can be responsible for both a loss in the system or its non-occurrence. It is important not only to determine the losses in the network, but also to allocate them, to allow for a system in which each user pays or gets rewarded accordingly to its merits or demerits in the network. In addition to that, losses allocation, as it will be later discuss in this thesis, can give a lot of information to the Distribution System Operator to lever the efficiency of the grid and lower the overall losses costs.

2.1 Introduction to losses allocation

There are a lot of different methods to allocate losses and, prior to discussing them, it is important to understand why this is. Let's introduce a very basic, one-conductor, network with four nodes and three branches, as shown in Fig.2.1. There are two loads attached, called L_3 and L_4 , which are absorbing two currents, I_{L3} and I_{L4} respectively.

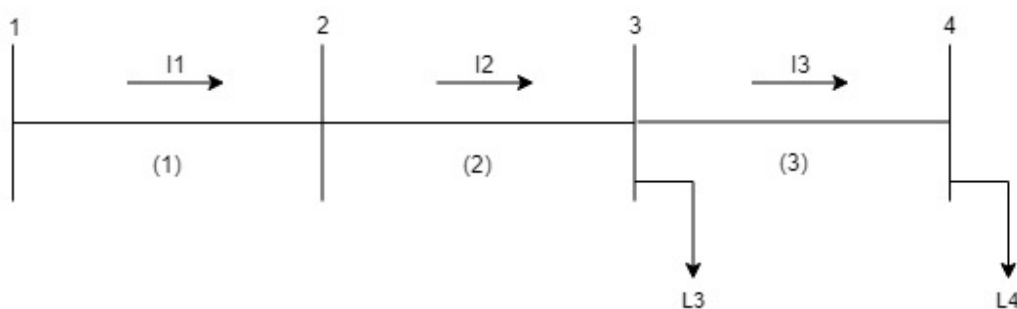


Figure 2.1: Single-conductor schematic for a four-node network

It is known that every line has an electrical impedance, with a real part, which is the resistance and is responsible for Joule losses, and an imaginary part, which is the reactance. The flow of a current through a line, hence through an impedance, determines a loss. In electrical grids, the term "loss" indicates the power dissipated on a line by Joule effect, thus referring just to the real part of said power.

Since the power dissipated on the branch is equal to the difference of input and output power, the loss on the branch can also be written as:

$$\bar{S}_{loss} = \bar{S}_{in} + \bar{S}_{out} = \bar{V}_{in}\bar{I} - \bar{V}_{out}\bar{I} \quad (2.1)$$

where \bar{S}_{loss} is the dissipated power phasor, \bar{S}_{in} and \bar{S}_{out} are the phasors of the nodal power on the nodes in input and output of the branch, \bar{V} and \bar{I} are the phasors of voltage and current respectively. The used convention considers positive the currents when directed outward of the node. The same convention is used for power.

Let us consider branch a , therefore the line linking node 1 to node 2. The current flowing is in it I_a .

$$\bar{S}_{a,loss} = P_{a,loss} + jQ_{a,loss} = \bar{V}_1\bar{I}_a - \bar{V}_2\bar{I}_a = (\bar{V}_1 - \bar{V}_2)\bar{I}_a \quad (2.2)$$

The term $\bar{V}_1 - \bar{V}_2$, which is the voltage drop on the line, can be obtained as:

$$\bar{V}_1 - \bar{V}_2 = \dot{Z}\bar{I}_a = (R + jX)\bar{I}_a \quad (2.3)$$

in the very basic assumption that line a can be model as an impedance \dot{Z} , shown in Fig.2.2.

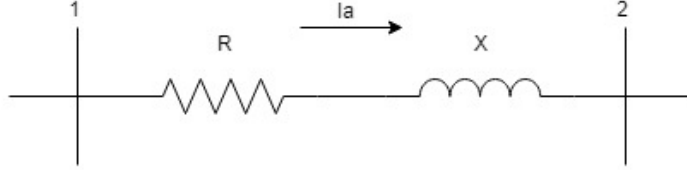


Figure 2.2: Simplified model for a distribution line

Knowing that the current \bar{I}_a can be written as the sum of the current absorbed by the loads connected downstream of node 2 and that the focus is only on the active power dissipated, from eq. 2.2 and 2.3 follows that:

$$P_{a,loss} = R(\bar{I}_{L3} + \bar{I}_{L4})^2 = R|\bar{I}_{L3}|^2 + R|\bar{I}_{L4}|^2 + 2R\bar{I}_{L3}\bar{I}_{L4} \quad (2.4)$$

Equation 2.4 highlights how losses on the branch depend on the squared values of the currents of each load but also on the cross product between them. Losses are non-linear function of line flows. The presence of cross-terms associated with a quadratic function is what prevent us from allocating directly losses to generators and loads [4]. Equation 2.4 also points out how the effect of the very same user on the network varies widely according to the conditions of the grid. For instance, even if \bar{I}_{L3} is constant, the cross-terms depend on all the currents going through the branch.

To use an analogy, in an highway a car is not responsible for a certain amount of traffic by itself. If a user drives the car during the night, when the highway is mostly empty, we can consider it responsible for very little traffic, but if the car were to be driven during peak hour, its contribution to the traffic level would have been definitely higher. The presence of cross-terms means that losses allocation give each car a fee that is not related to the mileage covered, but to the time spent driving in the highway.

This important notion precludes the utilization of one unique losses allocation procedure and opens up multiple possibilities.

2.2 Losses allocation methods

A LA method is designed to divide the totality of the active losses in the network amongst all users in the fairest way possible.

To perform this task, many approaches can be undertaken, leading to different calculations and results. Some of these procedures are design to work better in a transmission system, others in a distribution one, some are tailored for a low DG environment, others work well also with high imbalances in the network.

Prior to give a general overview of the main methods known in scientific literature, it is convenient to divide them into three families of procedures, accordingly to their initial approach:

- Pro Rata (PR) Procedures: These methods are based on *a priori* assumptions. Losses are globally assigned to generators and consumers, e.g. 50% to each category. A proportional law is then applied to allocate losses to users accordingly to the level of energy consumed or generated [5].
- Marginal Procedures: Losses are assigned to users through incremental coefficients, built from the jacobian of the power flow to mimic the trend of variation of powers and losses. Therefore these are also called derivative-based methods and the most widely known are the Marginal Loss Coefficients (MLC) and the Direct Loss Coefficients (DLC).
- Proportional Sharing Procedure: This method is based on the power flow results and it states that the total nodal power coming from the input branches on a certain node is proportionally shared between the output branches accordingly to their power flow [4].
- Circuit-based Procedures: These methods consider the whole circuitual model of the network and from it they calculate the losses on the exact nodal power injections performed by the users at a given instant in time. Z-bus, Succinct Method for Loss Allocation (SMLA), Branch Current Decomposition method for Loss Allocation (BCDLA) and Power Summation Method for Loss Allocation (PSMLA) are examples.

Since PR Procedures are based on prior assumptions that are too approximated for an unbalanced distribution network [6], they are not discussed further. Similarly, Proportional Sharing Procedure is not going to be presented in this thesis, since in this method losses have to be assigned firstly to consumers and secondly to generators (or vice versa) without any account for the cross effect of the two and this cannot be a viable option in an unbalanced network with high presence of DG.

Prior to discussing each method in details, it should be emphasized that the purpose of loss allocation is to assign the "cost" of losses to each user, it being either a consumer or a generator. Considering no unique or ideal method exists, desirable features for a loss allocation mechanism are [4]:

- Consistency with the power flow results;
- Dependency on the actual load/generation amount and conditions;
- Topology-dependency, since each network has a different topology and layout and this conditions the losses;

- Fairness between users;
- Being easy to understand, simple to implement;
- Giver of appropriate economic marginal signals and technical indicators.

The following paragraphs will give an overview on how marginal procedures and circuit-based procedures work, since they are the only two categories of methods eligible for unbalanced low voltage networks with high penetration of DG.

2.3 Marginal Procedures

Derivative-based methods use the results of the load flow to build a set of loss allocation factors able to follow the trend of variation of the losses when the generated or absorbed power varies [6].

2.3.1 Marginal Loss Coefficients

By definition, this method calculates the Marginal Loss Coefficients (MLC) ρ , which are adimensional and measure the change in total active power losses L due to a marginal change in consumption or generation of both active and reactive power at each node i of the network, except for the slack node [7].

$$\rho_{P_i} = \frac{\partial L}{\partial P_i} \qquad \rho_{Q_i} = \frac{\partial L}{\partial Q_i} \qquad (2.5)$$

If a user, i.e. a generator, takes part in the network regulation (voltage control) by injecting a certain amount of reactive power (behaving as a PV node instead of a PQ node), no reactive losses can be charged for this user.

$$\rho_{Q_i} = 0 \qquad \text{where } i \text{ is a PV node} \qquad (2.6)$$

Since the slack node is, by definition, the node meant to supply all the losses calculated in the load flow, the coefficients for this particular node has to be zero. Total power losses L are insensitive to changes in active and reactive power at the slack node.

$$\rho_{P_s} = 0 \qquad \rho_{Q_s} = 0 \qquad \text{where } s \text{ is the slack node} \qquad (2.7)$$

This makes the choice of the slack node influential in the losses allocation process on transmission networks, but it does not impact the procedure in distribution networks, where there is always a point of connection to the transmission network to be taken as univocal slack node.

A vector $\boldsymbol{\rho}$ with dimension $2N \times 1$, where N is the number of nodes, is built using all MLCs ρ , putting first all ρ_{P_i} and then all ρ_{Q_i} . Multiplying the $2N \times 2N$ matrix \mathbf{A} , which is the transpose of the jacobian in the Newton-Rapson load flow, by $\boldsymbol{\rho}$, a new $2N \times 1$ vector $\tilde{\boldsymbol{\rho}}$ can be obtain.

$$\begin{bmatrix} \frac{\partial L}{\partial \theta_1} \\ \vdots \\ \frac{\partial L}{\partial \theta_N} \\ \frac{\partial L}{\partial V_1} \\ \vdots \\ \frac{\partial L}{\partial V_N} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_1}{\partial \theta_1} & \cdots & \frac{\partial P_N}{\partial \theta_1} & \frac{\partial Q_1}{\partial \theta_1} & \cdots & \frac{\partial Q_N}{\partial \theta_1} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial P_1}{\partial P_1} & \cdots & \frac{\partial P_N}{\partial P_1} & \frac{\partial Q_1}{\partial P_1} & \cdots & \frac{\partial Q_N}{\partial P_1} \\ \frac{\partial P_1}{\partial V_1} & \cdots & \frac{\partial P_N}{\partial V_1} & \frac{\partial Q_1}{\partial V_1} & \cdots & \frac{\partial Q_N}{\partial V_1} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial P_1}{\partial V_N} & \cdots & \frac{\partial P_N}{\partial V_N} & \frac{\partial Q_1}{\partial V_N} & \cdots & \frac{\partial Q_N}{\partial V_N} \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial P_1} \\ \vdots \\ \frac{\partial L}{\partial P_N} \\ \frac{\partial L}{\partial Q_1} \\ \vdots \\ \frac{\partial L}{\partial Q_N} \end{bmatrix} \quad (2.8)$$

$$\tilde{\rho} = \mathbf{A} \cdot \rho \quad \mathbf{A} = \mathbf{J}^T \quad (2.9)$$

Thus ρ represents MLCs vector whilst $\tilde{\rho}$ is the vector representing total losses sensitivity with respect to voltage angle and magnitude at each node. The system in eq.2.8 and 2.9 is linear and easy to solve, given L .

Total system losses L can be calculated as:

$$L = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N G_{ij} [|\bar{V}_i|^2 + |\bar{V}_j|^2 - 2|\bar{V}_i||\bar{V}_j|\cos(\theta_i - \theta_j)] \quad (2.10)$$

where G_{ij} stands for the conductance of the network from node i to node j . Going forward with the substitutions:

$$\frac{\partial L}{\partial \theta_i} = 2 \sum_{j=1}^N G_{ij} |\bar{V}_i| |\bar{V}_j| \sin(\theta_i - \theta_j) \quad i = 1, \dots, N \quad (2.11)$$

$$\frac{\partial L}{\partial |\bar{V}_i|} = 2 \sum_{j=1}^N G_{ij} [|\bar{V}_i| - |\bar{V}_j| \cos(\theta_i - \theta_j)] \quad i = 1, \dots, N \quad (2.12)$$

Note that the calculation of total losses using MLCs with the outlined procedure yields to estimate roughly double of the actual amount [7].

$$\sum_{i=1}^{N-1} [\rho_{P_i} P_i + \rho_{Q_i} Q_i] \simeq 2L \quad (2.13)$$

Therefore a reconciliation process is much needed and it is applied through a constant multiplier reconciliation factor k_0 , calculated as follow:

$$\hat{\rho} = k_0 \rho \quad k_0 = \frac{L}{\sum_{i=1}^{N-1} (\rho_{P_i} P_i + \rho_{Q_i} Q_i)} \quad (2.14)$$

Reconciled MLCs $\hat{\rho}$ enable the total losses to be corrected to match the ones from the load flow whilst allocating active power losses to each individual user.

2.3.2 Direct Loss Coefficients

DLC process overcomes the issue of reconciliation of the previous method with a direct relation between nodal powers and losses calculated through DL coefficients γ . Whilst MLC allocates marginal losses, DLC allocates total losses. The objective of the latter is to obtain a relationship such that losses can be directly expressed as terms of nodal injections (i.e. nodal powers).

Similarly to MLC, total losses L in the system can be compute as in eq.2.10, knowing the load flow solution after its convergence.

For a given variation Δ in the variables V and θ , indicating respectively magnitude and angle of the voltage phasor \bar{V} , losses can be allocated using Taylor series expansion around the initial operating points V^0 and θ^0 . Using a matrix notation:

$$\begin{aligned}
 L &= f(\theta^0 + \Delta\theta, V^0 + \Delta V) \\
 &= f(\theta^0, V^0) + [\Delta\theta \quad \Delta V] \left[\frac{\partial L}{\partial \theta} \quad \frac{\partial L}{\partial V} \right]^T + \frac{1}{2} [\Delta\theta \quad \Delta V] [\mathbf{H}] [\Delta\theta \quad \Delta V]^T + \dots \\
 &\approx \frac{1}{2} [\Delta\theta \quad \Delta V] [\mathbf{H}] [\Delta\theta \quad \Delta V]^T \\
 &= \frac{1}{2} [\Delta\theta \quad \Delta V] [\mathbf{H}][\mathbf{J}]^{-1} [\Delta P \quad \Delta Q]^T \\
 &= \gamma [\Delta P \quad \Delta Q]^T
 \end{aligned} \tag{2.15}$$

where $[\mathbf{H}]$ is the hessian matrix, $\Delta\theta$ and ΔV are the vectors representing changes in operating point with respect to the initial one, assuming $V_i^0 = 1$ and $\theta_i^0 = 0$ so that:

$$L_0 = f(\theta^0, V^0) = 0 \quad \text{System losses under initial condition} \tag{2.16}$$

Since it is important to express losses directly in terms of nodal power injections:

$$[\Delta\theta \quad \Delta V]^T = [\bar{\mathbf{J}}]^{-1} [\Delta P \quad \Delta Q]^T \tag{2.17}$$

where $[\bar{\mathbf{J}}]$ is an average jacobian computed from the initial $[\mathbf{J}^0]$ to the final convergence $[\mathbf{J}]$.

Thus, from eq.2.15, the vector γ is given, composed by every DLCs for each i node. Now, summing up all the losses allocated to each node, total losses L can be calculated, as a proof of the accuracy of the method and that no reconciliation is needed. Therefore, the use of assumptions and approximations does not impact severely the outcome.

Marginal procedures have the advantage of using the results of the load flow, without needing any other input. On the other hand, they both need the use of Newton-Rapson algorithm in order to obtain the load flow and this is a limitation, especially in low voltage distribution networks, where R/X ratio is high and systems are often unbalanced. In this conditions, a load flow algorithm that does not use any single phase equivalent circuit and converge easily, even with high R/X ratios, can yield to faster and better results, as will be discussed in chapter 3. The drawbacks of these procedures (possible inaccuracy and instability of the load flow and the dependency of the result upon the choice of the slack node) are to be carefully considered when applying them to unbalanced low voltage networks.

2.4 Circuit-based Procedures

Circuit-based methods rely on the system structure, expressed by the network impedance matrix, and on the results of the load flow. They do not exploit any simplifying assumption.

2.4.1 Z-bus Loss Allocation

As the name suggests, this method, originally designed for transmission networks, takes advantage of the bus impedance matrix. This LA process emphasizes currents more than powers, given that the firsts are dominant factors in loss determination. The objective of the method is to take a solved power flow and to distribute the system losses P_{loss} amongst the N network buses as L_k , with $k = 1, \dots, N$, so that the summation of all L_k gives back P_{loss} [8].

Firstly, consider the network admittance matrix $\dot{\mathbf{Y}}$, composed by $G + jB$ (G being the conductance of the branch and B the susceptance) terms, typically large, sparse and nonsingular. This last condition is important and it is verified only if the system has non-negligible nodal shunt admittances.

This is true for transmission networks, but it can be not true for distribution networks, e.g. MV systems with isolated neutral, where there is no reference to ground potential and $\dot{\mathbf{Y}}$ matrix will have null elements, resulting in the impossibility of inverting the matrix.

The power flow solution provides, among other quantities, the vectors of complex bus injected currents $\bar{\mathbf{i}}$ and complex bus voltages $\bar{\mathbf{v}}$. Real losses can be expressed as:

$$\bar{\mathbf{i}} = \begin{bmatrix} \bar{I}_1 \\ \vdots \\ \bar{I}_N \end{bmatrix} \quad \bar{\mathbf{v}} = \begin{bmatrix} \bar{V}_1 \\ \vdots \\ \bar{V}_N \end{bmatrix} \quad (2.18)$$

$$P_{loss} = \mathbb{R}\left\{\sum_{k=1}^n \bar{V}_k \bar{I}_k^*\right\} \quad (2.19)$$

$$P_{loss} = \mathbb{R}\left\{\sum_{k=1}^n \bar{V}_k \left(\sum_{j=1}^n \dot{Y}_{kj}^* \bar{V}_j^*\right)\right\} \quad (2.20)$$

$$P_{loss} = \mathbb{R}\left\{\sum_{k=1}^n \bar{I}_k^* \left(\sum_{j=1}^n \dot{Z}_{kj}^* \bar{I}_j\right)\right\} \quad (2.21)$$

Of these formulation, the last one, eq.2.21, is the one that provide the best results, since losses are directly linked to currents, which are independent variables. The impedance can be rewritten into resistance and reactance. It can be proven that the second term in eq.2.23 is null [8], since usually $\dot{\mathbf{Y}}$ matrix is symmetrical in transmission networks, so that losses are function of resistance only. This assumption presents a certain degree of approximation when distribution networks are considered.

$$P_{loss} = \mathbb{R}\left\{\sum_{k=1}^n \bar{I}_k^* \left(\sum_{j=1}^n \dot{Z}_{kj}^* \bar{I}_j\right)\right\} \quad (2.22)$$

$$= \mathbb{R}\left\{\sum_{k=1}^n \bar{I}_k^* \left(\sum_{j=1}^n R_{kj}^* \bar{I}_j\right)\right\} + \mathbb{R}\left\{\sum_{k=1}^n \bar{I}_k^* \left(\sum_{j=1}^n j X_{kj}^* \bar{I}_j\right)\right\} \quad (2.23)$$

$$= \mathbb{R}\left\{\sum_{k=1}^n \bar{I}_k^* \left(\sum_{j=1}^n R_{kj}^* \bar{I}_j\right)\right\} \quad (2.24)$$

This formulation to compute losses gives a natural separation of the system losses among the network buses. To reshape this in a matrix form, consider the vector of real power injections as:

$$\mathbf{p} = \mathbb{R}\{\text{diag}(\bar{\mathbf{i}}^*)\mathbf{R}\bar{\mathbf{i}}\} + \mathbb{R}\{j \cdot \text{diag}(\bar{\mathbf{i}}^*)\mathbf{X}\bar{\mathbf{i}}\} \quad (2.25)$$

Net real power injections can be divided into two separate components, representing the load demand and the loss occurring:

$$\mathbf{l} = \mathbb{R}\{\text{diag}(\bar{\mathbf{i}}^*)\mathbf{R}\bar{\mathbf{i}}\} \quad (2.26)$$

$$\mathbf{d} = \mathbb{R}\{j \text{diag}(\bar{\mathbf{i}}^*)\mathbf{X}\bar{\mathbf{i}}\} \quad (2.27)$$

By comparing 2.24 and 2.26 it is clear that the sum of the losses L_k , components of the vector \mathbf{l} , over all buses is equal to the system losses. The sum of the network terms D_k is equal to zero [8] and this confirms that system losses are naturally separated into L_k terms, depending on complex bus current injections.

$$L_k = \mathbb{R}\left\{\bar{I}_k^* \left(\sum_{j=1}^n R_{kj}^* \bar{I}_j\right)\right\} \quad (2.28)$$

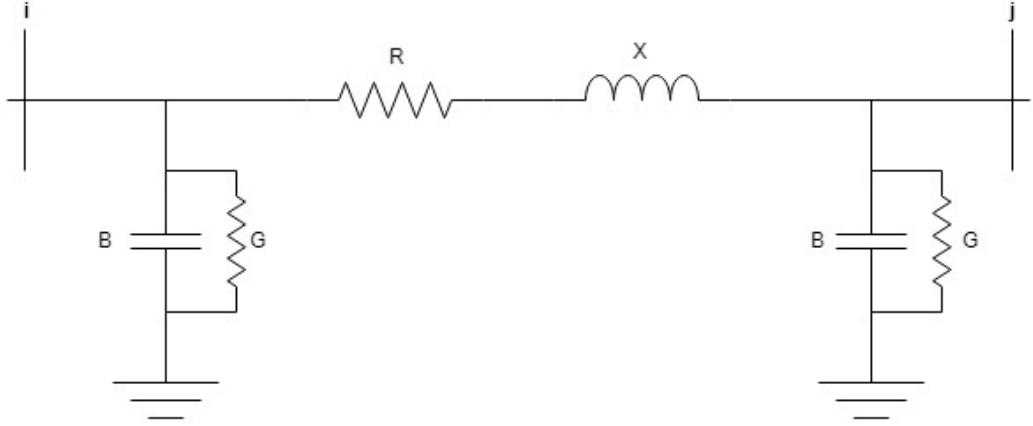
Therefore this method emphasizes the losses current dependency and uses no special assumption or approximation. Its limits are the already discussed requirements for the network, necessary to obtain a solution.

2.4.2 Succinct Method for Loss Allocation

SMLA is a procedure that, unlike Z-bus, is designed to allocate only the variable loss due to the series branch impedance, whereas the almost constant losses due to shunt admittances are allocated in average terms amongst all users. Only one assumption has to be made: maintaining the voltage profile of the network at an acceptable level is a common obligation between all participants [9].

Let us consider a generic power system network of N nodes, where active and reactive powers, P and Q respectively, are injected in the nodes. Fig.2.3 shows the impedances model of a branch of said network.

From the load flow computation, the amplitude V_i and phase angle δ_i of nodal voltages can be obtained. Since the non-variable component of losses, due to shunt branch admittances, will be allocated averagely amongst all users, consider now the branch b , with $b = 1, 2, \dots, B$, having i and j as sending and receiving ends and a series impedance $\dot{Z}^{(b)} = R + jX$. The losses L_k allocated to each node $k = 1, 2, \dots, N$ are:

Figure 2.3: π model for a conductor of a generic network branch

$$L_k = \Re \left\{ \sum_{b=1}^B \left(\frac{z_{ik} - z_{jk}}{\bar{Z}^{(b)}} \right) \left(\frac{\bar{V}_i - \bar{V}_j}{\bar{V}_k} \right)^* (P_k - jQ_k) \right\} \quad (2.29)$$

In [6] it is proven how this method, while being accurate in most cases and having a lot of advantages, presents a flaw, a paradox that prevents it from providing meaningful loss allocation under specific circumstances. It has been shown that, considering two loads A and B connected to the same bus, absorbing an equal amount of active power with different power factors, higher losses will be allocated to the load absorbing lower reactive power (therefore lower current) rather than to the load with higher reactive power. The authors propose a solution to this paradox, that can be found in [6].

2.4.3 Branch Current Decomposition method for Loss Allocation

BCDLA is proposed as the method that goes beyond the flaws of previously examined processes, that are limitations in the choice of load flow techniques and computationally intense algorithm for DLC and MLC, need for shunt elements for Z-bus, allocation paradoxes for SMLA.

BCDLA uses a simple and effective mechanism based on a forward sweep method [10], which does not require computing the power flow Jacobian nor the bus impedances matrix. This method is designed specifically for distribution networks and is able to work with any load flow technique, even with multi-conductor analysis on unbalanced systems, as proven in [11].

Consider a radial distribution system, where the root node 0 is taken as slack bus, with voltage magnitude and angle as reference. B_k is the set of the branches that connect node k to the root node and $K^{(b)}$ is the set of nodes supplied by branch b (located downward of branch b). Branch and nodal currents can be written respectively as:

$$\bar{I}^{(b)} = \alpha^{(b)} + j\beta^{(b)} \quad \bar{I}_k = \alpha_k + j\beta_k \quad (2.30)$$

Nodal currents can be expressed in terms of net active and reactive nodal output power (at least for the load convention):

$$\bar{I}_k = \alpha_k + j\beta_k = \frac{P_k - jQ_k}{\bar{V}_k^*} \quad (2.31)$$

Branch losses can then be expressed in the following way:

$$\begin{aligned} L^{(b)} &= R^{(b)} |\bar{I}^{(b)}|^2 = R^{(b)} \left[\left(\alpha^{(b)} \right)^2 + \left(\beta^{(b)} \right)^2 \right] \\ &= \left(R^{(b)} \alpha^{(b)} \right) \alpha^{(b)} + \left(R^{(b)} \beta^{(b)} \right) \beta^{(b)} \end{aligned} \quad (2.32)$$

where $R^{(b)}$ is the branch b resistance. Since the branch current flowing in b can be written as the sum of the currents injected in the nodes belonging to the set $K^{(b)}$, it follows:

$$L^{(b)} = R^{(b)} \alpha^{(b)} \sum_{k \in K^{(b)}} \alpha_k + R^{(b)} \beta^{(b)} \sum_{k \in K^{(b)}} \beta_k \quad (2.33)$$

$$L_k^{(b)} = R^{(b)} \alpha^{(b)} \alpha_k + R^{(b)} \beta^{(b)} \beta_k \quad \text{for } \beta \in K^{(b)} \quad (2.34)$$

Therefore total losses can be easily allocated to each k node through eq.2.35.

$$L_k = \sum_{b=1}^B L_k^{(b)} = \alpha_k \sum_{b \in B_k} \left(R^{(b)} \alpha^{(b)} \right) + \beta_k \sum_{b \in B_k} \left(R^{(b)} \beta^{(b)} \right) = c_k \alpha_k + d_k \beta_k \quad (2.35)$$

c_k and d_k can be seen as real and imaginary part of a "virtual" voltage $\bar{W}_k = c_k + jd_k$ at node k :

$$L_k = \mathbb{R} \left\{ \bar{W}_k \bar{I}_k^* \right\} \quad (2.36)$$

"Virtual" voltages for each node can be calculated simultaneously as the voltage drop on the path described by B_k from a single forward sweep on a modified system, where all the series branch impedances have been replaced by their resistive components.

This method is very simple, takes full advantage of the power flow results, with no additional assumptions nor approximations, it does not require any reconciliation mechanism, implicitly includes sensitivity information and is able to represent exactly the total losses. The main advantage over marginal procedures is the ability to allocate losses related to reactive power even to (A node defined by) Power and Voltage (PV) nodes across the grid (e.g. voltage-controllable local generators, a desirable feature in system with high presence of DG).

2.4.4 Power Summation Method for Loss Allocation

PSMLA works in a similar but opposite way to BCDLA. It is designed for radial distribution systems but it uses a backward sweep instead of a forward one. It also emphasizes powers more than currents [12]. PSMLA establishes a direct link between losses in each branch of the network and the injected active and reactive power in the nodes, considering the path connecting each node to the root node. Similarly to

BCDLA, active and reactive power at the receiving node of each branch are decomposed as sum of injected nodal powers and branch losses fed by said branch.

The power summation process starts from the terminal branches by allocating losses on each line to nodes supplied by it. Then, injected power at these nodes are corrected with allocated losses. Reactive power and quadratic cross-terms are included in the allocation. In the process, DGs are considered as negative active and reactive power injections, loads as positive ones.

Let us consider now a radial distribution network, with $N + 1$ nodes, where the slack node is assigned to node 0. Since the system is radial, there is only one path between each node and the root node. There is an order in the index assignment, each branch has an orientation: positive from sending to receiving node, sending node being closer to the root and therefore with a smaller index number.

Power losses in each branch m are calculated as follows:

$$\Delta P_m + j\Delta Q_m = \dot{Z}_m I_m^2 = \frac{R_m + jX_m}{V_m^2} \left(P_{m(end)}^2 + Q_{m(end)}^2 \right) \quad (2.37)$$

where \dot{Z}_m , R_m and X_m are impedance, resistance and reactance of branch m , I_m is the current magnitude in the branch m , V_m is voltage magnitude of the receiving end of branch m , $P_{m(end)}$ and $Q_{m(end)}$ are active and reactive power at the receiving end of branch m .

This losses can be decomposed along the set α of nodes connecting m to the root node as:

$$\Delta P_m + j\Delta Q_m = \sum_{i \in \alpha} \Delta P_m^i + j \sum_{i \in \alpha} \Delta Q_m^i \quad (2.38)$$

In order to determine each ΔP_m^i component, power flow at the receiving end of any branch m can be expressed as sum of powers of nodes and losses of branches fed by m .

$$P_{m(end)} + jQ_{m(end)} = \sum_{k \in \alpha} (P_k + jQ_k) + \sum_{k \in \alpha, k \neq m} (\Delta P_k + j\Delta Q_k) \quad (2.39)$$

A backward sweep process can now be run: the procedure for loss allocation will start from the branch with the highest index N (terminal branch). Once the active and reactive power losses for the terminal branch $\Delta P_N + j\Delta Q_N$ are calculated, they are added as shunt loads to receiving end of the branch and the line is now considered to be lossless. In any further calculation (next branch will be the one that feed N , so the furthest non-terminal branch from the root), corrected injected power $P_{N,corr} + jQ_{N,corr}$ will be used for the terminal node and impedance of N will be null.

$$P_{N,corr} + jQ_{N,corr} = (P_N + \Delta P_N) + j(Q_N + \Delta Q_N) \quad (2.40)$$

In this way all branches are processed, from terminal to root, considering cross-terms and mutual effects of loads and DGs. More in depth calculation and procedures can be found in [12].

The application of the method shows that PSMLA yields to good results in distribution networks, comparable with BCDLA. Losses are calculated with good precision, allocated fairly and equally, dealing very well with cross-terms. The procedure gives also sensible indications on the system behaviour. It shares all advantages with PSMLA with one exception: in literature, this method is yet to be tested in a multi-conductor

environment. Early implementations of this procedure in multi-conductor systems reveal that, whilst the allocation of losses is still generally fair, there are some issues in the computation of losses derived by mutual inductance between phases that lead minor to inconsistencies in highly unbalanced systems.

2.5 Admittance Branch Current Loss Allocation

Pros and cons of different procedures have been highlighted, with a particular attention to their application to unbalanced low voltage networks with DG. In his work [13], Nicola Bressan presents a new procedure, that proposes to overcome issues or limitation found in other methods.

Objectives of this procedure, in addition to the advisable feature already discussed for a generic loss allocation method, are:

- To work well in a distribution grid, where R/X ratio might be high and shunt elements of the branches might be so small to be considered negligible. This configuration makes it difficult for transmission designed power flow algorithms (e.g. Newton-Rapson method) to always converge to solution. Hence, this procedure need to be independent from the power flow algorithm.
- To work properly in an asymmetrical, unbalanced system, as a LV one is, where single phase loads and generators are not only present, but represent the vast majority of users. Single-phase equivalent circuits cannot be used under these conditions and multi-conductor power flows and loss allocation procedures are much desirable.
- To treat equally generators on the network that acts as voltage regulators and therefore are recognised as PV nodes rather than PQ nodes.
- To avoid allocation paradoxes as the one discussed in [6], where losses were not allocated equally by SMLA under particular reactive power conditions, given that reactive power in distribution networks can represent a significant portion of total power flow.

Admittance Branch Current Loss Allocation (ABCLA) is a circuit based procedure that mixes two of the previously analysed methods, Z-bus and BCDLA.

This method is inherently built for a multi-conductor environment. Therefore the following analysis of the process is made on the most representative situation of a real case scenario for a distribution network, a 4-conductor 3-phase system with neutral, but can be extended for any m -conductor network. It is important to point out that the neutral conductor is treated equally with respect to the other phase conductors, so much so it has losses allocated to it in the same way. The allocation of neutral losses to users will be discussed in chapter 5, but for now it is important to highlight how this reveals that ABCLA aims for a "technical allocation" more than for a "fair-to-the-user" one. Loss allocation factors are first of all indicators for the DSO, giving important information and signals about the operative conditions of network and the effects of DG on it. Since there is no immediate correlation in mind between these factors and an economic penalization/revenue for the user, these factors can be post-processed to improve fairness to the user, if needed.

2.5.1 The method

Let us consider a generic radial distribution system, like the one in fig.2.1. Power flow solution (meaning nodal voltages) is given, obtained through any methodology. In addition to that, the admittance matrix for the entire network is required. This can be outputted from the load flow calculations or obtained as follows.

Let us consider the branch represent in fig.2.4 and its conventions.

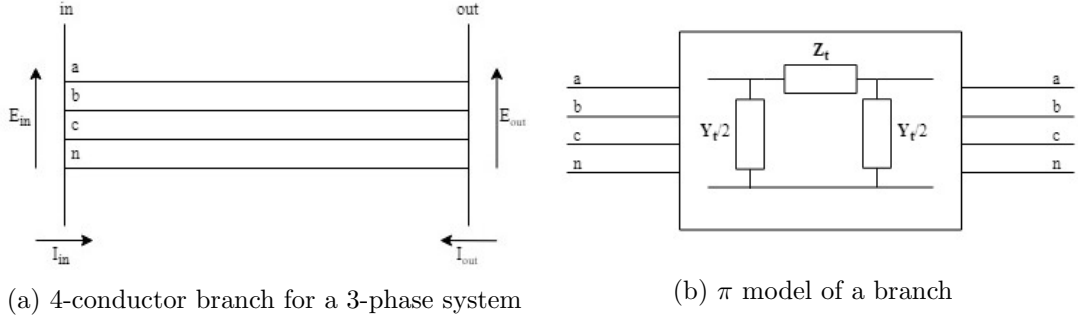


Figure 2.4: Network branch modelization

Using a π model to represent a m -conductor branch, it is possible to define the matrices for the longitudinal impedance $\dot{\mathbf{Z}}_1$ and for the transversal admittance $\dot{\mathbf{Y}}_t$ [1], as explained further in details in chapter 3. These matrices have a $m \times m$ dimension and represent self and mutual impedances and admittances for each conductor.

$$\dot{\mathbf{Z}}_1 = \begin{bmatrix} \dot{Z}_{aa} & \dot{Z}_{ab} & \dot{Z}_{ac} & \dot{Z}_{an} \\ \dot{Z}_{ba} & \dot{Z}_{bb} & \dot{Z}_{bc} & \dot{Z}_{bn} \\ \dot{Z}_{ca} & \dot{Z}_{cb} & \dot{Z}_{cc} & \dot{Z}_{cn} \\ \dot{Z}_{na} & \dot{Z}_{nb} & \dot{Z}_{nc} & \dot{Z}_{nn} \end{bmatrix} \quad \dot{\mathbf{Y}}_t = \begin{bmatrix} \dot{Y}_{aa} & \dot{Y}_{ab} & \dot{Y}_{ac} & \dot{Y}_{an} \\ \dot{Y}_{ba} & \dot{Y}_{bb} & \dot{Y}_{bc} & \dot{Y}_{bn} \\ \dot{Y}_{ca} & \dot{Y}_{cb} & \dot{Y}_{cc} & \dot{Y}_{cn} \\ \dot{Y}_{na} & \dot{Y}_{nb} & \dot{Y}_{nc} & \dot{Y}_{nn} \end{bmatrix} \quad (2.41)$$

Now, the $2m \times 2m$ branch admittance matrix $\dot{\mathbf{Y}}_{br}$ can be defined as:

$$\dot{\mathbf{Y}}_{br} = \begin{bmatrix} \dot{\mathbf{Z}}_1^{-1} + \frac{\dot{\mathbf{Y}}_t}{2} & -\dot{\mathbf{Z}}_1^{-1} \\ -\dot{\mathbf{Z}}_1^{-1} & \dot{\mathbf{Z}}_1^{-1} + \frac{\dot{\mathbf{Y}}_t}{2} \end{bmatrix} \quad (2.42)$$

From all the $\dot{\mathbf{Y}}_{br}$ calculated for each branch of the system a new $(2m \cdot N_{br}) \times (2m \cdot N_{br})$ matrix called $\dot{\mathbf{Y}}_{network}$ can be built, representing the whole network, where N_{br} is the total number of branches.

Voltages can be obtained from the load flow solution and then reorganized in a vector \mathbf{e} where voltages at the sending end of each branch are put first and last are the voltages at the receiving ends, as follows:

$$\bar{\mathbf{e}} = \begin{bmatrix} \bar{\mathbf{e}}_{\text{in}} \\ \bar{\mathbf{E}}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \bar{E}_{in,1}^a \\ \bar{E}_{in,1}^b \\ \vdots \\ \bar{E}_{in,N_{br}}^c \\ \bar{E}_{in,N_{br}}^n \\ \bar{E}_{out,1}^a \\ \bar{E}_{out,1}^b \\ \vdots \\ \bar{E}_{out,N_{br}}^c \\ \bar{E}_{out,N_{br}}^n \end{bmatrix} \quad (2.43)$$

Currents at sending and receiving end of each branch can then be calculated:

$$\bar{\mathbf{i}} = \begin{bmatrix} \bar{\mathbf{i}}_{\text{in}} \\ \bar{\mathbf{i}}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \bar{I}_{in,1}^a \\ \bar{I}_{in,1}^b \\ \vdots \\ \bar{I}_{in,N_{br}}^c \\ \bar{I}_{in,N_{br}}^n \\ \bar{I}_{out,1}^a \\ \bar{I}_{out,1}^b \\ \vdots \\ \bar{I}_{out,N_{br}}^c \\ \bar{I}_{out,N_{br}}^n \end{bmatrix} = \dot{\mathbf{Y}}_{\text{network}} \cdot \bar{\mathbf{e}} \quad (2.44)$$

The dimension of $\bar{\mathbf{i}}$ is therefore still $(2m \cdot N_{br})$, as for $\bar{\mathbf{e}}$.

All the tools for loss allocation are now available. In order to understand the procedure, let us consider one load and one branch. With respect to the 4-node network of fig.2.1, considering $m = 4$ let us choose load $L3$ and the branch (1). The objective is to calculate the losses $\mathbf{l}_{1,L3}$ on branch (1) to load $L3$, which is connected at the end of branch (2) and absorbs the current \bar{I}_{L3} .

$$\mathbf{l}_{1,L3} = \begin{bmatrix} L_{1,L3}^a \\ L_{1,L3}^b \\ L_{1,L3}^c \\ L_{1,L3}^n \end{bmatrix} = \mathbb{R} \left\{ (\bar{\mathbf{e}}_{\text{in},1} \cdot \bar{\mathbf{i}}_{L3}) - (\bar{\mathbf{e}}_{\text{out},1} \cdot \bar{\mathbf{i}}_{L3}) \right\} \quad (2.45)$$

$$= \mathbb{R} \left\{ (\dot{\mathbf{Y}}_{\text{br},1}^{-1} \bar{\mathbf{i}}_{\text{in},1} \cdot \bar{\mathbf{i}}_{L3}) - (\dot{\mathbf{Y}}_{\text{br},1}^{-1} \bar{\mathbf{i}}_{\text{out},1} \cdot \bar{\mathbf{i}}_{L3}) \right\} \quad (2.46)$$

Losses $\mathbf{l}_{1,L3}$ represent the effects of the load $L3$ on branch (1). Just one current at a time is being used, yet the mutual effects of all currents and cross-terms are taken into account, since the voltages are being calculated in the load flow with all the loads and generators together. If eq.2.45 is repeated for all the other branches in which $\bar{\mathbf{i}}_{L3}$ flows, i.e. branch (2), a loss allocation factor α_{L3} can be assigned to each phase of $L3$ for that specific load flow solution.

$$\mathbf{l}_{2,L3} = \begin{bmatrix} L_{2,L3}^a \\ L_{2,L3}^b \\ L_{2,L3}^c \\ L_{2,L3}^n \end{bmatrix} = \mathbb{R} \left\{ (\bar{\mathbf{e}}_{\text{in},2} \cdot \bar{\mathbf{i}}_{L3}) - (\bar{\mathbf{e}}_{\text{out},2} \cdot \bar{\mathbf{i}}_{L3}) \right\} \quad (2.47)$$

$$\begin{cases} \alpha_{L3}^a = L_{1,L3}^a + L_{2,L3}^a \\ \alpha_{L3}^b = L_{1,L3}^b + L_{2,L3}^b \\ \alpha_{L3}^c = L_{1,L3}^c + L_{2,L3}^c \\ \alpha_{L3}^n = L_{1,L3}^n + L_{2,L3}^n \end{cases} \quad (2.48)$$

To perform this calculation on the whole network, voltages can be rearranged in a $m \times (2 \cdot N_{br})$ matrix (eq.2.49) that is then repeated vertically N_{br} times in a $(m \cdot N_{br}) \times (2 \cdot N_{br})$ matrix called $\bar{\mathbf{E}}_{\text{mat}}$ as in 2.50. It has to be noted that losses are going to be allocated to the loads and the generators separately at each node. Thus, when more loads are connected to the same node, losses are going to be allocated as a whole to the node, then proportionally divided between loads.

$$\bar{\mathbf{E}}_{\text{reshaped}} = \begin{bmatrix} \bar{E}_{in,1}^a & \bar{E}_{out,1}^a & \cdots & \bar{E}_{out,N_{br}}^a \\ \bar{E}_{in,1}^b & \bar{E}_{out,1}^b & \cdots & \bar{E}_{out,N_{br}}^b \\ \bar{E}_{in,1}^c & \bar{E}_{out,1}^c & \cdots & \bar{E}_{out,N_{br}}^c \\ \bar{E}_{in,1}^n & \bar{E}_{out,1}^n & \cdots & \bar{E}_{out,N_{br}}^n \end{bmatrix} \quad (2.49)$$

$$\bar{\mathbf{E}}_{\text{mat}} = \begin{bmatrix} \mathbf{E}_{\text{reshaped}} \\ \mathbf{E}_{\text{reshaped}} \\ \vdots \\ \mathbf{E}_{\text{reshaped}} \end{bmatrix} \quad (2.50)$$

Loads and generators currents can be obtain either from the power flow or as net output currents at each node (difference between branch currents going in the node and going out the node). This last option is available only if shunt elements of the branches can be neglected, which is usually true in low voltage networks: currents flowing through shunt capacitance are very small, so that the current entering a line can be assumed to be equal to the one exiting.

Loads and generators currents are then stored in a $(m \cdot N_{br})$ vector where each group of m rows represent the currents absorbed/injected by the loads/generators at the receiving end of the branch. In the example of a 4-node network, this would mean that the first 4 elements of the vector are the 4 currents (one for each conductor) absorbed/injected by the loads/generators at the receiving end of the branch (1), i.e. node 2. Elements from 5 to 8 would be currents absorbed/injected in node 3 and elements from 9 to 12 are relative to node 4.

$$\bar{\mathbf{i}}_{\text{loads}} = \begin{bmatrix} \bar{I}_{load,1}^a \\ \bar{I}_{load,1}^b \\ \bar{I}_{load,1}^c \\ \bar{I}_{load,1}^n \\ \vdots \\ \bar{I}_{load,N_{br}}^a \\ \bar{I}_{load,N_{br}}^b \\ \bar{I}_{load,N_{br}}^c \\ \bar{I}_{load,N_{br}}^n \end{bmatrix} \quad \bar{\mathbf{i}}_{\text{gens}} = \begin{bmatrix} \bar{I}_{gens,1}^a \\ \bar{I}_{gens,1}^b \\ \bar{I}_{gens,1}^c \\ \bar{I}_{gens,1}^n \\ \vdots \\ \bar{I}_{gens,N_{br}}^a \\ \bar{I}_{gens,N_{br}}^b \\ \bar{I}_{gens,N_{br}}^c \\ \bar{I}_{gens,N_{br}}^n \end{bmatrix} \quad (2.51)$$

Two new matrices, $\mathbf{I}_{\text{mat,loads}}$ and $\mathbf{I}_{\text{mat,gens}}$, are to be built repeating horizontally $\bar{\mathbf{i}}_{\text{loads}}$ and $\bar{\mathbf{i}}_{\text{gens}}$ ($2 \cdot N_{br}$) times. They have the same dimension as $\bar{\mathbf{E}}_{\text{mat}}$. Grouping together the elements of the current matrices by $m \times 2$, they can be seen as square

matrices: all the elements of above the north-west diagonal have to be zeroed, in order to consider topology of the grid. For clarity's sake, eq.2.52 and 2.53 show the current matrices for the 4-nodes network.

$$\mathbf{I}_{\text{mat,loads}} = \begin{bmatrix} \bar{I}_{load,1}^a & \bar{I}_{load,1}^a & 0 & 0 & 0 & 0 \\ \bar{I}_{load,1}^b & \bar{I}_{load,1}^b & 0 & 0 & 0 & 0 \\ \bar{I}_{load,1}^c & \bar{I}_{load,1}^c & 0 & 0 & 0 & 0 \\ \bar{I}_{load,1}^n & \bar{I}_{load,1}^n & 0 & 0 & 0 & 0 \\ \bar{I}_{load,2}^a & \bar{I}_{load,2}^a & \bar{I}_{load,2}^a & \bar{I}_{load,2}^a & 0 & 0 \\ \bar{I}_{load,2}^b & \bar{I}_{load,2}^b & \bar{I}_{load,2}^b & \bar{I}_{load,2}^b & 0 & 0 \\ \bar{I}_{load,2}^c & \bar{I}_{load,2}^c & \bar{I}_{load,2}^c & \bar{I}_{load,2}^c & 0 & 0 \\ \bar{I}_{load,2}^n & \bar{I}_{load,2}^n & \bar{I}_{load,2}^n & \bar{I}_{load,2}^n & 0 & 0 \\ \bar{I}_{load,3}^a & \bar{I}_{load,3}^a & \bar{I}_{load,3}^a & \bar{I}_{load,3}^a & \bar{I}_{load,3}^a & \bar{I}_{load,3}^a \\ \bar{I}_{load,3}^b & \bar{I}_{load,3}^b & \bar{I}_{load,3}^b & \bar{I}_{load,3}^b & \bar{I}_{load,3}^b & \bar{I}_{load,3}^b \\ \bar{I}_{load,3}^c & \bar{I}_{load,3}^c & \bar{I}_{load,3}^c & \bar{I}_{load,3}^c & \bar{I}_{load,3}^c & \bar{I}_{load,3}^c \\ \bar{I}_{load,3}^n & \bar{I}_{load,3}^n & \bar{I}_{load,3}^n & \bar{I}_{load,3}^n & \bar{I}_{load,3}^n & \bar{I}_{load,3}^n \end{bmatrix} \quad (2.52)$$

$$\mathbf{I}_{\text{mat,gens}} = \begin{bmatrix} \bar{I}_{gen,1}^a & \bar{I}_{gen,1}^a & 0 & 0 & 0 & 0 \\ \bar{I}_{gen,1}^b & \bar{I}_{gen,1}^b & 0 & 0 & 0 & 0 \\ \bar{I}_{gen,1}^c & \bar{I}_{gen,1}^c & 0 & 0 & 0 & 0 \\ \bar{I}_{gen,1}^n & \bar{I}_{gen,1}^n & 0 & 0 & 0 & 0 \\ \bar{I}_{gen,2}^a & \bar{I}_{gen,2}^a & \bar{I}_{gen,2}^a & \bar{I}_{gen,2}^a & 0 & 0 \\ \bar{I}_{gen,2}^b & \bar{I}_{gen,2}^b & \bar{I}_{gen,2}^b & \bar{I}_{gen,2}^b & 0 & 0 \\ \bar{I}_{gen,2}^c & \bar{I}_{gen,2}^c & \bar{I}_{gen,2}^c & \bar{I}_{gen,2}^c & 0 & 0 \\ \bar{I}_{gen,2}^n & \bar{I}_{gen,2}^n & \bar{I}_{gen,2}^n & \bar{I}_{gen,2}^n & 0 & 0 \\ \bar{I}_{gen,3}^a & \bar{I}_{gen,3}^a & \bar{I}_{gen,3}^a & \bar{I}_{gen,3}^a & \bar{I}_{gen,3}^a & \bar{I}_{gen,3}^a \\ \bar{I}_{gen,3}^b & \bar{I}_{gen,3}^b & \bar{I}_{gen,3}^b & \bar{I}_{gen,3}^b & \bar{I}_{gen,3}^b & \bar{I}_{gen,3}^b \\ \bar{I}_{gen,3}^c & \bar{I}_{gen,3}^c & \bar{I}_{gen,3}^c & \bar{I}_{gen,3}^c & \bar{I}_{gen,3}^c & \bar{I}_{gen,3}^c \\ \bar{I}_{gen,3}^n & \bar{I}_{gen,3}^n & \bar{I}_{gen,3}^n & \bar{I}_{gen,3}^n & \bar{I}_{gen,3}^n & \bar{I}_{gen,3}^n \end{bmatrix} \quad (2.53)$$

From this point on, the discussion of the method will be done considering only loads, since for the generators the exact same principle should be applied. The component multiplication of $\bar{\mathbf{E}}_{\text{mat}}$ with $\mathbf{I}_{\text{mat,loads}}$ gives nodal powers due to each shunt current absorbed at each node.

$$\bar{\mathbf{S}}_{\text{nodal}} = \begin{bmatrix} \bar{E}_{in,1}^a \bar{I}_{load,1}^{a*} & \bar{E}_{out,1}^a \bar{I}_{load,1}^{a*} & 0 & 0 & \dots & 0 \\ \bar{E}_{in,1}^b \bar{I}_{load,1}^{b*} & \bar{E}_{out,1}^b \bar{I}_{load,1}^{b*} & 0 & 0 & \dots & 0 \\ \bar{E}_{in,1}^c \bar{I}_{load,1}^{c*} & \bar{E}_{out,1}^c \bar{I}_{load,1}^{c*} & 0 & 0 & \dots & 0 \\ \bar{E}_{in,1}^n \bar{I}_{load,1}^{n*} & \bar{E}_{out,1}^n \bar{I}_{load,1}^{n*} & 0 & 0 & \dots & 0 \\ \bar{E}_{in,1}^a \bar{I}_{load,2}^{a*} & \bar{E}_{out,1}^a \bar{I}_{load,2}^{a*} & \bar{E}_{in,2}^a \bar{I}_{load,2}^{a*} & \bar{E}_{out,2}^a \bar{I}_{load,2}^{a*} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{E}_{in,N_1}^a \bar{I}_{load,N_{br}}^{a*} & \bar{E}_{out,1}^a \bar{I}_{load,N_{br}}^{a*} & \bar{E}_{in,2}^a \bar{I}_{load,N_{br}}^{a*} & \bar{E}_{out,2}^a \bar{I}_{load,N_{br}}^{a*} & \dots & \bar{E}_{out,N_{br}}^a \bar{I}_{load,N_{br}}^{a*} \end{bmatrix} \quad (2.54)$$

The difference between two consecutive columns (1-2,3-4 and so on) gives as result the voltage drop on a specific branch times the conjugate of the current absorbed at a given node, which is the power loss on a branch due to a load.

$$\bar{\mathbf{S}}_{\text{branch}} = \begin{bmatrix} \bar{S}_{br1,1}^a & 0 & \dots & 0 \\ \bar{S}_{br1,1}^b & 0 & \dots & 0 \\ \bar{S}_{br1,1}^c & 0 & \dots & 0 \\ \bar{S}_{br1,1}^n & 0 & \dots & 0 \\ \bar{S}_{br1,2}^a & \bar{S}_{br2,2}^a & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \bar{S}_{br1,N_{br}}^n & \bar{S}_{br2,N_{br}}^n & \dots & \bar{S}_{brN_{br},N_{br}}^n \end{bmatrix} \quad (2.55)$$

$$\bar{S}_{br1,1}^a = \bar{E}_{in,1}^a \bar{I}_{load,1}^{a*} - \bar{E}_{out,1}^a \bar{I}_{load,1}^{a*} \quad (2.56)$$

$\bar{\mathbf{S}}_{\text{branch}}$ is a $m \cdot N_{br} \times N_{br}$ matrix that has to be read as follow: each column indicates the branch in which the loss is happening (first index of the subscript), each row is the branch and phase at the receiving end of which the load responsible for that loss is placed (second index of the subscript). So, for instance, $\bar{S}_{br2,N_{br}}^b$ is the power loss on phase b of branch (2) due to the load connected on the same phase at the end of branch (N_{br}). This means that if the elements of $\bar{\mathbf{S}}_{\text{branch}}$ were to be summed up horizontally, all the elements of a row would represent total losses on the network generated by the load identified by that row (eq.2.57). If the elements of $\bar{\mathbf{S}}_{\text{branch}}$ were to be summed up vertically, all the elements of a column would represent total losses on the branch that the column identifies (eq.2.58).

$$\bar{\mathbf{L}}_{\text{alloc}} = \begin{bmatrix} \bar{S}_{br1,1}^a \\ \bar{S}_{br1,1}^b \\ \vdots \\ \bar{S}_{br1,N_{br}}^n + \bar{S}_{br2,N_{br}}^n + \dots + \bar{S}_{brN_{br},N_{br}}^n \end{bmatrix} \quad (2.57)$$

$$\bar{\mathbf{L}}_{\text{br}} = \begin{bmatrix} \bar{S}_{br1,1}^a + \bar{S}_{br1,2}^a + \dots + \bar{S}_{br1,N_{br}}^a \\ \bar{S}_{br1,1}^a + \bar{S}_{br1,2}^a + \dots + \bar{S}_{br1,N_{br}}^a \\ \vdots \\ \bar{S}_{br1,N_{br}}^n \end{bmatrix} \quad (2.58)$$

The real part of the elements of $\bar{\mathbf{L}}_{\text{alloc}}$ are the allocation terms for all loads, which can then be reshaped in a more practical order and post-processed if needed.

2.5.2 Considerations

ABCLA has the advantage, compared to the other methods, that loads and generators currents are processed individually. This allows for a deeper understanding of the grid operations and of the effect of DG on the network in a way the other loss allocation procedures cannot do.

As presented, the method works well on the 4-nodes network used as example. It has to be mentioned though that, as it is, the algorithm do not work in a multi-feeder environment, where from a node multiple branches start. For now, the network has to be radial and well organised, branch numeration has to start from the root node towards the terminal branch.

One or multiple transformers can be added to the network and be treated just as any other branch, as long as their group is 0 and the winding configuration on the primary side matches the one on the secondary. Phase shift and different winding configuration need to be taken into account separately.

These issues will be tackled in chapter 3, where the implementation of ABCLA in a specific multi-conductor environment creates, modification after modification, to the genesis of a procedure, called Multi-Phase and Neutral conductors Loss Allocation (MPNLA).

MPNLA does not directly use neither admittances matrices (with one exception) nor branch current, hence the name variation. Its inputs are nodal voltages and nodal currents absorbed/injected by the loads/generators. This procedure allows for both complex grid topologies and transformers presence, whilst being faster by avoiding redundant calculations.

Thus, from now on only the modified version, MPNLA, will be discussed, starting from its genesis (chapter 3) to its application (chapter 4), highlighting also key features and innovations brought by it (chapter 5).

Chapter 3

Implementation of MPNLA method

MPNLA is the name given to the modified version of ABCLA. Although MPNLA process can work on the results of any load flow, to further discuss it the multi-conductor power flow method used in this work will be now presented. Also, the way MPNLA is being implemented in the context of said methodology and the modifications applied to the original method will be discussed step by step.

3.1 Correction Current Injection method for power flow analysis of multi-conductor networks

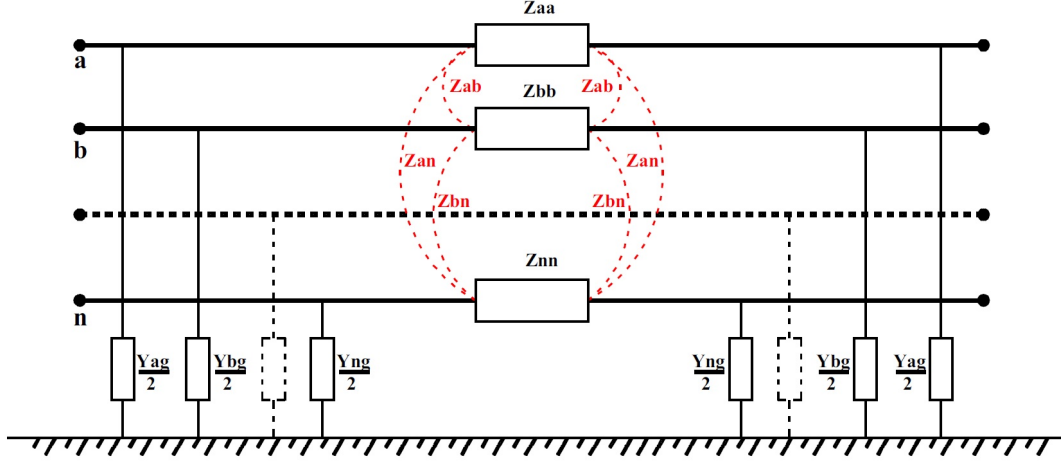
As previously mentioned, most of commercial softwares use traditional Newton-Raphson or Gauss-Seidel based load flow procedures and perform well in transmission networks, yet they have some limitations. Convergence for the algorithm is achieved with more difficulties when the system has a high R/X ratio or negligible shunt elements, plus results are lacking in precision when the system is highly unbalanced.

Therefore the methodology used promises robustness and high accuracy, even in systems where conditions are not optimal for classical power flow algorithms. In addition to that, the procedure can be implemented in Matlab without the need for a commercial, proprietary software.

Correction Current Injection (CCI) method is based on the novel idea to represent all loads and generators (with except for the slack bus) of the network as shunt elements which are then incorporated in the nodal admittance matrix. Having a fixed matrix representing the whole grid drastically reduces the number of iterations required to get to convergence when compared to traditional methods [14]. The system can be seen as inert, excited only on the slack bus by an applied voltage phasor. Correction currents are then injected in parallel to each shunt element to adjust the value of shunt admittances to match instant operating conditions of the network. CCI allows for a robust, fast and precise multi-conductor analysis that takes into consideration earthing, asymmetries and coupling between conductors.

3.1.1 Method description

In order to describe CCI algorithm and how it works, the first point is to create the admittance matrix for the whole m -conductors system. Starting from the branches,

Figure 3.1: π -model of an n -conductor branch element [1]

each one can be modelled as $2m \times 2m$ matrix $\dot{\mathbf{Y}}_{br}$ with a π -model.

$$\dot{\mathbf{Y}}_{br} = \begin{bmatrix} \dot{\mathbf{Z}}_1^{-1} + \frac{\dot{\mathbf{Y}}_t}{2} & -\dot{\mathbf{Z}}_1^{-1} \\ -\dot{\mathbf{Z}}_1^{-1} & \dot{\mathbf{Z}}_1^{-1} + \frac{\dot{\mathbf{Y}}_t}{2} \end{bmatrix} \quad (3.1)$$

Submatrices $\dot{\mathbf{Z}}_1$ and $\dot{\mathbf{Y}}_t$ are have $m \times m$ dimension and are calculated through Carson-Clem equations and Maxwell's potential coefficients respectively.

$$\dot{\mathbf{Z}}_1 = \begin{bmatrix} \dot{Z}_{aa} & \dot{Z}_{ab} & \dots & \dot{Z}_{an} \\ \dot{Z}_{ba} & \dot{Z}_{bb} & \dots & \dot{Z}_{bn} \\ \vdots & \vdots & \ddots & \vdots \\ \dot{Z}_{na} & \dot{Z}_{nb} & \dots & \dot{Z}_{nn} \end{bmatrix} \quad \dot{\mathbf{Y}}_t = \begin{bmatrix} \dot{Y}_{aa} & \dot{Y}_{ab} & \dots & \dot{Y}_{an} \\ \dot{Y}_{ba} & \dot{Y}_{bb} & \dots & \dot{Y}_{bn} \\ \vdots & \vdots & \ddots & \vdots \\ \dot{Y}_{na} & \dot{Y}_{nb} & \dots & \dot{Y}_{nn} \end{bmatrix} \quad (3.2)$$

$$\bar{Z}_{ii} = R_i + R_{earth} + j\omega \cdot 2 \cdot 10^{-4} \cdot \ln\left(\frac{D_e}{r_i}\right) \left[\frac{\Omega}{km}\right] \quad (3.3)$$

$$\bar{Z}_{ij} = R_{earth} + j\omega \cdot 2 \cdot 10^{-4} \cdot \ln\left(\frac{D_e}{d_{ij}}\right) \left[\frac{\Omega}{km}\right] \quad (3.4)$$

where R_i : DC resistance $[\Omega/km]$

r_i : phase conductor radius $[m]$

d_{ij} : mutual distance between conductors i and j $[m]$

$R_{earth} = \pi^2 \cdot f \cdot 10^{-4} [\Omega/km]$

$D_e = 659 \sqrt{\frac{\rho}{f}} [m]$

The characteristics of the return path through ground depend on the soil conductivity ρ (typically $100\Omega m$) and system frequency f . R_{earth} is a frequency-dependent resistance, since the distribution of AC current in the ground is not uniform.

The elements of $\dot{\mathbf{Y}}_t$ represent capacitive self and mutual susceptance and conductance. They are evaluated through Maxwell's potential coefficients but overall for a low voltage network their weight is just marginal.

Through the computation of an incidence matrix that defines the topology of the whole grid, with all branches $\dot{\mathbf{Y}}_{br}$ matrices, the system nodal admittance matrix $\dot{\mathbf{Y}}_{network}$ can be built. In a N -bus network, $\dot{\mathbf{Y}}_{network}$ is a square matrix with $(N \cdot m) \times (N \cdot m)$ dimension.

Another matrix $\dot{\mathbf{Y}}_{trans}$ of same dimensions can be built to take into account transformers on the grid. These are modelled considering transformation ratio, windings configuration, power, short circuit and off-load values, group number and even tap changer systems, both off-load and on-load, with line drop compensation.

Since this power flow method allows customizable connections between conductors and ground at each node, a matrix $\dot{\mathbf{Y}}_{ground}$ can be created where diagonal terms represent self-impedances between conductors and ground. For instance, in a three-phase network, neutral conductor at each bus can be insulated, connected to ground through tuned impedance or directly grounded, in accordance with IEC standards IT, TT, TNC and TNS [15].

Finally, a matrix called $\dot{\mathbf{Y}}_{pass}$ is generated to represent the whole passive network, without any loads or generators:

$$\dot{\mathbf{Y}}_{pass} = \dot{\mathbf{Y}}_{network} + \dot{\mathbf{Y}}_{trans} + \dot{\mathbf{Y}}_{ground} \quad (3.5)$$

Constant shunt elements representing loads and generators at nominal conditions can now be modelled. For a single-phase element connected between nodes k and h (where the nodes belong to the same bus):

$$\dot{Y}_{kh} = \frac{\bar{S}_{kh(0)}^*}{|\bar{U}_{kh(0)}|^2} \quad \text{where } (\bar{U}_{kh} = \bar{E}_k - \bar{E}_h) \quad (3.6)$$

$S_{kh(0)}$ is the rated power of the element, E are nodal potential and the subscript (0) indicates nominal values. Later on in the process, when the shunt elements are not operation under nominal conditions, correction currents are locally injected in parallel to adjust voltages and powers. This modelization enables the loads and generators matrices to be included in the network matrix. The constant shunt admittance allows for the element to be described power-wise as follow:

$$\bar{S}_{kh(i)}^* = \dot{Y}_{kh} \cdot |\bar{U}_{kh(i)}|^2 - \bar{U}_{kh(i)} \Delta \bar{I}_{kh(i)} \quad (3.7)$$

Eq.3.7 describe the shunt element power at its i -th iteration. A ZIP model can be used to highlight the correction approach.

$$\bar{S}_{kh(i)}^* = \bar{S}_{kh(0)}^* \left[\left(\frac{|\bar{U}_{kh(i)}|}{|\bar{U}_{kh(0)}|} \right)^2 k_Z + \left(\frac{|\bar{U}_{kh(i)}|}{|\bar{U}_{kh(0)}|} \right) k_I + k_P \right] \quad (3.8)$$

$$k_Z + k_I + k_P = 1$$

As eq.3.8 shows, the ZIP coefficients model represents the variation, with respect to voltage, of a load as a composition of three types of constant loads which have, respectively, fixed impedance, current and power: hence the three coefficients [16].

Therefore eq.3.7 can be rewritten into eq.3.9, as proven in [1].

$$\bar{S}_{kh(i)}^* = \dot{Y}_{kh} \cdot |\bar{U}_{kh(i)}|^2 - (\Delta \bar{I}_{kh(i)I\%} + \Delta \bar{I}_{kh(i)P\%}) \bar{U}_{kh(i)} \quad (3.9)$$

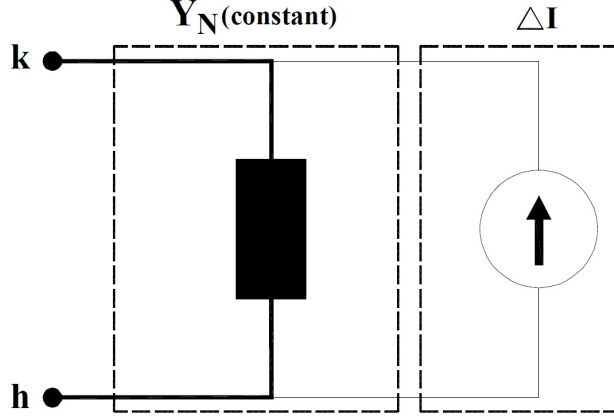


Figure 3.2: Shunt element representation where the impedance, current and power correction currents are grouped into $\Delta \mathbf{I}$ [2]

To be consistent with the multi-conductor analysis, three-phase elements, both wye and delta connected, are modelled as single-phase equivalents.

At this point, for a given instant in time, the constant admittance portion of the shunt elements is calculated and added to $\dot{\mathbf{Y}}_{\text{pass}}$, so that now the system buses are merely ports where shunt correction currents can be injected in an iterative process: their value is updated until the solution of the power flow match the target within a given margin of error. It has to be noted that the inclusion of shunt elements constant admittances in the system admittance matrix has the positive side effect of bettering the inversion of said matrix, lowering the risk of incurring into singularity problems.

CCI approach specifically isolates the slack bus, which has imposed voltage vector $\bar{\mathbf{e}}_{\text{SL}}$, from the other shunt elements.

$$\begin{bmatrix} \bar{\mathbf{i}}_{\text{SL}_{a\dots m}} \\ \Delta \bar{\mathbf{i}}_{2_{a\dots m}} \\ \vdots \\ \Delta \bar{\mathbf{i}}_{\text{N}_{a\dots m}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{Y}}_{\text{SL-SL}} & \dot{\mathbf{Y}}_{\text{SL-Sh}} \\ \dot{\mathbf{Y}}_{\text{Sh-SL}} & \dot{\mathbf{Y}}_{\text{Sh-Sh}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{e}}_{\text{SL}_{a\dots m}} \\ \bar{\mathbf{e}}_{2_{a\dots m}} \\ \vdots \\ \bar{\mathbf{e}}_{\text{N}_{a\dots m}} \end{bmatrix} \quad (3.10)$$

To find the solution to the system represented by eq.3.10, an iterative process start:

$$\bar{\mathbf{e}}_{2 \div \text{N}} = \dot{\mathbf{Y}}_{\text{Sh-Sh}}^{-1} (\Delta \bar{\mathbf{i}}_{2 \div \text{N}} - \dot{\mathbf{Y}}_{\text{Sh-SL}} \bar{\mathbf{e}}_{\text{SL}}) \quad (3.11)$$

Initial condition is calculated by zeroing all $\bar{\Delta i}$ elements. Convergence is reached at the i -th iteration when, given error thresholds for both voltage magnitude and phase shift:

$$|\bar{E}_{(i+1)}| - |\bar{E}_{(i)}| \leq \varepsilon_M \quad (3.12)$$

$$\angle \bar{E}_{(i+1)} - \angle \bar{E}_{(i)} \leq \varepsilon_A \quad (3.13)$$

This way, admittance matrix is never updated in the process, so that matrix inversion is simpler and needs to be performed just once.

When the iterations are completed, voltages across the whole network are known within margin of error. Thus, currents flowing in each branch can be calculated, as well as shunt currents absorbed/injected at each bus by loads/generators, losses in each

branch and every other parameter that can be useful to understand the operation of the grid. As previously mentioned, the procedure is completely open, it was implemented in Matlab and this allows the user to modified it and configure it to suit its needs. For instance, in the process of implementing MPNLA method it was clear that two parameters where needed and was initially not an explicit output of the software: loads and generators currents ($\bar{\mathbf{I}}_{\text{Loads}}$ and $\bar{\mathbf{I}}_{\text{Gen}}$ respectively) and the transformers admittance matrices $\bar{\mathbf{Y}}_{\text{tr}}$. As shown in Appendix A, this was easily solved adding custom code lines to the software.

3.2 Implementation of MPNLA method

The full Matlab code written by the author to implement MPNLA method, described in section 2.5, into the CCI power flow environment (in particular, the one developed by research group in University of Padova), is available in Appendix A.

The objective of the procedure is to effectively implement the method whilst writing a code that will run smoothly, correctly and fast. To ensure that, two different, very simple networks, here presented, were designed and the script was constantly tested throughout the coding to highlight flaws or malfunctionings.

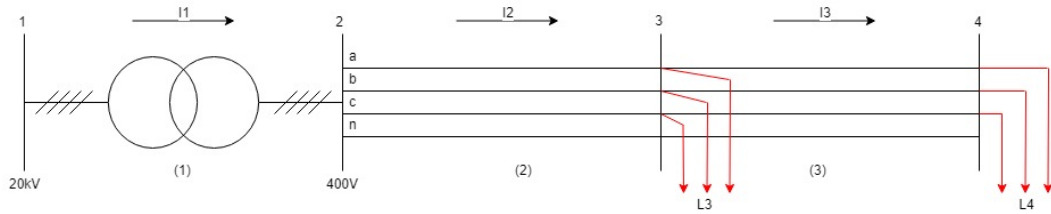


Figure 3.3: Passive 4-node radial distribution network with balanced loads

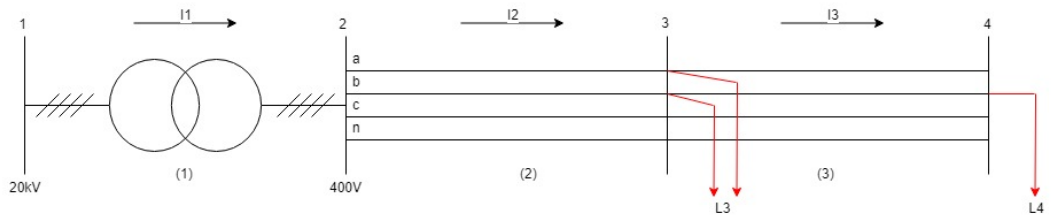


Figure 3.4: Passive 4-node radial distribution network with highly unbalanced loads

The networks share the following parameters:

- Transformer has always a Y-configuration on the secondary side (bus 2) and its star-point is directly grounded. Thus, the neutral conductor is single-grounded in bus 2. Nominal values for the transformer are: $V_1=20$ [kV], $V_2=400$ [V], $S_{nom}=50$ [kVA], $v_{cc}=4\%$, $p_{cc}=1.47\%$, $p_0=0\%$, $i_0=0\%$ (allocation of transformer magnetization losses will be discussed in chapter 5, for now let us assume them to be none). Transformer connections and group will be defined for each test.
- Each branch is an overhead line with a length of 100m, composed by three phase wires (with diameter equal to 18.3134 [mm], GMR coefficient of 0.812205,

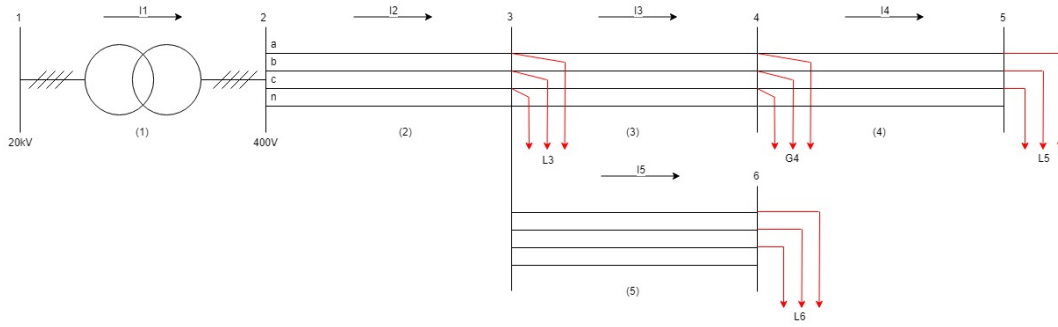


Figure 3.5: Active 6-node radial distribution network, diramated, with balanced loads and generators

$R=0.190139$ [Ω]) and one neutral wire (with diameter equal to 14.3002 [mm], GMR coefficient of 0.346998, $R=0.367851$ [Ω]). The mutual position of the wires is given in fig.3.6. This configuration corresponds is one of the IEEE branch variant used in its test network [17].

- Loads, when no further specified, have to be defined as $P=3$ [kW], $\cos\varphi=0.9$ for the three-phase case, $P=1$ [kW], $\cos\varphi=0.9$ for the single-phase. Generators, when non further specified, have to be defined as $P=3$ [kW], $\cos\varphi=1$ for the three-phase case, $P=1$ [kW], $\cos\varphi=1$ for the single-phase, with no voltage regulation or any type of control.

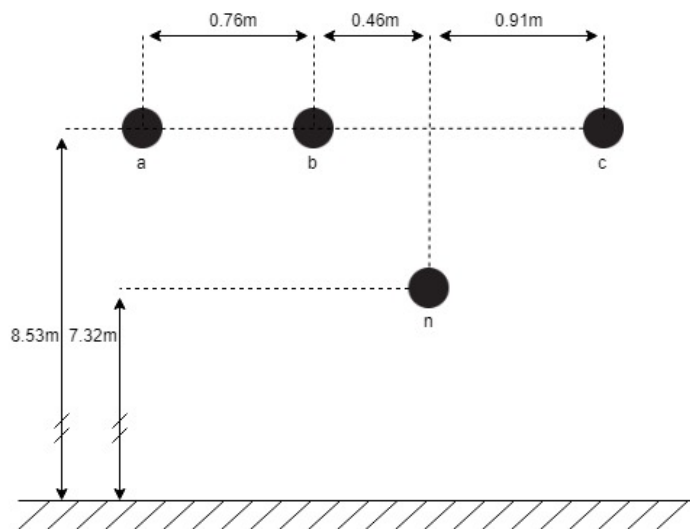


Figure 3.6: Position of the wires along the cross section of the line for fig.3.3, 3.4 and 3.5 networks

3.2.1 Input values

In order to perform the loss allocation, let us point out which results of the power flow are needed and how they should be arranged. MPNLA can work using just the solutions of the power flow (nodal voltages and shunt nodal currents), without the need for the branch admittance matrix nor current, as shown in eq.2.45. This feature makes the loss

allocation algorithm more robust and quicker at the same time, since the calculation of loads/generators absorbed/injected currents are performed within the load flow.

Needed inputs are general information about the grid topology: number of conductors m , number of buses, total number of transformer and branches N_{br} and their connections to buses and so on. All calculations are done in per unit, to guarantee seamless operations where multiple voltage levels are present in the same network. At the end of the procedure results can be multiply by their base values to return into S.I. units.

Voltages at each node are known from the load flow solutions: they are stored in a matrix in which rows are organized firstly by increasing bus number and secondly by conductor, whilst columns are sorted by time instances.

Loads and generators currents are calculated in the power flow, since each element has been modelled as in fig.3.2, as follows:

$$\bar{\mathbf{i}}_{\text{loads}} = \begin{bmatrix} \bar{I}_{load,1}^a \\ \bar{I}_{load,1}^b \\ \bar{I}_{load,1}^c \\ \bar{I}_{load,1}^n \\ \vdots \\ \bar{I}_{load,N_{br}}^a \\ \bar{I}_{load,N_{br}}^b \\ \bar{I}_{load,N_{br}}^c \\ \bar{I}_{load,N_{br}}^n \end{bmatrix} = \dot{\mathbf{Y}}_{\text{loads}} \cdot \bar{\mathbf{e}} - \Delta \bar{\mathbf{i}}_{\text{loads}} \quad (3.14)$$

$$\bar{\mathbf{i}}_{\text{gens}} = \begin{bmatrix} \bar{I}_{gens,1}^a \\ \bar{I}_{gens,1}^b \\ \bar{I}_{gens,1}^c \\ \bar{I}_{gens,1}^n \\ \vdots \\ \bar{I}_{gens,N_{br}}^a \\ \bar{I}_{gens,N_{br}}^b \\ \bar{I}_{gens,N_{br}}^c \\ \bar{I}_{gens,N_{br}}^n \end{bmatrix} = -\dot{\mathbf{Y}}_{\text{gens}} \cdot \bar{\mathbf{e}} + \Delta \bar{\mathbf{i}}_{\text{gens}} \quad (3.15)$$

3.2.2 Algorithm implementation and validation

As it can be seen in Appendix A, firstly voltage and current vectors have to be reshaped: for each instant in time t that has been analysed, $\bar{\mathbf{E}}_{\text{mat}}$, $\bar{\mathbf{I}}_{\text{mat,loads}}$ and $\bar{\mathbf{I}}_{\text{mat,gens}}$ will be created. First of all, voltages are reshaped into potentials at sending and receiving ends of each branch using the topology information given. A $(2m \cdot N_{br}) \times t$ matrix called $\bar{\mathbf{E}}_{\text{br}}$ is built, where each column store the right information for a given instant t .

In a very similar way, $\bar{\mathbf{I}}_{\text{br,loads}}$ and $\bar{\mathbf{I}}_{\text{br,gens}}$ are created with dimension $(m \cdot N_{br}) \times t$, where each entry represents the current absorbed/injected by the shunt element connected at the receiving end (i.e. the furthest away from the root node, the one downstream of the line) of the branch and conductor that row refers to.

At this point, a "for" cycle is initiated, to iterate the process for each time instant separately. For a given time t , $\bar{\mathbf{E}}_{\text{mat}}$, $\bar{\mathbf{I}}_{\text{mat,loads}}$ and $\bar{\mathbf{I}}_{\text{mat,gens}}$ (described in eq.2.50, 2.52 and 2.53) are built from the values stored in the t -th column of \mathbf{E}_{br} , $\mathbf{I}_{\text{br,loads}}$ and $\mathbf{I}_{\text{br,gens}}$.

The matrix $\bar{\mathbf{S}}_{\text{nodal}}$ is obtained for both loads and generators through the multiplication of the voltage matrix with the conjugated of the current one. The matrix $\bar{\mathbf{S}}_{\text{branch}}$ is then easily calculated. The summation of the elements on the same row and column give respectively losses allocated (in complex form) and total losses on each branch. Results are then reshaped so that losses are allocated with respect to the bus number and conductor rather than to the branch which receiving end they are connected to. A structure called MPNLA is created to gather all results.

A first testing was performed on the balanced network of fig.3.3, with the transformer connected as Dyn11. The results showed two main issues: losses on the transformer were a couple orders of magnitude higher than expected and losses on the branches were not equal amongst phases. Since the load are balanced, small differences in losses between phases could be explained as due to the asymmetries in the grid (conductors layout and no phase transposition). Yet, the high losses on the transformer are an indication of a much larger problem. Testing the algorithm on the same network but with a Yyn0 transformer gives results that are somewhat expected and comparable to other losses calculation method. A second testing on the network of fig.3.4 showed that, despite phase c being unloaded, on the Dyn11 transformer, losses occurs on phase c , even if they should be null when looking from the secondary side. At this point it was clear that transformers cannot be treated equally as a branch.

No mention of such a problem was found in the analysed scientific literature, since very little publications analyse losses allocation from a multi-conductor point of view. ABCLA was never tested for losses allocation on transformers. The presence of both phase-shift and different winding configuration cannot be tackled with the same approach used for the branches: modelling a transformer as a branch works only on a single-phase system

Inside MPNLA a new procedure has then been developed, with the help of PhD student Mohammed Usman, in order to correctly allocate transformer losses and make MPNLA suitable for multi-conductor networks with multiple voltage levels. This require the use of the transformer admittance matrix calculated during the load flow (it is constant, non time-dependent), that takes into account winding configuration and group. The relation that model the transformer is the following:

$$\begin{aligned} \bar{\mathbf{i}} &= \dot{\mathbf{Y}} \bar{\mathbf{v}} \\ \begin{bmatrix} \bar{\mathbf{i}}_P \\ \bar{\mathbf{i}}_S \end{bmatrix} &= \begin{bmatrix} \dot{\mathbf{Y}}_{PP} & \dot{\mathbf{Y}}_{PS} \\ \dot{\mathbf{Y}}_{SP} & \dot{\mathbf{Y}}_{SS} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{v}}_P \\ \bar{\mathbf{v}}_S \end{bmatrix} \end{aligned} \quad (3.16)$$

where the subscripts P and S indicate respectively primary and secondary. The issue with using ABCLA for a transformer is that the method works by subtracting nodal powers at the ends of each element. For a generic multi-conductor (three-phase) transformer, currents from secondary to primary are being shifted in their angles and are mixed between phases, since phase current is typically different from the winding current.

MPNLA overcomes this issue by calculating transformer losses with respect to each load/generator current (absorbed/injected at the secondary side) while using just the admittance matrix and the secondary voltage and current values. From eq.3.16:

$$\bar{\mathbf{v}}_s = \dot{\mathbf{Y}}_{SS}^+ (\bar{\mathbf{i}}_s - \dot{\mathbf{Y}}_{SP} \bar{\mathbf{v}}_P) \quad (3.17)$$

$$\bar{\mathbf{i}}_P = \dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+ \bar{\mathbf{i}}_S + (\dot{\mathbf{Y}}_{PP} - \dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+ \dot{\mathbf{Y}}_{SP}) \bar{\mathbf{v}}_P \approx \dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+ \bar{\mathbf{i}}_S \quad (3.18)$$

In eq.3.18 the second term is very small, thus negligible. Now that primary current can be rewritten as function of secondary current, eq.3.16 can be inverted to calculate voltages:

$$\begin{bmatrix} \bar{\mathbf{v}}_P \\ \bar{\mathbf{v}}_S \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{Y}}_{PP} & \dot{\mathbf{Y}}_{PS} \\ \dot{\mathbf{Y}}_{SP} & \dot{\mathbf{Y}}_{SS} \end{bmatrix}^+ \begin{bmatrix} \bar{\mathbf{i}}_P \\ \bar{\mathbf{i}}_S \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{Z}}_{PP} & \dot{\mathbf{Z}}_{PS} \\ \dot{\mathbf{Z}}_{SP} & \dot{\mathbf{Z}}_{SS} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{i}}_P \\ \bar{\mathbf{i}}_S \end{bmatrix} \quad (3.19)$$

$$\bar{\mathbf{v}}_P = \dot{\mathbf{Z}}_{PP} (\dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+ \bar{\mathbf{i}}_S) + \dot{\mathbf{Z}}_{PS} \bar{\mathbf{i}}_S \quad (3.20)$$

$$\bar{\mathbf{v}}_S = \dot{\mathbf{Z}}_{SP} (\dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+ \bar{\mathbf{i}}_S) + \dot{\mathbf{Z}}_{SS} \bar{\mathbf{i}}_S \quad (3.21)$$

Losses can then be calculated as:

$$\begin{aligned} \bar{\mathbf{S}}_{tr,loss} &= \bar{\mathbf{v}}_P * \bar{\mathbf{i}}_P^* + \bar{\mathbf{v}}_S * \bar{\mathbf{i}}_S^* \quad (3.22) \\ &= \bar{\mathbf{i}}_S^* * \left[(\dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+)^* \dot{\mathbf{Z}}_{PP} (\dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+) + (\dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+)^* \dot{\mathbf{Z}}_{PS} + \dot{\mathbf{Z}}_{SP} (\dot{\mathbf{Y}}_{PS} \dot{\mathbf{Y}}_{SS}^+) + \dot{\mathbf{Z}}_{SS} \right] \bar{\mathbf{i}}_S \end{aligned}$$

where * indicates the component product between two matrices and \mathbf{A}^+ the pseudo-inverse of matrix \mathbf{A} . In this way, transformer losses can be computed exactly using the secondary current as only variable. Losses in the transformer can then be assigned to each secondary current of each conductor, without worrying about how currents and voltages are transformed to the primary side, since the topology, group and connection type of the transformer are already included into its admittance matrix.

In Appendix A can be seen how this methodology was included in MPNLA. In the aforementioned network of fig.3.3, ABCLA would have simply generated a $\bar{\mathbf{S}}_{branch}$ matrix as the one represented in eq.3.23. MPNLA instead substitutes the columns concerning transformers with the new, exact, losses calculated from the secondary side, building the matrix that in eq.3.23 is called $\bar{\mathbf{S}}_{branch,mod}$.

$$\bar{\mathbf{S}}_{branch} = \begin{bmatrix} \bar{S}_{br1,1}^a & 0 & 0 \\ \bar{S}_{br1,1}^b & 0 & 0 \\ \bar{S}_{br1,1}^c & 0 & 0 \\ \bar{S}_{br1,1}^n & 0 & 0 \\ \bar{S}_{br1,2}^a & \bar{S}_{br2,2}^a & 0 \\ \bar{S}_{br1,2}^b & \bar{S}_{br2,2}^b & 0 \\ \bar{S}_{br1,2}^c & \bar{S}_{br2,2}^c & 0 \\ \bar{S}_{br1,2}^n & \bar{S}_{br2,2}^n & 0 \\ \bar{S}_{br1,3}^a & \bar{S}_{br2,3}^a & \bar{S}_{br3,3}^a \\ \bar{S}_{br1,3}^b & \bar{S}_{br2,3}^b & \bar{S}_{br3,3}^b \\ \bar{S}_{br1,3}^c & \bar{S}_{br2,3}^c & \bar{S}_{br3,3}^c \\ \bar{S}_{br1,3}^n & \bar{S}_{br2,3}^n & \bar{S}_{br3,3}^n \end{bmatrix} \quad \bar{\mathbf{S}}_{branch,mod} = \begin{bmatrix} \bar{S}_{tr,loss,1}^a & 0 & 0 \\ \bar{S}_{tr,loss,1}^b & 0 & 0 \\ \bar{S}_{tr,loss,1}^c & 0 & 0 \\ \bar{S}_{tr,loss,1}^n & 0 & 0 \\ \bar{S}_{tr,loss,2}^a & \bar{S}_{br2,2}^a & 0 \\ \bar{S}_{tr,loss,2}^b & \bar{S}_{br2,2}^b & 0 \\ \bar{S}_{tr,loss,2}^c & \bar{S}_{br2,2}^c & 0 \\ \bar{S}_{tr,loss,2}^n & \bar{S}_{br2,2}^n & 0 \\ \bar{S}_{tr,loss,3}^a & \bar{S}_{br2,3}^a & \bar{S}_{br3,3}^a \\ \bar{S}_{tr,loss,3}^b & \bar{S}_{br2,3}^b & \bar{S}_{br3,3}^b \\ \bar{S}_{tr,loss,3}^c & \bar{S}_{br2,3}^c & \bar{S}_{br3,3}^c \\ \bar{S}_{tr,loss,3}^n & \bar{S}_{br2,3}^n & \bar{S}_{br3,3}^n \end{bmatrix} \quad (3.23)$$

Here are reported results for balanced 4-buses network (Fig.3.3), with a Dyn11 configuration for the transformer and where three-phase loads L_3 and L_4 are each defined as $P=9$ [kW] and $\cos\varphi=0.9$.

Losses are:

$$\begin{aligned}\bar{\mathbf{S}}_{\text{loss,tr}} &= \begin{bmatrix} 20,3219 + 51,3938i \\ 20,2733 + 51,2710i \\ 20,2629 + 51,3478i \\ 1,7739e-05 + 4,4885e-05i \end{bmatrix} [W] \\ \bar{\mathbf{S}}_{\text{loss,br(2)}} &= \begin{bmatrix} 21,2532 + 27,8936i \\ 13,6179 + 26,6063i \\ 14,3273 + 29,4383i \\ -5,7939e-05 + 0,0025i \end{bmatrix} [W] \\ \bar{\mathbf{S}}_{\text{loss,br(3)}} &= \begin{bmatrix} 5,3254 + 6,9898i \\ 3,4100 + 6,6628i \\ 3,5890 + 7,3745i \\ -1,0590e-05 + 0,0007i \end{bmatrix} [W]\end{aligned}$$

And they are allocated to loads as follow:

$$\begin{aligned}\bar{\mathbf{L}}_{\text{alloc,L3}} &= \begin{bmatrix} 20,7600 \\ 16,9291 \\ 17,2805 \\ -1,7468e-05 \end{bmatrix} [W] \\ \bar{\mathbf{L}}_{\text{alloc,L4}} &= \begin{bmatrix} 26,1406 \\ 20,3722 \\ 20,8987 \\ -3,3321e-05 \end{bmatrix} [W]\end{aligned}$$

The sum of the losses allocated phase by phase match exactly the total losses on the system for each conductor. Losses calculated and allocated to neutral are to be consider negligible, since they are due to mutual coupling between the conductors and no current flows in the neutral due to the loads, which are perfectly balanced.

Considering now the unbalanced 4-buses network (Fig.3.4), with a Dyn11 configuration for the transformer and where loads L_3 and L_4 are each defined as $P=3$ [kW] per phase and $\cos\varphi=0.9$ for each phase connected, losses are computed as:

$$\begin{aligned}\bar{\mathbf{S}}_{\text{loss,tr}} &= \begin{bmatrix} 3,3269 + 24,4473i \\ 22,1555 + 40,0408i \\ 0,0000 + 0,0000i \\ 5,0264 + 12,7203i \end{bmatrix} [W] \\ \bar{\mathbf{S}}_{\text{loss,br(2)}} &= \begin{bmatrix} 5,9347 + 6,1316i \\ 15,3348 + 27,5117i \\ 0,0000 + 0,0000i \\ 22,8909 + 25,1384i \end{bmatrix} [W] \\ \bar{\mathbf{S}}_{\text{loss,br(3)}} &= \begin{bmatrix} 0,0000 + 0,0000i \\ 4,1658 + 7,1025i \\ 0,0000 + 0,0000i \\ 8,0594 + 8,6121i \end{bmatrix} [W]\end{aligned}$$

And they are allocated to loads as follow:

$$\bar{\mathbf{L}}_{\text{alloc,L3}} = \begin{bmatrix} 9,2616 \\ 22,0907 \\ 0 \\ 20,9830 \end{bmatrix} [W]$$

$$\bar{\mathbf{L}}_{\text{alloc,L4}} = \begin{bmatrix} 0 \\ 19,5650 \\ 0 \\ 14,9938 \end{bmatrix} [W]$$

This time, a significant portion of losses are allocate on the neutral, since loads are not balanced and currents have to return back to the star point of the transformer. The meaning of this allocation will be analysed in chapter 5. It is worth pointing out the big allocation difference between L_3^a and L_3^b : the loads are equal, they are at the same bus, common knowledge would suggest they should be allocated with similar losses. The asymmetries in the network cannot explain such a difference: load on phase b has 138% more losses allocated to it when compared to the load on phase a . The reason behind this behaviour has been explained in section 2.1. On phase a , L_3 is the only load connected, the losses allocated to it are independent from the network conditions. On phase b instead there are other loads already connected which are generating losses. The insertion of load L_3 intensify the losses also for the the other loads on the same phase. This is why loads that share the same conductor are allocated much more losses to.

Testing the method on the network of fig.3.5 reveals that losses computed are higher, when compared with the load flow result. This shows another, final, problem to address. As it has been modified so far, MPNLA method takes into consideration just partially the topology of the grid. The creation of $\bar{\mathbf{I}}_{\text{mat}}$ takes for granted that the branches are previously sorted from the one closer to the root bus to further away (terminal branch). In this way, loads and branches are already organized properly so that a load at the end of branch (3) will be responsible for the losses occurring on the all previous branches: (1), (2) and (3). Unfortunately this method do not work when from one bus start multiple branches: there is no suitable numeration scheme for the branches since some of them are running in "parallel" on multiple feeder. For the tested network, $\bar{\mathbf{S}}_{\text{branch}}$ obtained from MPNLA original method is:

$$\bar{\mathbf{S}}_{\text{branch}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,2}} & \bar{\mathbf{S}}_{\text{br2,2}} & 0 & 0 & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,3}} & \bar{\mathbf{S}}_{\text{br2,3}} & \bar{\mathbf{S}}_{\text{br3,3}} & 0 & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,4}} & \bar{\mathbf{S}}_{\text{br2,4}} & \bar{\mathbf{S}}_{\text{br3,4}} & \bar{\mathbf{S}}_{\text{br4,4}} & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,5}} & \bar{\mathbf{S}}_{\text{br2,5}} & \bar{\mathbf{S}}_{\text{br3,5}} & \bar{\mathbf{S}}_{\text{br4,5}} & \bar{\mathbf{S}}_{\text{br5,5}} \end{bmatrix} \quad (3.24)$$

Where, for compactness, $\bar{\mathbf{S}}_{\text{br1,1}}$ is the vector containing $\bar{S}_{br1,1}$ for each phase, whilst $\bar{\mathbf{S}}_{\text{branch}}$ should have been:

$$\bar{\mathbf{S}}_{\text{branch}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,2}} & \bar{\mathbf{S}}_{\text{br2,2}} & 0 & 0 & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,3}} & \bar{\mathbf{S}}_{\text{br2,3}} & \bar{\mathbf{S}}_{\text{br3,3}} & 0 & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,4}} & \bar{\mathbf{S}}_{\text{br2,4}} & \bar{\mathbf{S}}_{\text{br3,4}} & \bar{\mathbf{S}}_{\text{br4,4}} & 0 \\ \bar{\mathbf{S}}_{\text{tr_loss,5}} & \bar{\mathbf{S}}_{\text{br2,5}} & 0 & 0 & \bar{\mathbf{S}}_{\text{br5,5}} \end{bmatrix} \quad (3.25)$$

$\bar{\mathbf{S}}_{\text{tr_loss},1}$ is null because there is no load attached at the receiving end of the transformer (i.e. bus 2), $\bar{\mathbf{S}}_{\text{br}3,5}$ and $\bar{\mathbf{S}}_{\text{br}4,5}$ should be null because the load at the receiving end of line (5) (i.e. bus 6) is not responsible for any of the losses on branches (3) and (4), since they are not along its path towards the root bus.

To overcome this issue, a technique, capable of making MPNLA algorithm topology-sensible, has been developed using an incidence matrix.

A single-phase equivalent incidence matrix \mathbf{A} with $N_{br} \times N$ dimension is created as follows. Each row represents a branch: the node where the branch starts has a 1 assigned to its column, the column where it ends a -1 . For the network of fig.3.5:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \quad (3.26)$$

The full $\bar{\mathbf{S}}_{\text{branch}}$ is now computed, as in eq.3.24, then a new matrix $\bar{\mathbf{S}}_{\text{branch,new}}$ with the same dimension is initialized. The algorithm swipes the incidence matrix to understand the topology of the grid, it generates position indices and then copies elements with said indices from $\bar{\mathbf{S}}_{\text{branch}}$ to $\bar{\mathbf{S}}_{\text{branch,new}}$.

Let us clarify with an example the test network: the algorithm reads each row of \mathbf{A} separately. Let us for instance analyse the process for $n_{row}=2$. The script reads the second row and searches for the -1 , then reads the column where it finds it (3). The script reads now column $n_{col}=3$ and searches for 1, it reads 3 and 5, then goes on with the same process (search for -1 in rows, search for 1 in columns) and finds 4. At this point the process stops because no more 1s are found. To recap, starting from $n_{row}=2$ we found and stored the indices 3,5 and 4, representing the rows where 1s were present throughout our path. Thus, the elements that are going to be copied from $\bar{\mathbf{S}}_{\text{branch}}$ to $\bar{\mathbf{S}}_{\text{branch,new}}$ in this swipe belong to column $[n_{row}]$ and rows $[n_{row},3,4,5]$, therefore are $\bar{\mathbf{S}}_{\text{br}2,2}$, $\bar{\mathbf{S}}_{\text{br}2,3}$, $\bar{\mathbf{S}}_{\text{br}2,4}$ and $\bar{\mathbf{S}}_{\text{br}2,5}$.

In the same way, if n_{row} is taken equal to 3, the only index found will be 4: from $\bar{\mathbf{S}}_{\text{branch}}$ the elements that are copied over are $\bar{\mathbf{S}}_{\text{br}3,3}$ and $\bar{\mathbf{S}}_{\text{br}4,3}$ but not $\bar{\mathbf{S}}_{\text{br}3,5}$.

This modified version of ABCLA, called indeed MPNLA, optimized to require load flow results only able to work on multi-voltage level systems and topology sensible, here presented and implemented, has proven to perform well. As already mentioned, the full code can be found in Appendix A for further an in-depth analysis.

Chapter 4

Simulation on e-distribuzione network

MPNLA method was tested on a real network, courtesy of e-distribuzione S.p.A., under different scenarios. The main objective of the testing is to use the procedure developed to analyse and gain more knowledge on how DG affects losses on a real grid, designed to be passive and slowly turning active due to a constantly increasing penetration of renewable energies.

4.1 The network

The datas for the network modelling were provided by the italian DSO, e-distribuzione, and they were gather from a small 37-node network in Veneto region, north of Italy. Due to privacy concerns, datas cannot be reported on this theses, only the main characteristics are here represented.

The distribution network is shown in fig.4.1. It counts 37 nodes and 36 lines. The transformer works between medium voltage (20 [kV]) and low voltage (415 [V]), where all loads and generators are connected. It is a Dyn11 transformer, an italian standard for distribution networks, where the star point of the secondary side is directly grounded. This is the only grounding point for the grid, as it follows the TT standard [15]: each user provides for its own earthing system. Nominal power of the transformer is 250 [kVA], short circuit voltage is 4%, off-load tap changing is possible with 2.5% voltage steps.

There are 19 different line configuration, most of which are overhead, using multiple wires, ranging from 150 [mm²] copper (longitudinal resistance $R_c = 0.127[\Omega/km]$) to 16 [mm²] aluminium ($R_c = 3.06[\Omega/km]$). Minimum length of a branch is 14 [m], whilst the maximum is 170 [m]. The longest feeder, which is towards node 34, shows a total length from the root node of 492 [m].

The majority of loads are domestic and single-phase, with a nominal power of 3 [kW] and a non-capped power factor, assumed to be 0.9 in the following simulations. The user connected to the network with the maximum nominal power is a three-phase 22 [kW] commercial load on node 15. ZIP coefficients for the loads are experimental measurement-based coefficients present in scientific literature [16]. Although the installed nominal power of the loads is just above 400 [kVA], during the simulations total absorbed power was rarely exceeding 200 [kVA]

Generators are few, all photovoltaic and mostly single-phase.

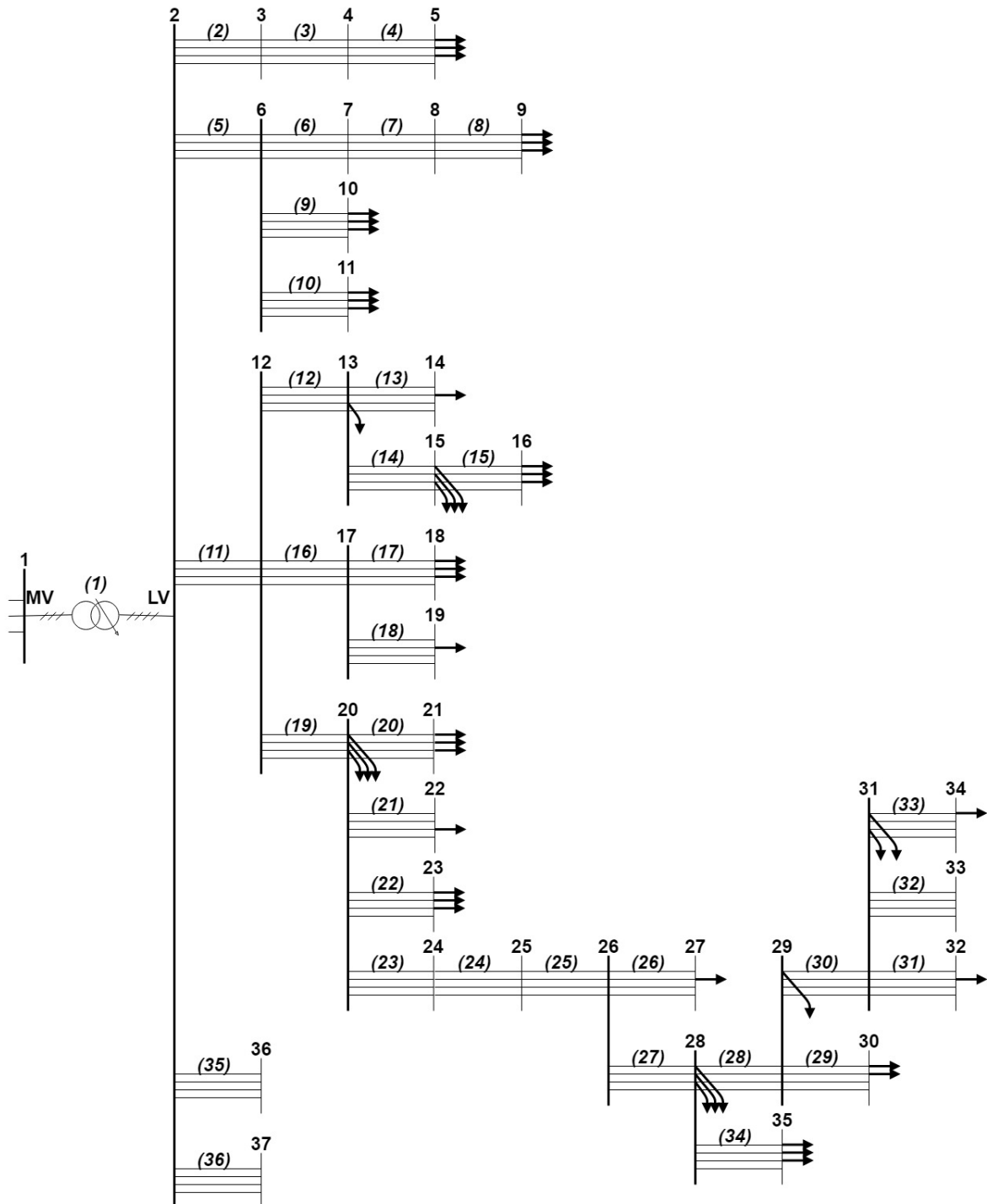


Figure 4.1: Schematic of e-distribuzione network

Load profiles, both domestic and commercial, are experimental, measured on the grid during year 2013, generator profiles instead are taken from literature. The time steps are 15 minutes, so that each instant in time represent the average power for 15 minutes. The following simulation are run just for one instant in time. After an extent analysis of the network operation, the instant selected to run the following simulations was chosen as optimum point between high distributed generation and high load demand, in order to be able to compare fairly an highly loaded passive network with an active network where current inversion occurs. Therefore, the instant chosen for the simulation was June 22nd 2013 at 10:30 am.

4.2 Passive scenario

In this scenario, the whole grid is fed by the medium voltage network through the transformer. DG is completely absent. The load flow is calculated and the solutions are plotted for an easy understanding, using the code shown in Appendix B. Phase voltages for each node are shown in fig.4.2, represented as magnitude of the difference between the phase potential phasor and neutral potential phasor. Zero and negative sequences potentials are shown too.

Voltages are within expectations and within the $\pm 10\%$ range with respect to the nominal value, which is a variation considered to be acceptable by the italian DSO [18]. It has to be noted that phase b is the most loaded phase ($+8.1\%$ compared to phase a , $+25.9\%$ to phase c), so that the network is fairly unbalanced, but still within reasonable limits. Since phase b is the most loaded, it is expected to have more losses and the highest voltage drop, and this is true for some node. Yet, the lowest voltage can be found on phase a , node 30 (0.9198 p.u.). This phenomenon is shown in fig.4.3 and testify the importance of the topology of the grid in the losses analysis. Feeder 3 (the one feeding node 34) is highly loaded phase a towards its end on. The location of this loads is cause of higher losses and voltage drops when compared to similar, or even higher, loads locate in a more advantageous position.

Both the load flow and MPNLA provide solutions for the branch losses. They match with a margin of error of about 10^{-4} and they are plotted in fig.4.4. Core losses in the transformer are neglected in the calculations, so that they will not be allocated to users (see chapter 5). On the other hand, winding losses in the transformer accounts for:

$$S_{loss,winding} = \begin{bmatrix} 21,3633 + 64,9137i \\ 37,0960 + 95,9532i \\ 27,7503 + 57,3017i \\ 4,94719 + 12,5197i \end{bmatrix} [W]$$

Total losses in the system are $2830.1[W] + 1149.2[VAR]$.

It is clear from fig.4.4 how branch (10) experiences very high losses (above 1.3 [kW]), after all at its end (node 11) is connected a very demanding load and . It is important though to highlight how, in fig.4.5, to node 11 very few losses are allocated compared to the losses allocated to node 16, which is still a considerable but smaller load. Node 16 is fairly close to the root node, yet its voltage is pretty low. This is due to a very obsolete line: branch (15) (feeding node 16) has a longitudinal resistance of $3.06[\Omega/km]$ due to its $4 \times 6 [mm^2]$ copper conductors, so even if the line is short, the voltage drop produce is high, as shown in fig.4.2. The load on node 16 has therefore a big impact on surrounding buses, lowering their voltage. Moreover, load on node 16 is highly unbalanced (phase b is $+270\%$ more loaded than phase c), which is another reason for the consistent allocation term.

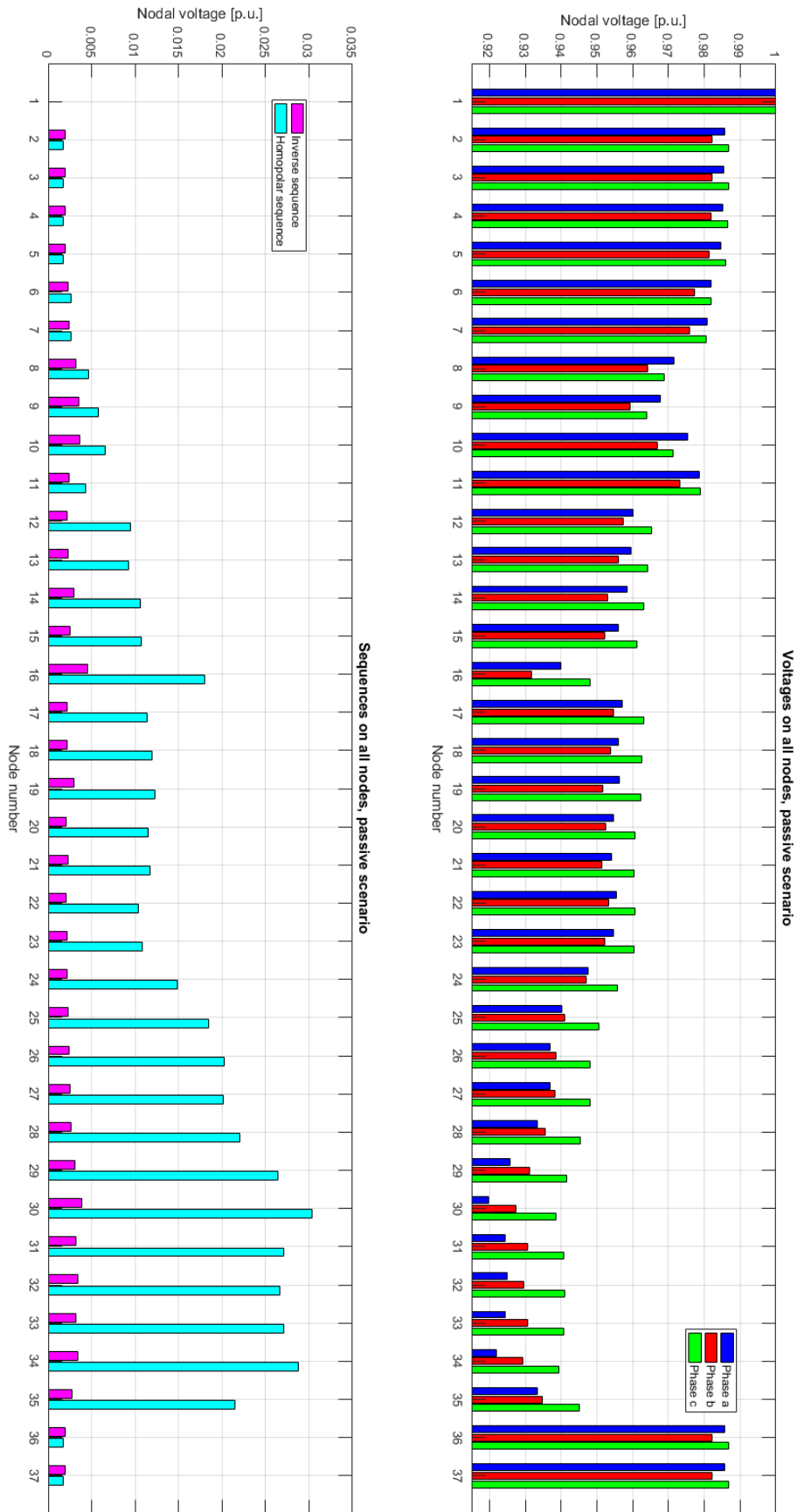


Figure 4.2: Voltage magnitudes in e-distribuzione network, passive scenario

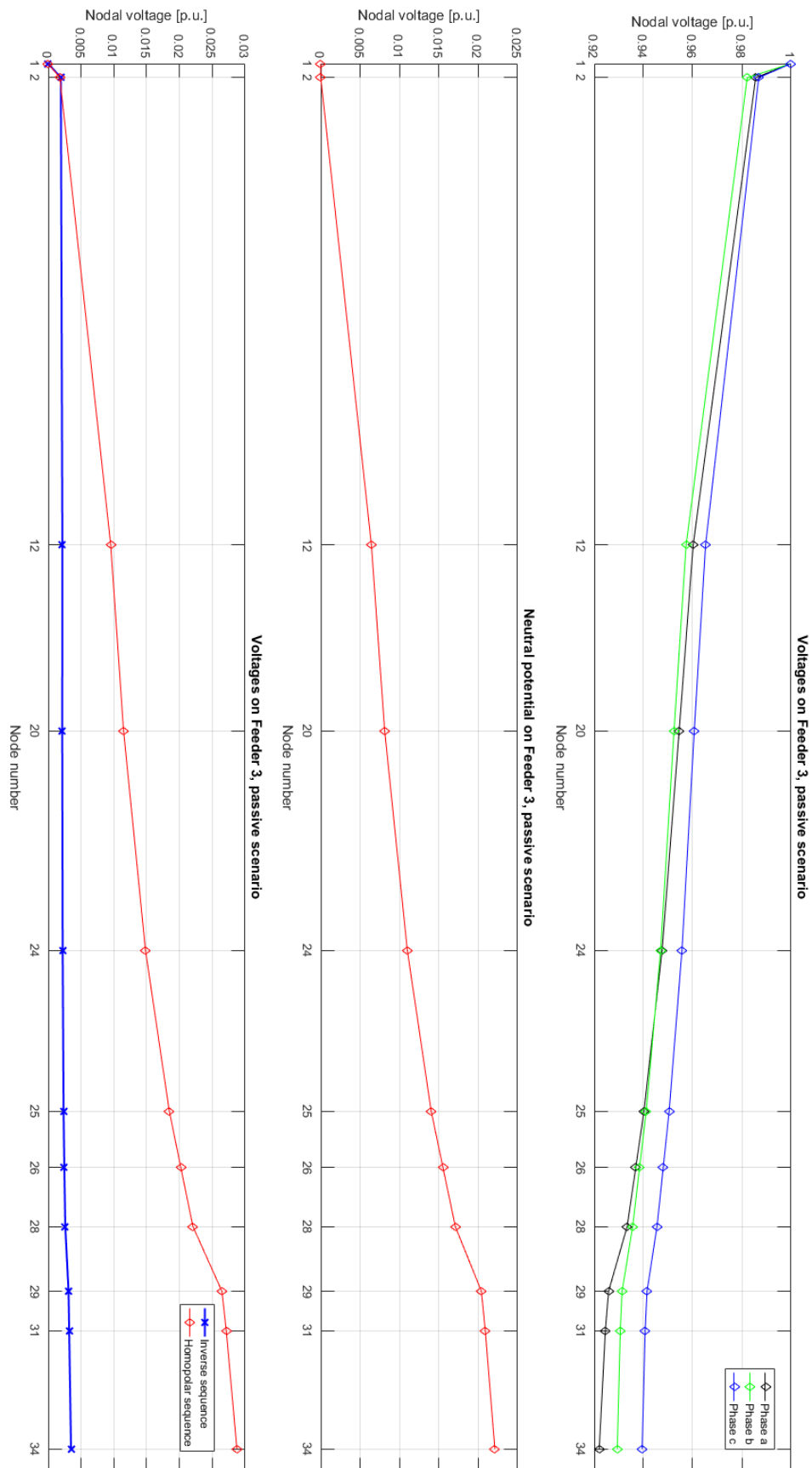


Figure 4.3: Voltage magnitudes in the longest feeder of e-distribuzione network, passive scenario

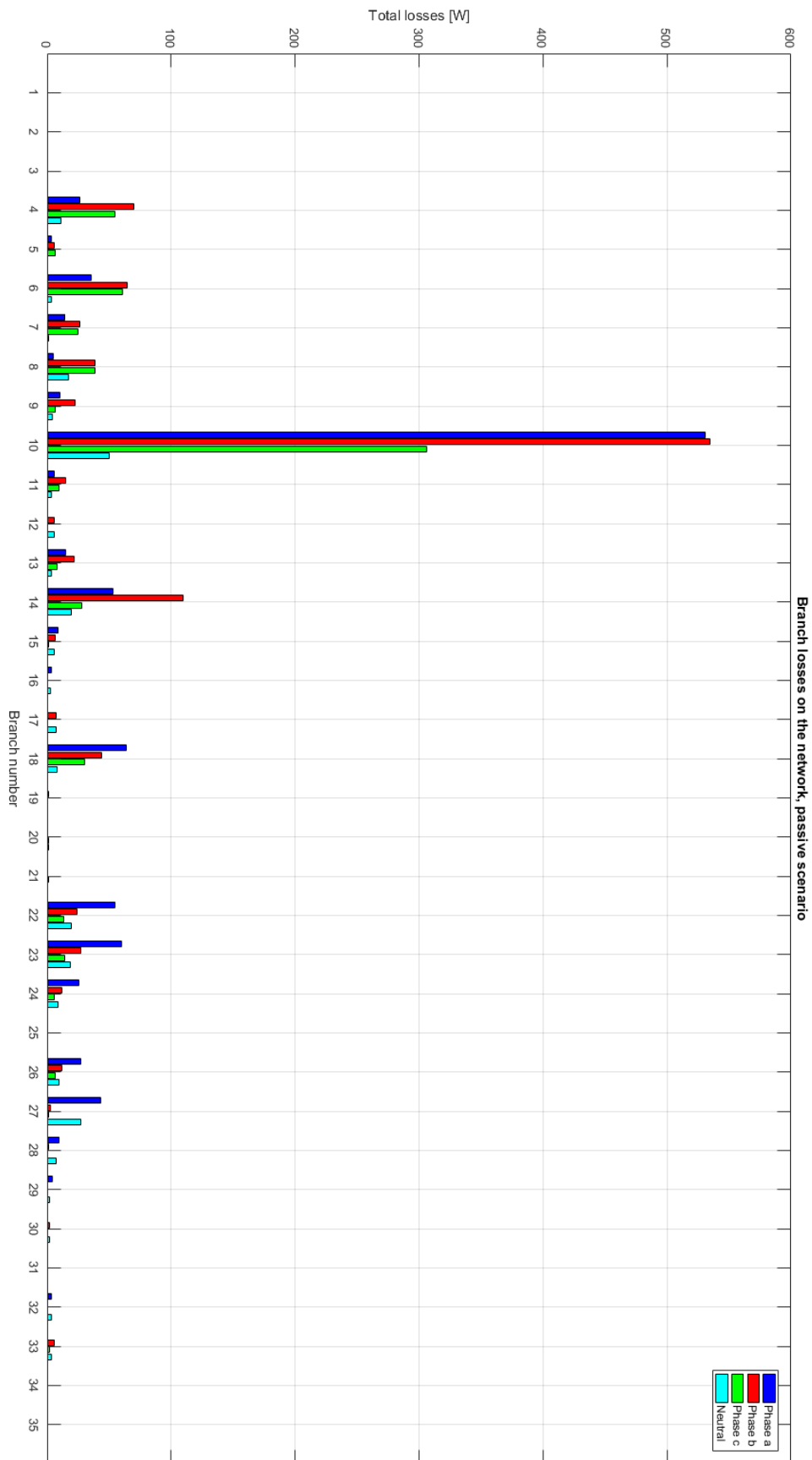


Figure 4.4: Magnitude of the branch power losses in e-distribuzione network, passive scenario

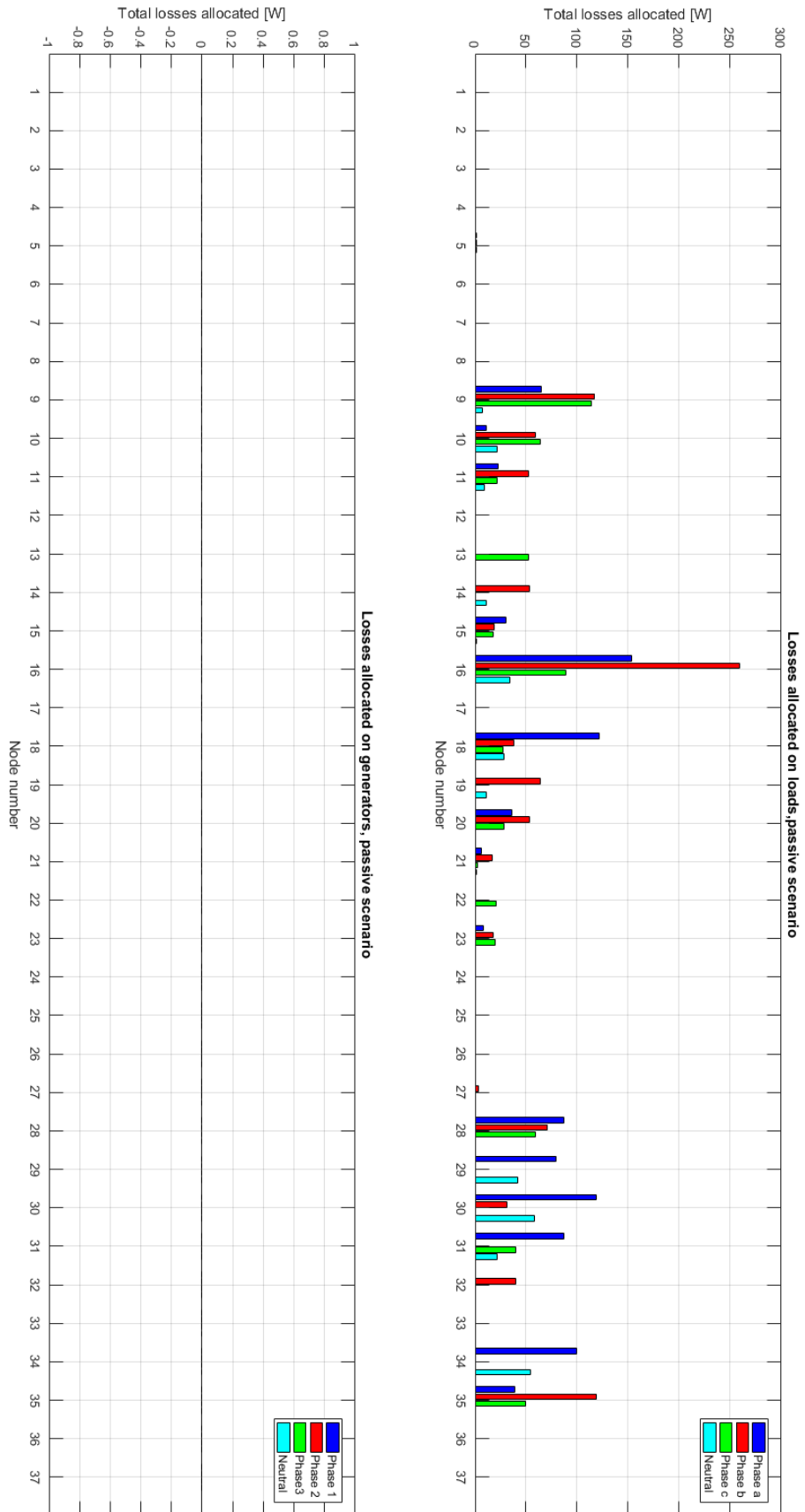


Figure 4.5: Allocated losses in e-distribuzione network, passive scenario, through MPNLA

To demonstrate this concept, the whole load flow and allocation process were run again on the same network, where loads on node 16 was reduced by around -40%, making them smaller. Losses on branch (10) are reduced by almost -20%, even though the loads connected to said line are still exactly the same as in the previous simulation. More importantly, losses allocated to node 16 are much less: -62% on phase a , -78% on phase b and -90% on phase c . This comparison well testify the strict relation between absorbed powers on very different points on the network and their mutual conditioning.

4.3 Real case scenario

This simulation will include the very few photovoltaic generators which are present in the actual network provided by the DSO. They are place on nodes 5, 16, 19 and 23. The simulation results are shown in the following pages, in fig. 4.6, 4.7 and 4.8.

The voltage plot reveal a similar trend to the passive scenario, with a slight voltage increase (within 1%) across all nodes. The only exception is given by the feeder that ends with bus 5: the voltage now does not slowly decrease from the root node to the terminal node, vice versa it increases. The reason behind this behaviour is that the power injected in bus 5 is greater than the power absorbed by loads in bus 5. Therefore the surplus current flows from node 5 to node 2, in the opposite direction as compared as in the passive scenario.

Thus the losses on the branches (2), (3) and (4) are now due to a current flowing in the opposite direction, generated by DG in bus 5. Generators in node 5 are therefore responsible for the losses on branches (2), (3) and (4) and are allocated with positive allocation terms, meaning they are "sanctioned", as shown in fig.4.8. Loads on node 5 instead are allocated null allocation terms, for they are consuming locally generated power, with no current demand for the network.

Losses on the branches follow a similar trend as the passive scenario as well, with a small but noticeable reduction in branches (10) and (14), as a consequence of the local generation inserted in node 16 phase b , which, as discovered earlier, is a critical node for the whole grid. The insertion of DG makes the network see a smaller load in node 16, which lower losses on branch (10) by roughly -8% on phase a and -15% on phase b . Since the generator in bus 16 is highly beneficial for the whole grid, it is highly rewarded. In fact, the generators on node 16 and 19 have same output power on the same phase, but one is rewarded +72% more than the other simply because its positioning is much more beneficial for the network than the other.

Winding losses on the transformer are about the same as the previous case, but with a very small increase:

$$S_{loss,winding} = \begin{bmatrix} 22,2439 + 67,1959i \\ 38,2184 + 98,7048i \\ 28,4544 + 59,1188i \\ 5,1800 + 13,1090i \end{bmatrix} [W]$$

Total losses in the system are 2595.7[W]+1076.3[VAR]: a total reduction of -8.01% when compared to the passive scenario.

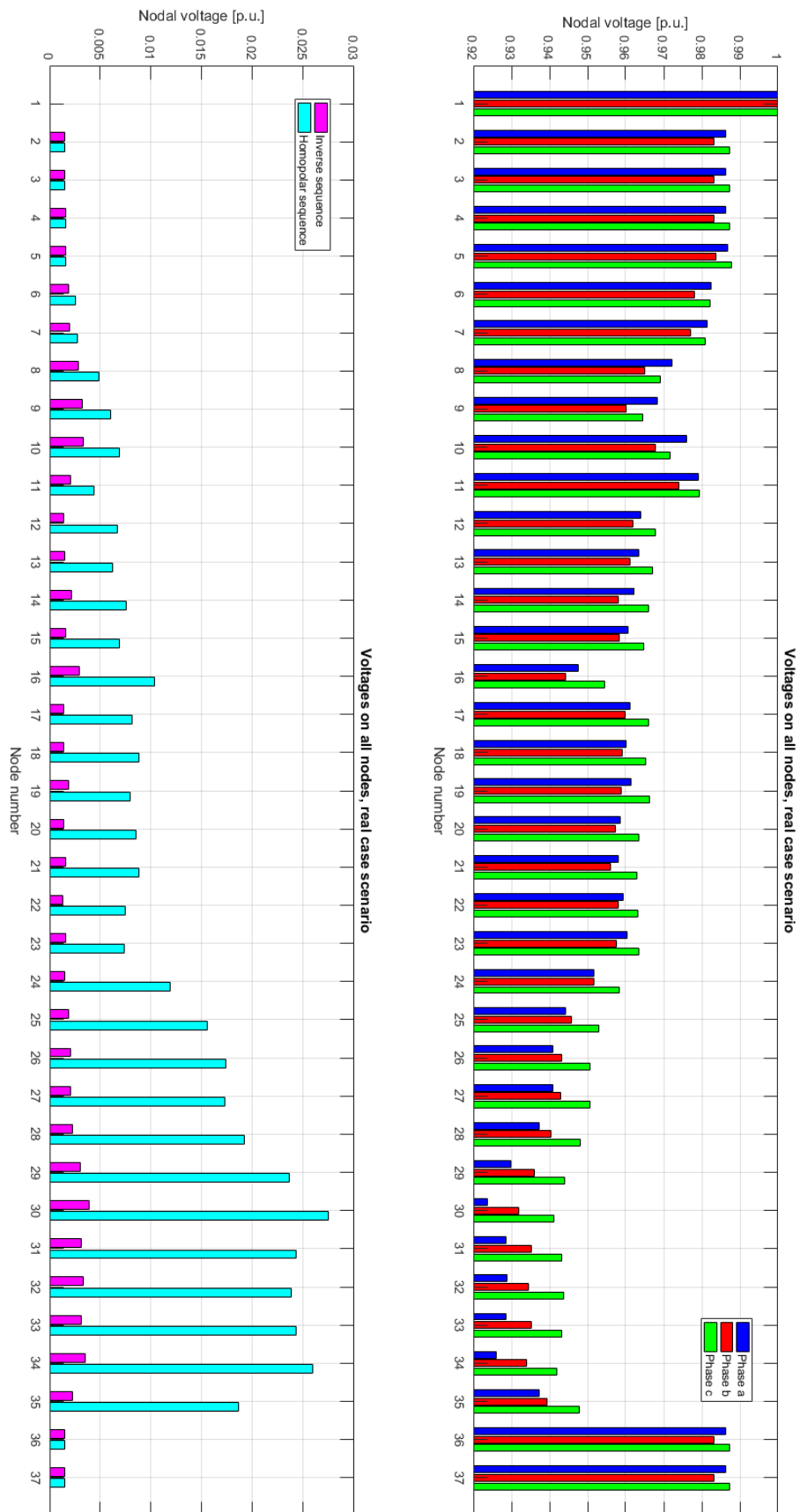


Figure 4.6: Voltage magnitudes in e-distribuzione network, real case scenario

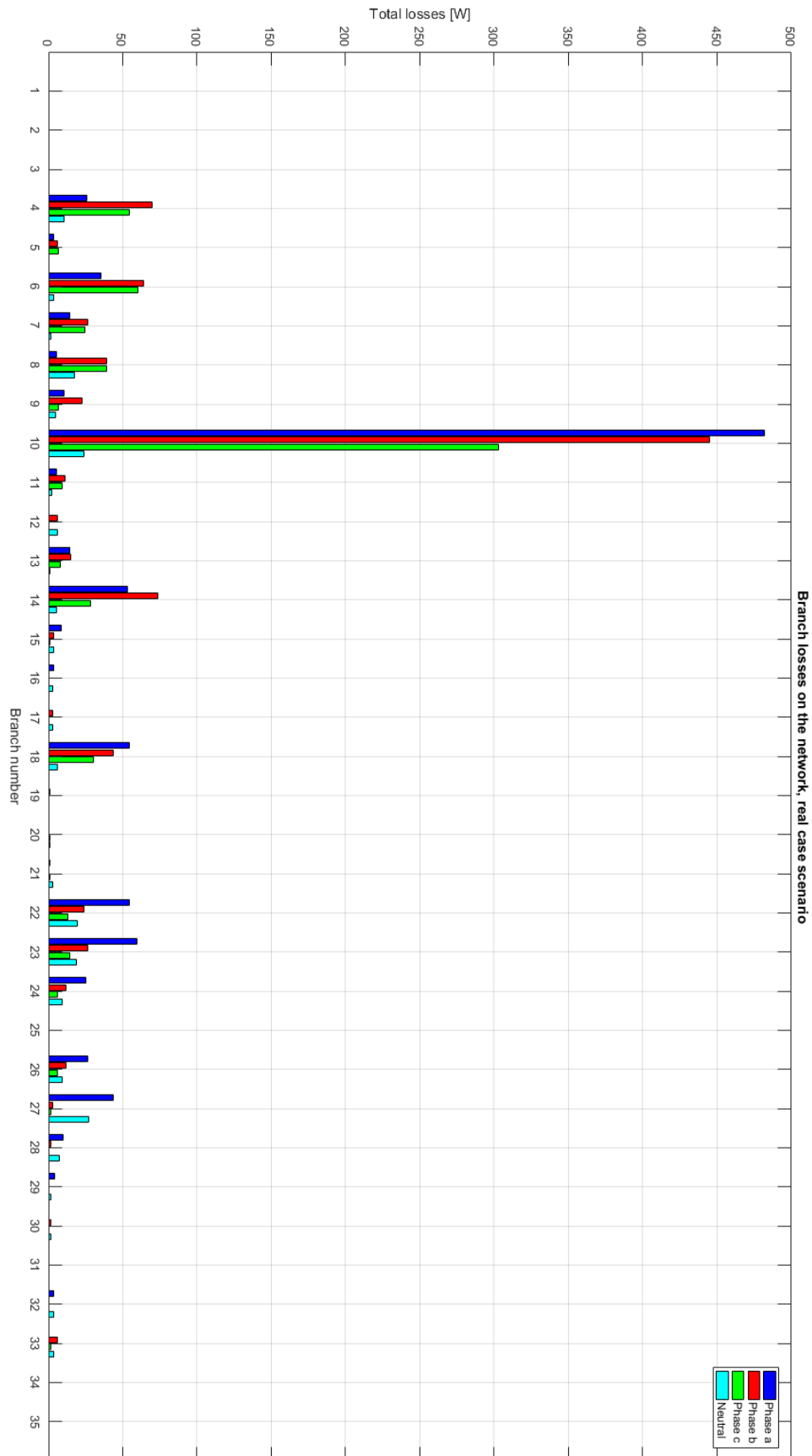


Figure 4.7: Magnitude of the branch power losses in e-distribuzione network, real case scenario

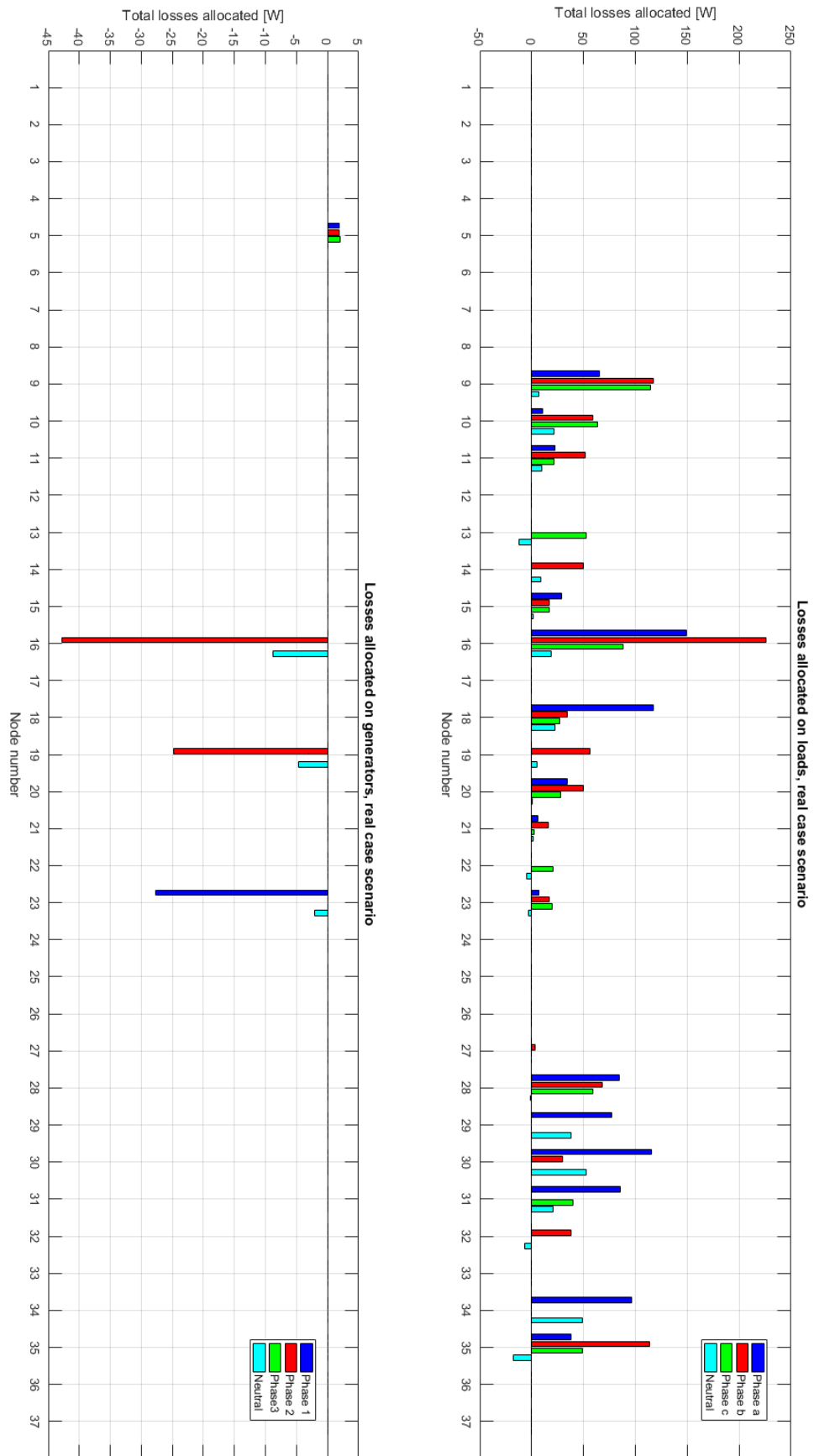


Figure 4.8: Allocated losses in e-distribuzione network, real case scenario

4.4 Optimal scenario

This scenario should mimic a future smart grid operation, where penetration of DG is high, each user is a prosumer who also generates electricity and the whole network is closely monitored and optimized. To do so, at every bus where a load is connected, a generator has been connected as well. The nominal power of the generator was chosen to reduce the losses as much as possible.

It has to be noted that the resulting configuration is not the optimal one for the grid, but is merely the configuration the lower losses the most in this specific instant in time.

The building process of this scenario is interesting and it is worth to be briefly discussed, since it shows unmentioned potentiality for MPNLA. Starting from the passive scenario, each load was backed up by a photovoltaic generator with the same nominal power. Obviously, this is not good enough. Generators nominal power has to be adjusted accordingly to the actual power absorbed at the given instant in time, considering the power injected by the generators themselves varies for each time instant with solar irradiance. But locally compensating active power is not enough either. Currents will flow through the branches nonetheless to provide reactive power and compensate for the voltage differences between nodes. This is why the optimization process was carried out using the losses allocation coefficients, since they are able to give strong signals on the conditions of the network. Thus, after each load was backed up by a photovoltaic generator of equal nominal power, load flow and MPNLA were performed. At this point, by looking merely at the Losses Allocated to Generators (LAG), the nominal power of each generator was adjust aiming to minimize phase losses allocated to all generators. Changing nominal power to one generator affect all the network and all losses allocated. The rationale behind this optimization process is the following: when LAG are negative, it means the generators are being rewarded for they are lowering the losses by locally injecting power. As generators nominal power is increased on a node, LAG will raise until they reach zero. If LAG are positive instead, it means the generators are contributing to raise the losses in the system by inverting the current direction in the branches, feeding loads that are not connected to the same node.

So this very rough process that works by trial and error was applied until LAG were considered satisfyingly low across all network. Now, looking at the load flow results (fig.4.9, 4.10 and 4.11), it is clear that voltages are way higher and consistent on all nodes with respect to the previous scenarios, losses are lower and Losses Allocated to Loads (LAL) are considerably smaller.

Losses on branch 10 are still remarkably high, accounting for more than half of the total active losses, but nonetheless they are almost one third compared to the ones of the passive scenario. Increasing power of the generators does not solve the problem and instead raise total network losses, since to lower losses on branch (10) would mean to insert more DG in nodes that have no load connected and this fall outside this particular scenario. Total generated power in the system accounts for 96% of the total load demand.

Total losses in the system are $738.5[W]+597.1[VAR]$: a total reduction of -68.9% when compared to the passive scenario, of -66.2% to the real case scenario, even though winding losses on the transformer are higher than in the previous cases:

$$S_{loss,winding} = \begin{bmatrix} 38,5473 + 114,9577i \\ 65,7630 + 168,3987i \\ 49,3485 + 105,5043i \\ 8,05935 + 20,3956i \end{bmatrix} [W]$$

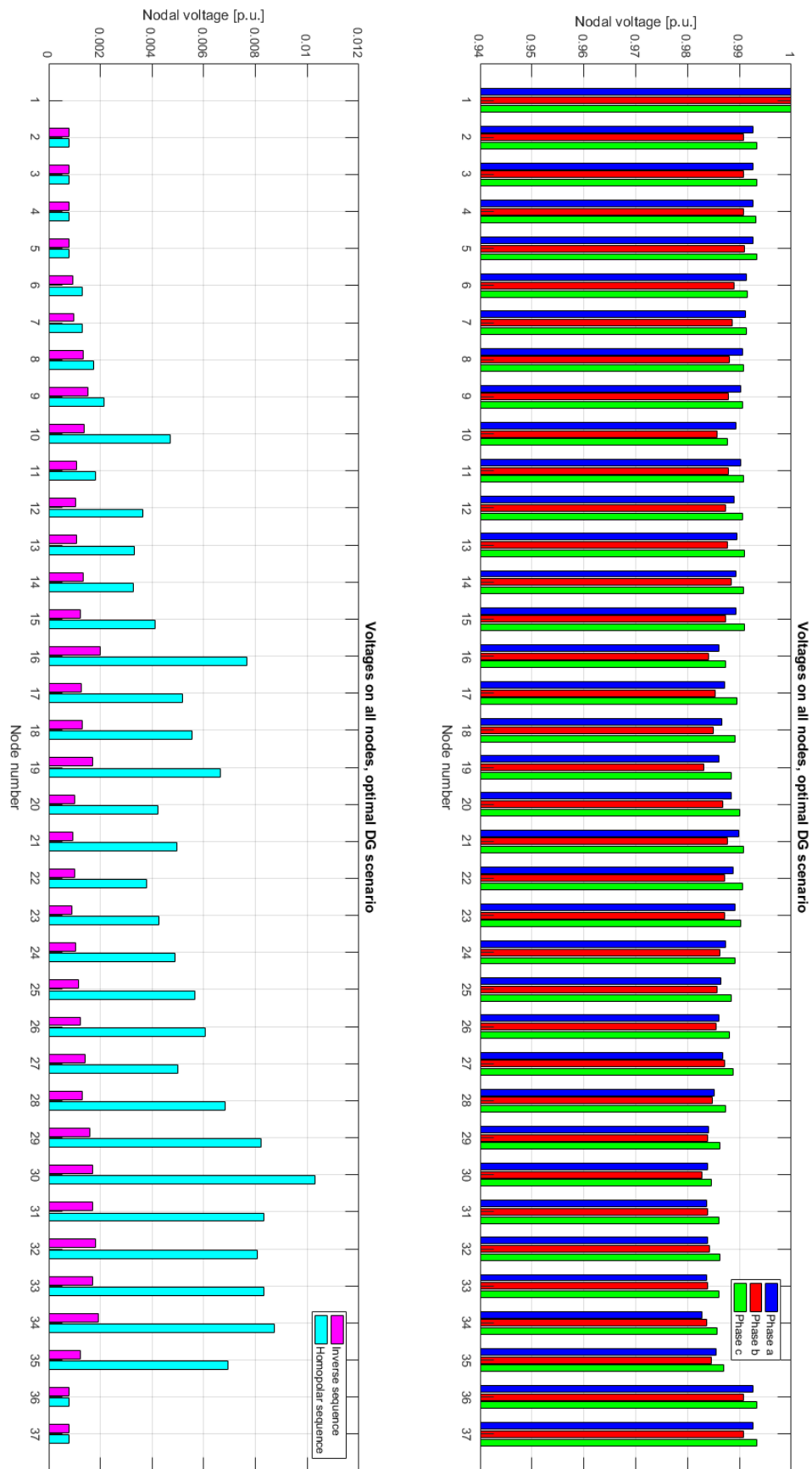


Figure 4.9: Voltage magnitudes in e-distribuzione network, optimal DG scenario

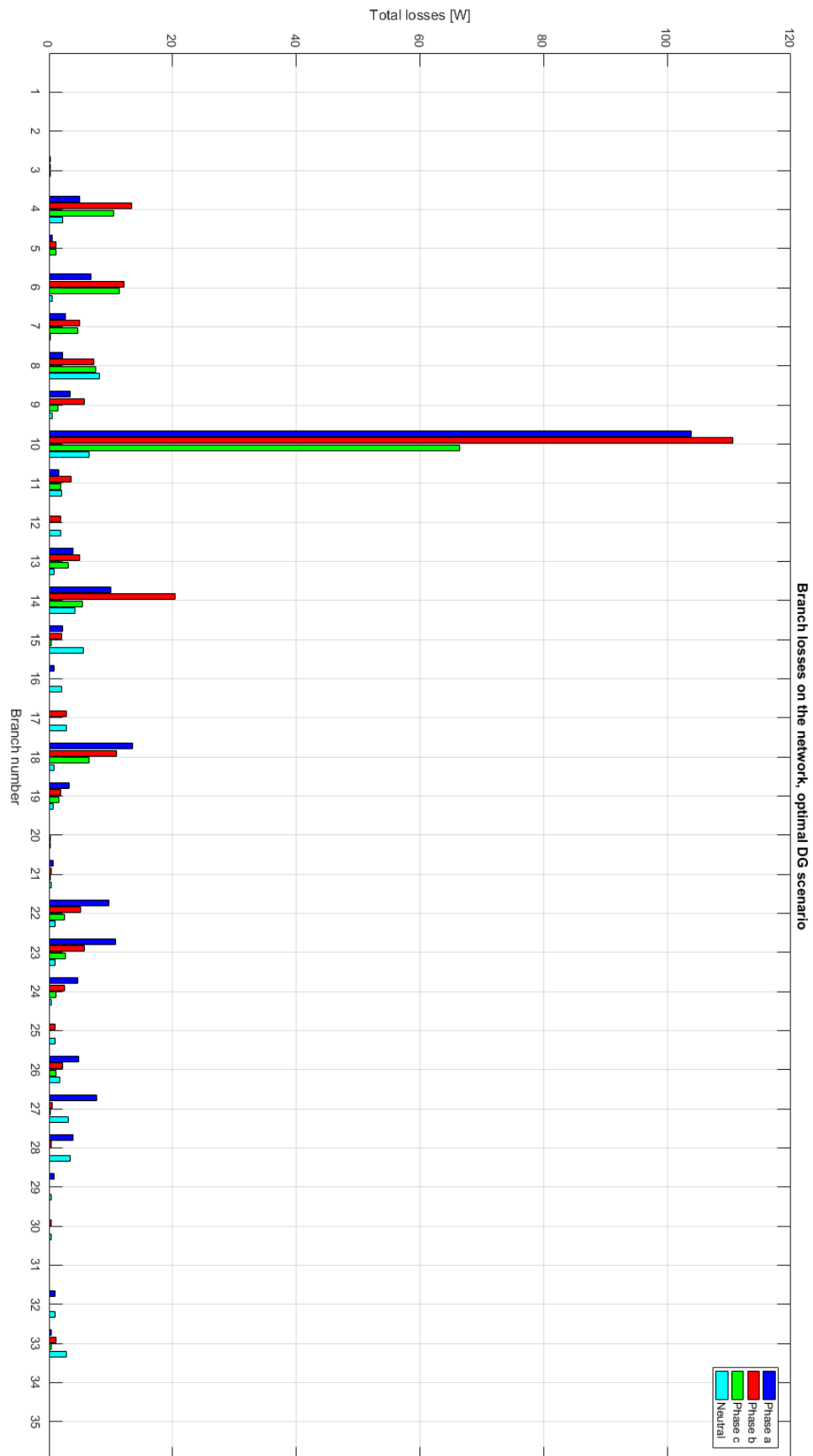


Figure 4.10: Magnitude of the branch power losses in e-distribuzione network, optimal DG scenario

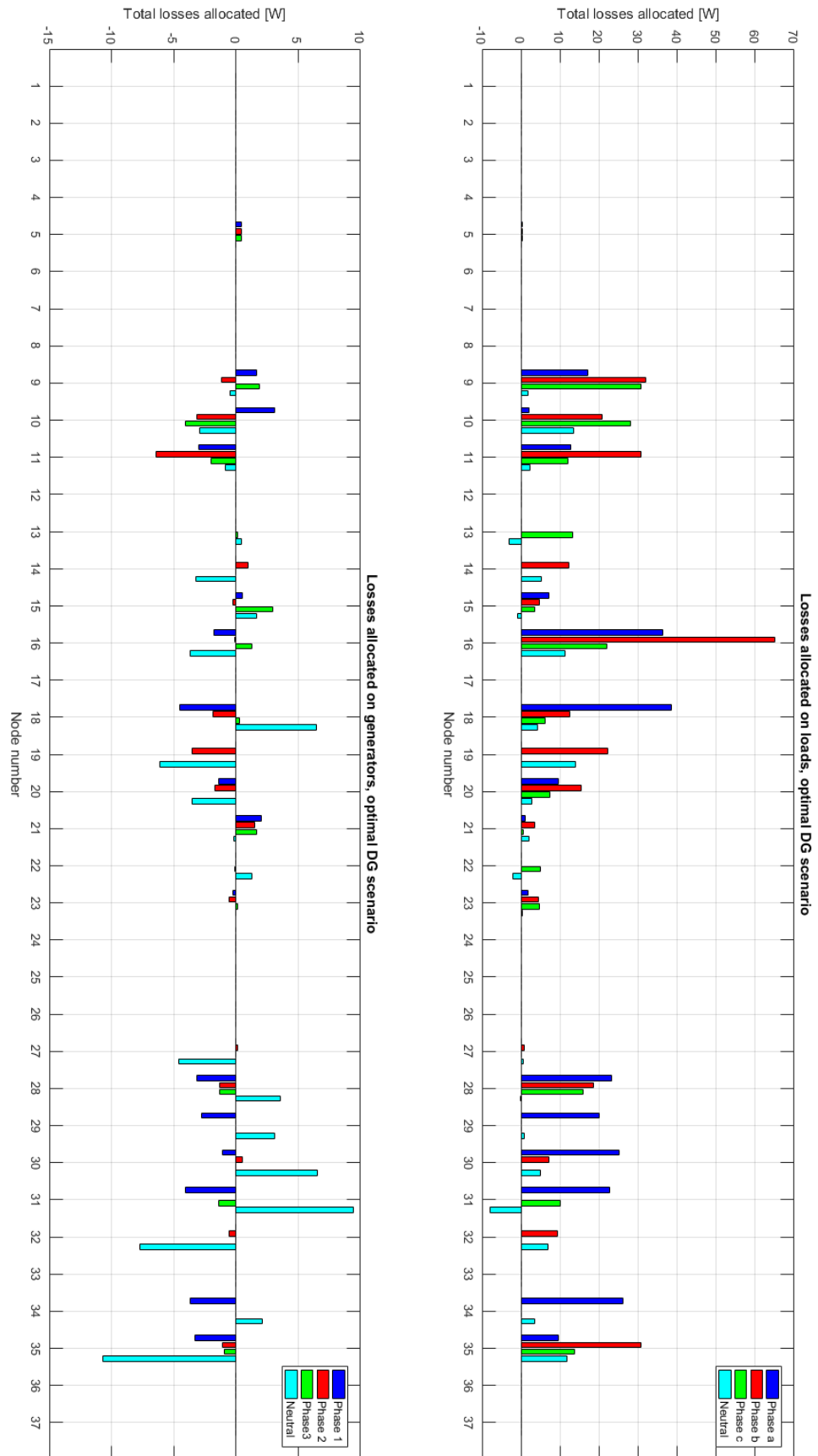


Figure 4.11: Allocated losses in e-distribuzione network, optimal DG scenario

4.5 Very active scenario

The objective of this last scenario is different from the previous one and it is to prove that a surplus in DG as well as a non-optimal placement of the generators leads to more losses. This time, starting from the passive network configuration, DG was added at the terminal nodes of each terminal branch, generating 15.4% more power than the one absorbed by the system. The total generated power in the network was equally spread between the phases, but this is not true for the generation in each node. The power at each bus and phase was chosen randomly, with the only focus being to install DG to roughly match the total nominal power of generators and loads.

The rationale behind this configuration is that, since terminal nodes are usually the ones experiencing the lowest voltages, by placing a copious amount of DG power, voltages will increase, reducing line voltage drops, thus losses.

The results (fig.4.12, 4.13, 4.14 and 4.15) show that this is just partially true and that a random placement of generators can be not only unuseful but also cause of higher losses.

Voltages across the node are definitely higher compared to the passive scenario, but lower than in the optimal scenario and, more importantly, not as consistent. Homopolar (zero) sequence is also much more present, pointing out the high degree of unbalances in the network. The analysis of voltages along the longest feeder (fig.4.13) reveals a major change in trend with respect to the passive scenario (fig.4.3). The terminal nodes are subject to a considerable voltage increase, bringing them to a higher potential compared to the middle nodes and therefore inverting the current flow from terminal branches towards the root node.

Overall losses in the system are $1493.6[W]+847.5[VAR]$: adding more DG in a non optimal way increases the losses by +80.8% compared to the optimal scenario. Losses in the transformer are higher as well:

$$S_{loss,winding} = \begin{bmatrix} 52,0746 + 157,9115i \\ 79,1879 + 186,3199i \\ 52,4222 + 120,6153i \\ 9,5333 + 24,1259i \end{bmatrix} [W]$$

Since this generator configuration is random, the same testing have been carried out also on a different random DG arrangement (yet of equal total power), resulting in double the losses of obtained in the optimal scenario. This, along with the allocation factors provided in fig.4.15, corroborate the initial statement: whilst low penetration of DG can be assumed as always beneficial, high shares of generation can lead to unwanted losses. Losses allocation terms represent a very useful tool to optimize the network, understand how to maximize the beneficial effect on the grid of a generator and reduce total system losses.

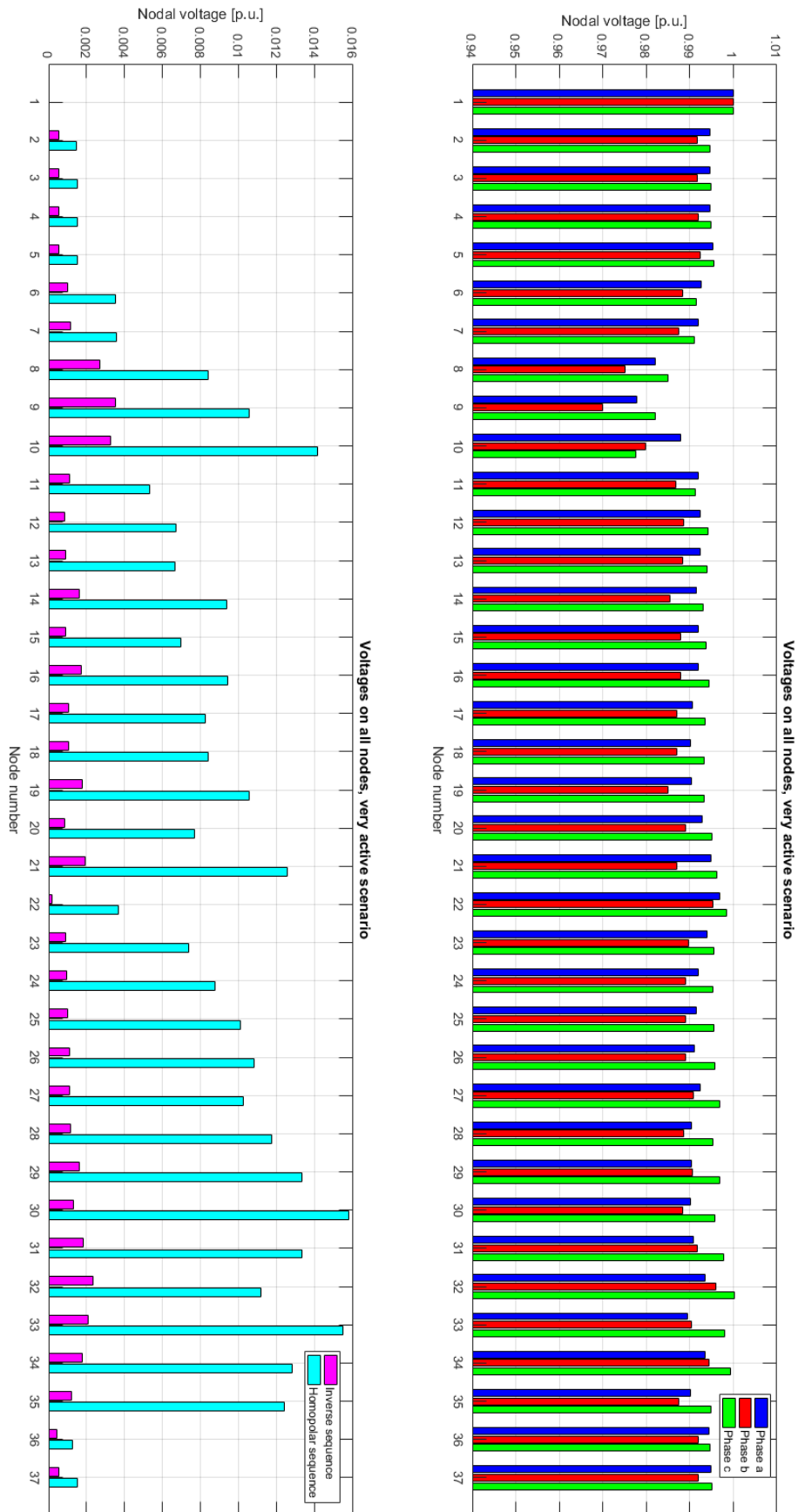


Figure 4.12: Voltage magnitudes in e-distribuzione network, current inversion scenario

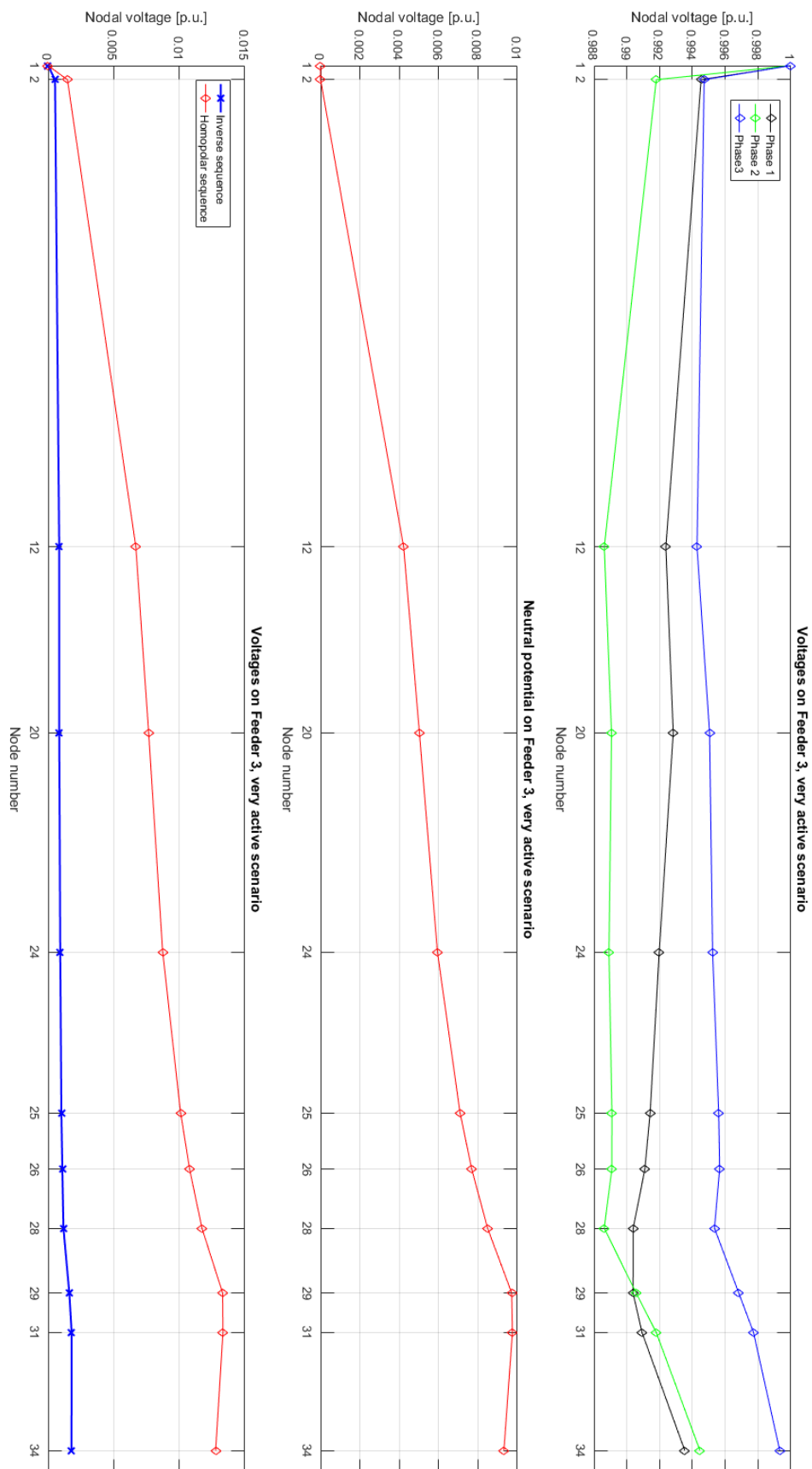


Figure 4.13: Voltage magnitudes in the longest feeder of e-distribuzione network, very active scenario

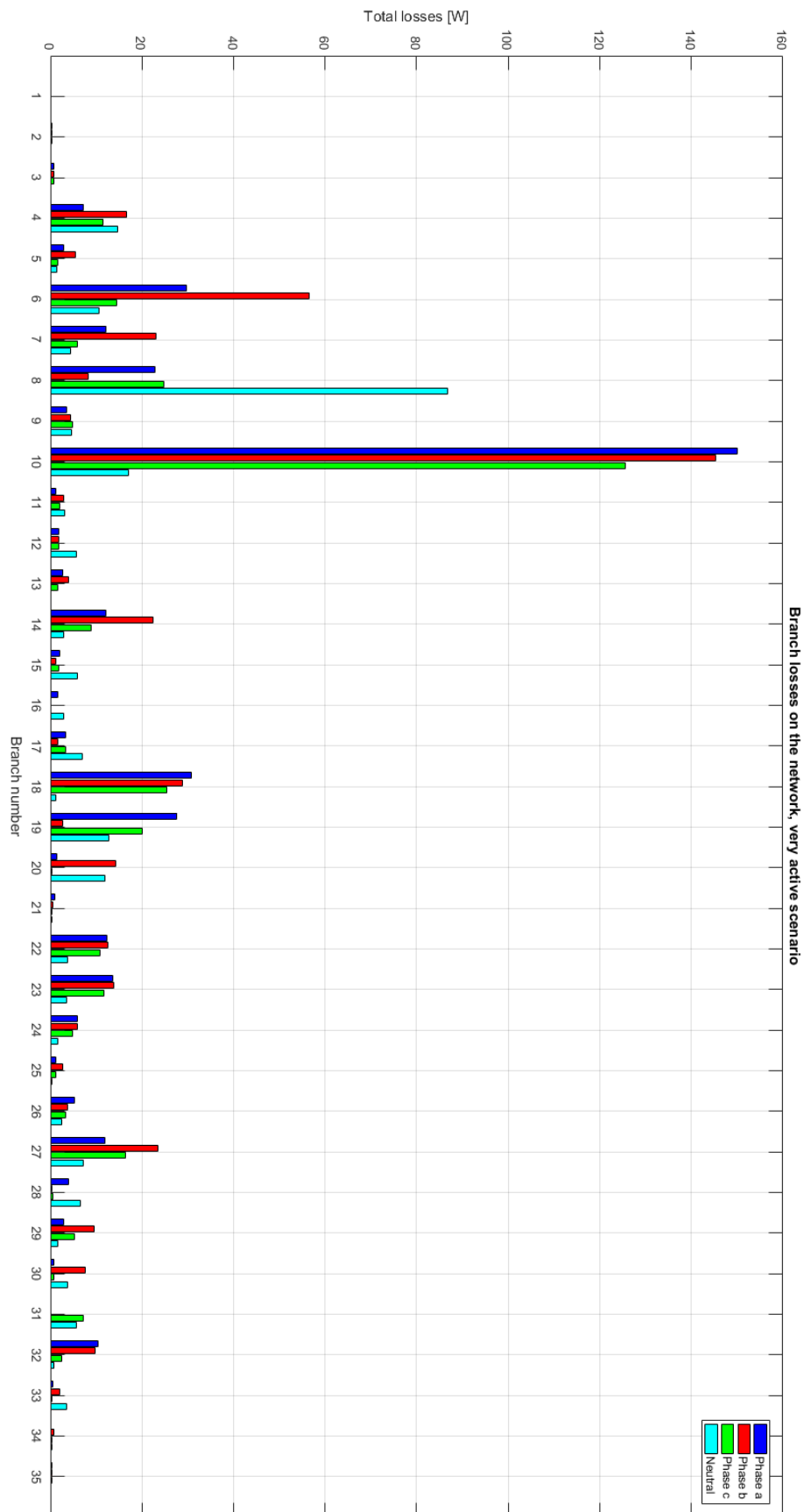


Figure 4.14: Magnitude of the branch power losses in e-distribuzione network, current inversion scenario

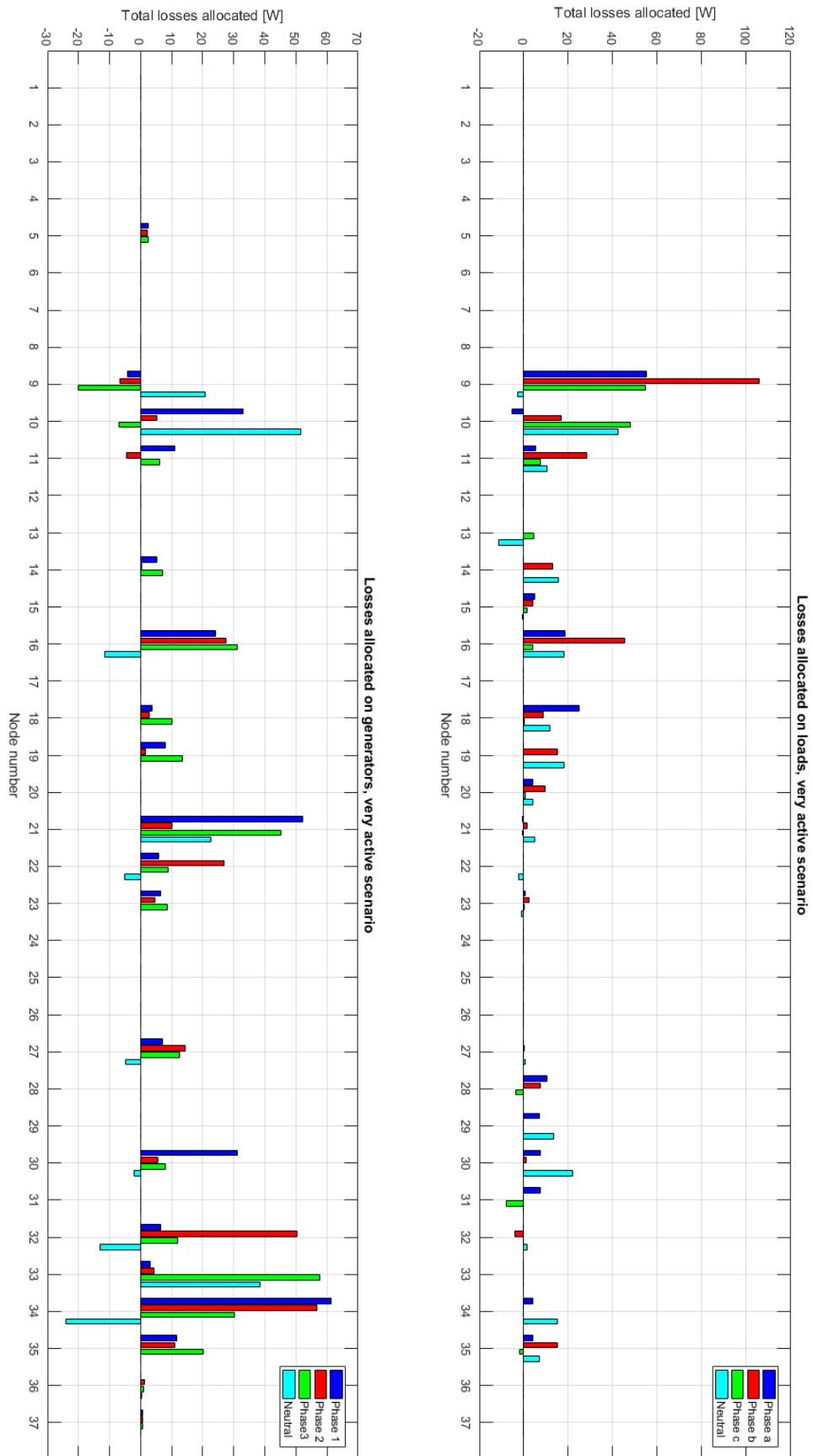


Figure 4.15: Allocated losses on e-distribuzione network, current inversion scenario

Chapter 5

Discussion on findings

In this chapter various findings will be discussed, results of the in-depth analysis of LA methods and of MPNLA in particular. The implementation process on a multi-conductor network revealed aspects of LA that are mostly still to be tackled in scientific literature.

In most cases methods are presented and discussed for a single-phase equivalent circuit representing the network. This is not a problem when the losses allocation procedure is meant for a transmission network, but when an unbalanced low voltage grid is analysed in a multi-conductor environment, loss allocation should be as well.

Amongst the procedures analysed in chapter 2, BCDLA is the only one that was extended to a three-phase unbalanced distribution system with DG in what came to be the Resistive Component-based Loss Partitioning (RCLP) method [11].

5.1 Allocation of neutral losses

When considering a three-phase unbalanced distribution system, where many users are single-phase, neutral is an essential conductor. It experiences mutual interactions with the phase conductors, it is place of induced currents but, most importantly, it has its own current, due to the unbalancing nature of the users connected.

The voltage across neutral conductor is much influenced by the earthing scheme: in IT system neutral has no point of connection with ground, thus its potential is floating. In TT and TN systems instead neutral is grounded at the star point of the transformer or even in multiple places along the network (multi-grounded system), so that its potential is fixed in these points.

Therefore neutral is a conductor that, alike phase conductors, carries a current and is defined point by point by a potential. This means that voltage drops and losses occur in this wire as well.

Multi-conductor load flow through CCI method and MPNLA treat neutral as any other conductor, thus losses are allocated to it. But the meaning of this allocation is different, and it opens more questions than it answers.

Phase losses allocated to a node or a specific user represent the losses that a load causes on the branches due to its power absorption, so that the correlation between the demand (to absorb power) and the "fee" the load has to pay to get its request met (losses to deliver power to its location) is direct and easy to understand. Phases losses are allocated to the load because they are necessary to satisfy the load needs and would not occur otherwise.

On the other hand, losses on neutral conductor are determined by the necessity of the return currents to find their way back to the star point of the transformer. Neutral

conductor is just the preferred return path (in the case of an isolated or single-grounded network, ground path has to be considered as well in a multi-grounded system).

Return (neutral) currents though are not defined by a single load in a three phase system, they are instead just a symptom of the degree of imbalance of the network.

In such a network, three different load connections are possible:

- Three-phase load without neutral connection. In this case there is no return path, due to the most basic application of Kirchhoff's laws. The load might be source of imbalances, but concerning solely the negative sequence, without bothering the zero sequence.
- Three-phase load with neutral connection. This user do have a return path for the current and, if the load is not balanced, it might generate a zero sequence component for the current.
- Single-phase load. Neutral connection is obviously required and unavoidable for the phase current to flow through the element and back.

Three-phase loads can be considered responsible if they generate imbalances in the network, and this is why they are asked a minimum degree of symmetry at the connection point by the DSO [18]. Single-phase users on the other hand cannot be considered responsible for a contingent unbalancing of the grid for both they are not able not to unbalance the system and the imbalances they cause depend on the DSO, not on them. Consistent current flows in the neutral only if the DSO is not able to locally compensate single-phase loads, which are the most present on low voltage networks, in an effective way by properly spreading them amongst the different phases.

This is why is not appropriate to allocate, and therefore assign different shares of responsibility for the losses on neutral conductor to nodes and users.

The very different nature and meaning of losses allocated to neutral is the reason why the author takes distances from the approach used by Carpaneto *et al.* [11], in which losses attributed to the return path (neutral conductor and, in case of a multi-grounded system, ground) are redistributed across amongst the three phases. A preferred practice is instead, at least in the context of this dissertation, to keep neutral allocations separated, making the DSO decide how to manage to losses on neutral and how to recover their costs.

This methodology aligns with the principle stated at the beginning of this work: losses allocation are here treated and meant as technical tools for the grid operation and planning. Therefore, now that the meaning of losses allocated to neutral has been cleared out, let us understand how they work and why they are an important tool.

Neutral losses allocation signals how much loads connected to a node are contributing to losses on neutral in the system. Since losses on neutral are due to neutral current, these allocation factors are a good indicator of imbalances related to the homopolar sequence. If a node has high positive losses allocated to its neutral, it means that it is contributing to the worsening of the network operation, whilst if the allocation is negative, then the elements on the node has a positive impact on the overall system, by lowering the current on neutral.

As already discussed, losses allocated to loads can only be positive, since the absorption of power from a phase conductor can only lead to more losses. Allocation factors for generators can be both negative (if they improve the system by providing locally power that would otherwise need to flow through branches, causing losses) and positive (if they burden the lines with more current that is not locally consumed).

Losses allocated to neutral as well can be both positive and negative, and their sign has no correlation to the sign of the phase allocation factor at the very same node.

Let us take for instance the losses allocated in the optimal scenario, fig.4.11. On node 16, there are loads on each phase, but phase *b* is much more loaded than *c*, as it is clear from the allocation. Node 16 is contributing to unbalance the network, as a result phase *b* is more loaded along the whole feeder and there is a net current flowing and a loss in the neutral, which is therefore allocated in 16 with a positive term. The meaning is that loads on node 16 have a negative impact on the system balance. Looking now at node 31, the loads are allocated losses on phase *c*. It appears to be a single-phase load connected on *c*, which is the lightest loaded phase in both the network and in the feeder to which bus 31 belongs to. Said feeder is highly loaded on phase *a* instead. Thus this load, placed in this very position on this very phase, by raising power absorbed on phase *c* is levelling and balancing the total load and, as a consequence, the whole system.

This is another reason why neutral losses allocation has a total different meaning compared to their phase equivalents. A negative factor means that more load on node 31, phase *c* is welcome, since it would have the beneficial side effect of lowering neutral losses. If we where to redistributing allocated neutral losses amongst the phases and, eventually the users, what kind of signal would that be?

Users in such a position would be incentivize to consume more: the more power they absorb, the more revenue they would have from neutral losses. Even at the net of phase losses allocated to them, they might still be pushed to consume more. This is why neutral losses has to remain, in the author's own opinion, a separated entity and a powerful tool for the DSO in planning of the grid. This information tells the DSO that, for instance, if a single-phase load were to be added to node 31, phase *c* would be a very suitable connection and a beneficial one for the whole system.

Looking at generators, fig. 4.11 tells again a similar story. Generators might be rewarded phase-wise for their service, providing locally the right amount of power for each phase, but if this results in higher imbalances for the grid, neutral losses allocated will have a positive sign (nodes 16, 28, 29, 30, 31 and 34). It might as well happen the other way around (node 14), but this is less likely, since users in low distribution network are smaller and less prone to generate big amount of power they will not consume. Node 11 and 35 are instead examples of a well dimension generator, from an overall grid losses point of view, in which all allocation factors are negative, but not hugely negative. A very negative loss allocation factor means that the impact on the system is very good, but is also much needed and much more could be done to lower losses in that point.

5.2 Allocation of transformer losses

Allocation of transformer losses is a topic that is never really discussed in scientific literature. This might be due to the fact that it is much more common to just analyse one single voltage level, but even then, if we consider the transformer at the slack bus as part of our network, as this thesis did, it is important to allocate its losses too. So is it possible to just extend the algorithm to these devices too?

After all, transformers are modelled into a $2m \times 2m$ matrix, the same dimension as any branch matrix in the network. If both transformers and branches can be represented as double m-port, there should be no reason to distinguish between the two kinds in the losses allocation process.

Yet, the implementation of MPNLA method, that originally did not take into consideration transformers, highlighted that this is not the case.

While the use of a single-phase equivalent circuit suggests that, when working in per unit, there should be no issues, in real life as well as in a multi-conductor environment, things change. The phase shift of the phasors, determined by the group belonging of the transformer, has to be taken into account. The current, when transformed from the secondary side (which is the typically the side where users are connected) to the primary, need to be properly shifted. To do so, if the network is unbalanced, an option would be to decompose the phasors at the secondary into positive, negative and zero sequences, rotate them and then recombine them. But even so, the winding structure of the transformer will most likely prevent from obtaining the correct solution. Whilst total losses on the transformer would result in the correct value, the phase by phase values would not. In fact, the assumption that phase currents correspond to winding currents is not feasible.

Let us consider the popular, in the italian distribution system, Dyn11 transformer, that is used in e-distribuzione network as well and pretend currents are flowing from secondary to primary side. Phase currents at the secondary will enter the winding, connected in a star configuration, and meet at the star point, where, if present, neutral current is injected, and cancel each other. After the magnetic circuit transfers the currents to the primary windings, they will then be splitted and combined by the triangle connection, so that phase currents are a recombination of winding ones.

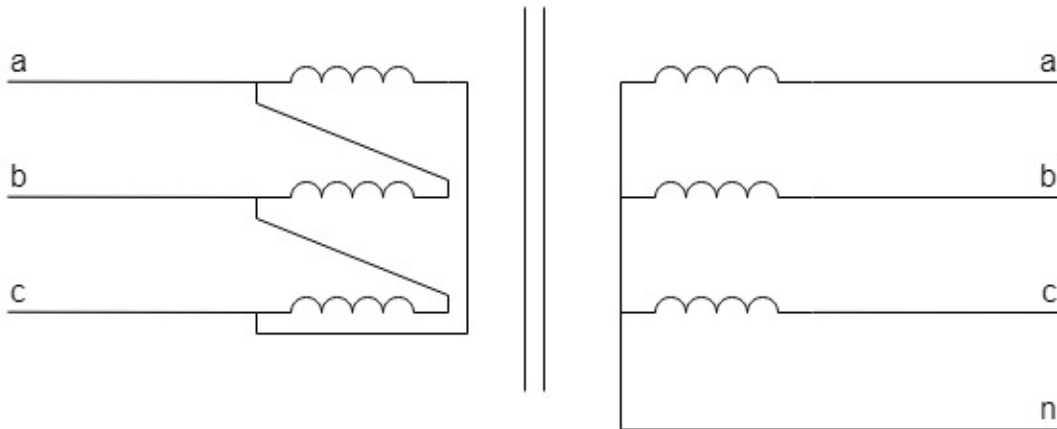


Figure 5.1: Winding configuration for a Dyn11 transformer

Let us suppose current on the secondary side of the transformer is flowing only in phase c . On the primary side current will be flowing on multiple phases and phase c current is now responsible for the losses on multiple phases on the primary side.

This makes not feasible to consider the transformer just as any other branch in the allocation process.

The solution to this issue, adopted in the implementation of MPNLA, is to rewrite the calculation of losses in the transformer as function of just its admittance matrix and the secondary currents, as presented in chapter 3. This allows for a consistent loss allocation for the transformer accordingly to the currents that are drawn at the secondary side by the loads, so that any group and winding configuration can be used seamlessly.

It is known that losses in a transformer can be divided into two main categories: winding losses (also known as copper losses, since it is the most common material for windings) and core losses (also known as iron losses, for the same reason). Copper losses can be linked mainly to voltage drop that occurs on winding, since they are impedances crossed by a current. This means that their magnitude is mainly linked to the total load connected: the more load, the more losses. Iron losses on the other hand are mainly due to the magnetization of the core, meaning that they occur even when no load is attached. Even if one winding is an open circuit and there is no net current flowing in the transformer, if the other coil is excited by a voltage, core losses will be present.

Once again, as for the neutral, different meanings in the losses lead to different allocation. Whilst copper losses can be allocated just as branch ones, iron losses deserve more attention, since they are very little dependent from the load and much more constant. If we were to allocate iron losses between users, a condition might occur in which the network is very lightly loaded and a single, small load is connected. This user, being the only current purposefully flowing, would have all core losses allocated to it, reflecting a situation which is untrue. Core losses are generated not by a single user, but by all of them, even is not connected in that specific moment in time. They have to be consider as a fee, to be distributed amongst all users, for they are able consume electricity, even if they are not. In fact, in the unlikely eventuality that no user is connected, core losses would still be present but they will not be allocated to anyone. This is the reason why iron losses have to be taken out of the allocation process and spread out between all users using the criteria that the DSO consider to be fairer.

To sum up, in a multi-conductor environment losses on the transformer have to be allocated carefully, considering both group and windings configuration. Only winding losses can be allocated to users, whilst core losses needs to be divided between users with a different, "static" criteria.

5.3 MPNLA on a multiple voltage levels network

Since the allocation of losses on the transformer differs so much from the branches that it needs a totally different approach, as shown in Appendix A, the utilization of MPNLA on networks with multiple transformers is not immediate.

As it is written, the code allows for the exact computation of the losses in multiple transformers networks but not their correct allocation. It is meant to serve the most common scenario, in which the distribution transformer is placed on the slack bus and from it all the lines start to build the radial grid.

To explain why more than one transformer needs a special treatment, let us introduce a test system.

Fig.5.2 shows a network with two different transformers, one connecting medium to low voltage and the other connecting two low voltage areas, either to compensate an excessive line voltage drop or to feed an obsolete portion of the network with a different low voltage level.

The load flow, followed by MPNLA algorithm, will calculate the correct losses for each branch of this network.

Yet, let us consider node 22, the load generates losses on phase a , since this is the phase it is connected to, at least inside the sub-area A . Transformer (16), accordingly to its group and windings configuration, might recombine currents so that current on the secondary side, phase a , will appear on the primary as a combination of different phases. Thus, current absorbed in 22 will generate losses on a conductor in one part of

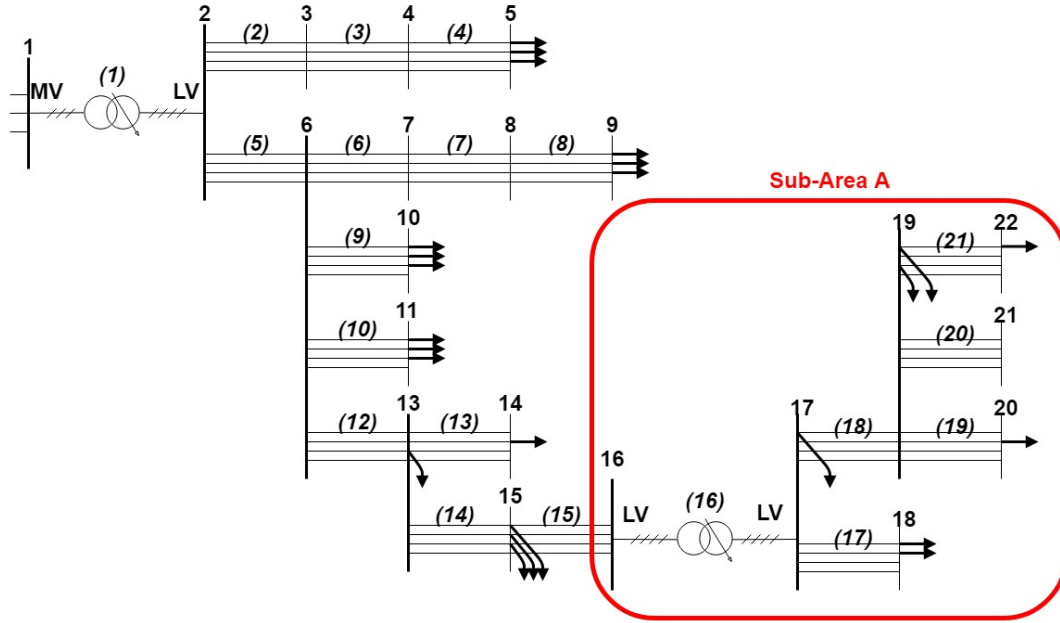


Figure 5.2: Example of a multiple transformers network

the network and on another one in a different portion of the grid. This means that one specific current will load different conductors in different portion of the grid, so that the relation between this current and the losses that it generates becomes way more complex than it used to be.

The following is the theorized solution, yet to be proven, that the author propose. Firstly, the load flow for the whole network determines the voltage phasors for node 16, which represent the slack bus for the sub-area A . At this point, a separate load flow (setting the appropriate values for the slack bus, calculated earlier on node 16) followed by MPNLA on the sub-area A will determine the loss allocation factors $L_{k,A}$ for the each k user connected downstream of the transformer (16) regarding the sub-area A . This means that the losses caused by users connect to A an A are now known.

At this point, the next step is to transpose each current \bar{I}_k absorbed by users in the sub-area A to the primary side of the transformer, as if it was absorbed directly in 16. This transformation can be easily done within the transformer losses program in MPNLA. The sum of each current $\bar{I}_{k,primary}$ will sum up into $\bar{I}_{primary}$ model an equivalent total load in 16. Now the load flow and the MPNLA algorithm for the whole grid, minus the sub-area A , can be run again placing the equivalent load in 16. Losses in the main portion of the grid will be allocated to all its users and to node 16 as well, as $L_{16,main}$.

The last step is post-processing of the allocation data for the users connected in A . Losses allocated to node 16 are proportionally divided amongst loads using the ratio between the user's current transposed to the primary side of the transformer and the total load current in 16. Each user in A will be allocated a loss L_k equal to the sum of its losses allocated in A and its share of the losses generated in the main portion of the grid.

$$L_k = L_{k,A} + \frac{\bar{I}_{k,primary}}{\bar{I}_{primary}} \cdot L_{16,main} \quad k = 17, \dots, 22 \quad (5.1)$$

It has to be noted though that this scenario represent an extremely rare situation. Most low voltage networks has a small extension, meaning the odds of having more than one transformer at the slack bus are pretty low. This justify decision not to create a rather complex automated procedure to tackle this problem and instead use a "manual" approach.

Chapter 6

Conclusions

The analysis of losses allocation processes present in literature showed that very little of them are suitable for unbalanced low voltage networks. Amongst them, Admittance Branch Current Loss Allocation (ABCLA), Branch Current Decomposition method for Loss Allocation (BCDLA) and Power Summation Method for Loss Allocation (PSMLA) have shown to be the most promising, with the first two being already successfully implemented in a multi-conductor environment [11] [13].

The implementation of MPNLA inside the CCI method for load flow have shown multiple point of strength, such as robustness, simplicity and speed.

It has also highlighted some innovative concepts regarding losses in the transformers and the meaning of neutral losses allocation.

The goal set for the ideal loss allocation method, such as consistency with the load flow results, topology-dependency, simplicity and fairness to user, are all, on different levels, met. MPNLA stands out for the aforementioned innovations and for its ability to give important technical signals. The separate allocation of losses to passive loads and generators individually, that the other methods join together instead, provides important information on the grid that would otherwise be disguised. In chapter 4 these information are used to optimize a network scenario, showing this potential ability for LA coefficients.

This thesis has demonstrated that the impact of DG on system losses is not always beneficial, it has shown a way to properly allocate transformer losses and it has stated the importance of the neutral conductor and discussed its key role in losses allocation.

On the other hand, whilst this method has shown to be technically accurate, the fairness-to-user can be improved with a post-processing stage in which load and generator losses allocated to the same user are reconciled, to take into account power self-consumed by the user.

Other future development to this work might be a further in-depth analysis of neutral allocation term, an investigation on their meaning and on their utilization in management and planning of the network.

Furthermore, since MPNLA uses the whole impedance matrix of the network and not just its real part, as suggested by Carpaneto *et al.* [6], the allocation process happens for the reactive power as well, but only the active part is then used. It might be useful in the future to consider reactive power as well, which is not a loss in the system but might be a sensible indicator to understand the role of imaginary power. In fact, since the R/X ratio is high in low voltage distribution networks, the way reactive power flow interact with voltage regulation and losses is different from transmission grids.

An optimization algorithm can be built to understand the technical signals given by the allocation factors and assist the DSO in planning and managing of the grid, deciding

for instance on which phase to connect a new user.

To conclude, there is the possibility to use the allocation terms, properly weighted and processed, to calculate economical rates, fair and transparent to the final user. This can leverage the level of responsibility and awareness of the consumer, now able to understand its impact on the grid and to lower its losses and costs, pursuing higher efficiency and sustainability.

Bibliography

- [1] K Sunderland, M Coppo, M Conlon, and R Turri. A correction current injection method for power flow analysis of unbalanced multiple-grounded 4-wire distribution networks. *Electric Power Systems Research*, 132:30–38, 2016.
- [2] Keith Sunderland, Massimiliano Coppo, Michael F Conlon, and Roberto Turri. Application of a correction current injection power flow algorithm to an unbalanced 4-wire distribution network incorporating tn-cs earthing. In *Power Engineering Conference (UPEC), 2013 48th International Universities'*, pages 1–6. IEEE, 2013.
- [3] George Constable and Bob Somerville. *A century of innovation: Twenty engineering achievements that transformed our lives*. Joseph Henry Press, 2003.
- [4] AJ Conejo, JM Arroyo, N Alguacil, and AL Guijarro. Transmission loss allocation: a comparison of different practical algorithms. *IEEE transactions on Power Systems*, 17(3):571–576, 2002.
- [5] JS Savier and Debapriya Das. Energy loss allocation in radial distribution systems: A comparison of practical algorithms. *IEEE Transactions on Power Delivery*, 24(1):260–267, 2009.
- [6] Enrico Carpaneto, Gianfranco Chicco, and Jean Sumaili Akilimali. Branch current decomposition method for loss allocation in radial distribution systems with distributed generation. *IEEE Transactions on Power Systems*, 21(3):1170–1179, 2006.
- [7] J Mutale, G Strbac, S Curcic, and N Jenkins. Allocation of losses in distribution systems with embedded generation. *IEE Proceedings-Generation, Transmission and Distribution*, 147(1):7–14, 2000.
- [8] Antonio J Conejo, Francisco D Galiana, and Ivana Kockar. Z-bus loss allocation. *IEEE Transactions on Power Systems*, 16(1):105–110, 2001.
- [9] WL Fang and HW Ngan. Succinct method for allocation of network losses. *IEE Proceedings-Generation, Transmission and Distribution*, 149(2):171–174, 2002.
- [10] E Bompard, Enrico Carpaneto, Gianfranco Chicco, and Roberto Napoli. Convergence of the backward/forward sweep method for the load-flow analysis of radial distribution systems. *International journal of electrical power & energy systems*, 22(7):521–530, 2000.
- [11] Enrico Carpaneto, Gianfranco Chicco, and Jean Sumaili Akilimali. Loss partitioning and loss allocation in three-phase radial distribution systems with distributed generation. *IEEE Transactions on Power Systems*, 23(3):1039–1049, 2008.

- [12] Metodija Atanasovski and Rubin Taleski. Power summation method for loss allocation in radial distribution networks with dg. *IEEE Transactions on Power Systems*, 26(4):2491–2499, 2011.
- [13] Nicola Bressan. Master thesis: Stima ed allocazione delle perdite nelle reti di bassa tensione attive. *Università degli studi di Padova*, 2016.
- [14] R Benato, A Paolucci, and R Turri. Power flow solution by a complex admittance matrix method. *International Transactions on Electrical Energy Systems*, 11(3):181–188, 2001.
- [15] International Electrotechnical Commission et al. Iec 60364-5-54—low voltage electrical installations—part 5-54: Selection and erection of electrical equipment—earthing arrangements, protective conductors and protective bonding conductors. *International Standard*, 2005.
- [16] Abdullah Bokhari, Ali Alkan, Rasim Dogan, Marc Diaz-Aguiló, Francisco De Leon, Dariusz Czarkowski, Zivan Zabar, Leo Birenbaum, Anthony Noel, and Resk Ebrahim Uosef. Experimental determination of the zip coefficients for modern residential, commercial, and industrial loads. *IEEE Transactions on Power Delivery*, 29(3):1372–1381, 2014.
- [17] William H Kersting. Radial distribution test feeders. In *Power Engineering Society Winter Meeting, 2001. IEEE*, volume 2, pages 908–912. IEEE, 2001.
- [18] CEI Norm. Cei 0-21: Regola tecnica di riferimento per la connessione di utenti attivi e passive alle reti bt delle imprese distributrici di energia elettrica. *Comitato Elettrotecnico Italiano, Mailand*, 2016.

List of acronyms

DG	Distributed Generation
IoT	Internet of Things
DSO	Distribution System Operator
LA	Losses Allocation
LV	Low Voltage
MV	Medium Voltage
HV	High Voltage
PR	Pro Rata
MLC	Marginal Loss Coefficients
PV	(A node defined by) Power and Voltage
PQ	(A node defined by) Power and reactive power Q
DLC	Direct Loss Coefficients
SMLA	Succinct Method for Loss Allocation
BCDLA	Branch Current Decomposition method for Loss Allocation
PSMLA	Power Summation Method for Loss Allocation
CCI	Correction Current Injection
ITL	Incremental Transmission Loss
MPNLA	Multi-Phase and Neutral conductors Loss Allocation
ABCLA	Admittance Branch Current Loss Allocation
LAG	Losses Allocated to Generators
LAL	Losses Allocated to Loads
RCLP	Resistive Component-based Loss Partitioning

Appendix A

MPNLA Matlab code

MPNLA Matlab code is presented in its implementation in the CCI load flow environment (main_CCI3F.m, version of January 2018). Two main variations have been applied:

- In CCI_LF_fast.m I_Loads and I_Gen have been calculated adding a few code lines at the end of the script, as follow, then inserted in LF structure as LF.I_Loads and LF.I_Gen.

```
Delta_I_Loads=[zeros(4,1); Icorr1];
Delta_I_Gen=[zeros(4,1); Icorr2];
I_Loads=Y_loads*E-Delta_I_Loads;
I_Gen=-Y_gen*E+Delta_I_Gen;
```

- In Matrix_Ytras.m one line of code was added inside the for cycle to obtain Y_tr matrix, then outputted for the function into the workspace.

```
Y_tr((ni*2*nph)-2*nph+1:(ni*2*nph), (ni*2*nph)-2*nph+1:(ni*2*nph))=Ybr;
```

Once the load flow is run, MPNLA.m can be launched to obtain a structure called MPNLA, in which the allocation results are stored.

```
%%%%%%%%%      MULTI-PHASE AND NEUTRAL CONDUCTORS LOSS ALLOCATION      %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This method will be used to allocate branch losses to each node of the
% network, indipendently for loads and generators.

function [MPNLA]=MPNLA(Network,LF,Sbase,Num.buses,m,Y_tr)

time=size(Network.P_profiles,1);

% Construction of the branch voltages vector
N.br=size(Network.Trafo,1)+size(Network.Branches,1);
%Transformers are counted as branches
E.br=zeros(2*m*N.br,time);
%E.br is composed as [E.in; E.out] for each instant in time (columns)

%From nodal voltages to branch voltages.
for ni=1:size(Network.Trafo,1)
    Nn1=Network.Trafo(ni,1); %From node 'Nn1'
    Nn2=Network.Trafo(ni,2); %To node 'Nn2'
```

```

E_br((ni-1)*m+1:(ni-1)*m+m, :)=LF.E((Nn1-1)*m+1:(Nn1-1)*m+m, :);
E_br(N_br*m+(ni-1)*m+1:N_br*m+(ni-1)*m+m, :)=LF.E((Nn2-1)*m+1:(Nn2...
-1)*m+1+(m-1)), :);
end

for ni=1:size(Network.Branches,1)
Nn1=Network.Branches(ni,1);
Nn2=Network.Branches(ni,2);
nj=size(Network.Trafo,1)+ni;
E_br((nj-1)*m+1:(nj-1)*m+m, :)=LF.E((Nn1-1)*m+1:(Nn1-1)*m+1+(m-1)), :);
E_br(N_br*m+(nj-1)*m+1:N_br*m+(nj-1)*m+m, :)=LF.E((Nn2-1)*m+1:(Nn2...
-1)*m+1+(m-1)), :);
end

%% Construction the current vectors for loads and generators for the branches
%% For the branches
I_brLoads=zeros(m*N_br,time);

%Nodal load currents get reshaped as currents absorbed at the terminal node
%of each branch.
for ni=1:size(Network.Trafo,1)
%Connection node for the terminal load of the brach
Nn2=Network.Trafo(ni,2);
I_brLoads((ni-1)*m+1:(ni-1)*m+m, :)= LF.I_Loads((Nn2-1)*m+1:(Nn2-1)...
*m+m, :);
end

for ni=1:size(Network.Branches,1)
%Connection node for the terminal load of the brach
Nn2=Network.Branches(ni,2);
nj=size(Network.Trafo,1)+ni;
I_brLoads((nj-1)*m+1:(nj-1)*m+m, :)= LF.I_Loads((Nn2-1)*m+1:(Nn2-1)...
*m+1+(m-1)), :);
end

%Nodal generator currents get reshaped as currents absorbed at the terminal
%node of each branch.
I_brGen=zeros(m*N_br,time);

for ni=1:size(Network.Trafo,1)
%Connection node for the terminal load of the brach
Nn2=Network.Trafo(ni,2);
I_brGen((ni-1)*m+1:(ni-1)*m+m, :)= LF.I_Gen((Nn2-1)*m+1:(Nn2-1)*m+m, :);
end

for ni=1:size(Network.Branches,1)
%Connection node for the terminal load of the brach
Nn2=Network.Branches(ni,2);
nj=size(Network.Trafo,1)+ni;
I_brGen((nj-1)*m+1:(nj-1)*m+m, :)= LF.I_Gen((Nn2-1)*m+1:(Nn2-1)*m+1 ...
+(m-1)), :);
end

%% Losses calculation for the main transformer, which is connected with the
% primary side at the slack bus.

%The function Trans_losses.m gives in the outputs the branch currents
%(I_brLoads, I_brGen) transposed at the primary side and calculates already
%the losses on the transformes for each instant in time due to each load.
[I_Loads_trans, L_trL_new]=Trans_losses(Y_tr, LF.I_Loads, I_brLoads, time, ...
Num_buses, N_br, m);

```

```

[I.Gen_trans,L_trG_new]=Trans_losses(Y_tr,LF.I.Gen,I_brGen,time,Num.buses,...
    N_br,m);

%Since we do not want to use the original ACBLA to calculate the losses on the
%transformer connected to the slack bus (the method cannot give exact results
%when there is a phase shift, e.g. with a Delta-Wye transformer), we are
%going to assign zero values to the currents on the transformer.
I_brLoads_trans=zeros(size(I_brLoads,1),2);
I_brGen_trans=zeros(size(I_brGen,1),2);

%% Construction of the incidence matrix (considering just one conductor per line)

Amono=zeros(N_br,size(Network.Buses,1));
for ni=1:size(Network.Branches,1)
    nj=size(Network.Trafo,1)+ni;
    Amono(nj,Network.Branches(ni,1))=1;
    Amono(nj,Network.Branches(ni,2))=-1;
end

%% TIME ITERATION
% Initialization of the result matrices
L_alloc_Loads=zeros(m*N_br,time);
L_alloc_Gen=zeros(m*N_br,time);
L_br_ver_reorg=zeros(m*N_br,time);

for t=1:time
    %% Construction the matrices
    % Voltages
    E_br1=zeros(m,2*N_br);
    for ni=1:N_br
        E_br1(:,(2*N_br-(2*N_br-2*ni+1)))=E_br(((ni-1)*m)+1:((ni-1)*m)...
            +m),t);
        E_br1(:,(2*N_br-(2*N_br-2*ni+1))+1)=E_br((N_br*m)+((ni-1)*m)+1)...
            :(N_br*m)+((ni-1)*m)+m),t);
    end
    E_mat=repmat(E_br1,N_br,1);

    % Currents
    I_matLoads_B=repmat(I_brLoads(:,t),1,2*(size(Network.Branches,1)));
    I_matGen_B=repmat(I_brGen(:,t),1,2*(size(Network.Branches,1)));
    I_matLoads=[I_brLoads_trans I_matLoads_B];
    I_matGen=[I_brGen_trans I_matGen_B];
    for ni=1:(N_br-1)
        I_matLoads(((ni-1)*m)+1:((ni-1)*m)+m,(2*ni)+1:(2*N_br))=0;
        I_matGen(((ni-1)*m)+1:((ni-1)*m)+m,(2*ni)+1:(2*N_br))=0;
    end

    %% Losses calculation for the branches due to each current at every node
    S_L=E_mat.*conj(I_matLoads);
    S_G=E_mat.*conj(I_matGen);
    L_brLoads=zeros(m*N_br,N_br);
    L_brGen=zeros(m*N_br,N_br);
    for ni=1:N_br
        L_brLoads(:,ni)=S_L(:,(2*N_br)-((2*N_br-ni)-(ni-1)))-S_L(:,(2*...
            N_br)-((2*N_br-ni)-ni));
        L_brGen(:,ni)=S_G(:,(2*N_br)-((2*N_br-ni)-(ni-1)))-S_G(:,(2*N_br)...
            -(2*N_br-ni)-ni));
    end

    %% Creation of the losses matrices L_brLoads_new and L_brGen_new
    L_brLoads_new=zeros(m*N_br,N_br);
    L_brGen_new=zeros(m*N_br,N_br);

```

```

%On the first column transformer losses are placed
L_brLoads_new(:,1)=L_trL_new(:,t);
L_brGen_new(:,1)=L_trG_new(:,t);

%Using the incidence matrix the code understands the topology of the
%network and copies only the right values from L_brLoads and L_brGen
%into L_brLoads_new and L_brGen_new
for nl=2:(size(Network.Branches,1)+1)
    Nrows=zeros(N_br,1);
    Nrows(nl,1)=nl;

    for nh=1:size(Nrows)
        nll=Nrows(nh,1);
        if nll>0
            [Nrows]=finder(Amono,nll,Nrows);
        end
    end

    Nrows(Nrows==0)=[];

    for nh=1:size(Nrows,1)
        nrow=Nrows(nh,1);
        L_brLoads_new((nrow-1)*m+1:nrow*m,nl)=L_brLoads((nrow-1)*m+...
            1:nrow*m,nl);
        L_brGen_new((nrow-1)*m+1:nrow*m,nl)=L_brGen((nrow-1)*m+1:nrow...
            *m,nl);
    end
end

%% Losses allocation

%L_alloc_Loads and L_alloc_Gen are the losses allocated to the node
%at the end of each branch.
L_alloc_Loads(:,t)=sum(L_brLoads_new,2);
L_alloc_Gen(:,t)=sum(L_brGen_new,2);

%From power losses, only the real part is taken
L_alloc_Loads=real(L_alloc_Loads);
L_alloc_Gen=real(L_alloc_Gen);

%% Branch Losses

L_br_ver = zeros(m,N_br);
for ni=1:N_br
    L_br_ver(:,:)=L_br_ver(:,:)+L_brLoads_new(((ni-1)*m)+1:((ni-1)*m)...
        +m,:)+L_brGen_new(((ni-1)*m)+1:((ni-1)*m)+m,:);
end

for ni=1:N_br
    L_br_ver_reorg((ni-1)*m+1:(ni-1)*m+m,t)=L_br_ver(:,ni);
end

end

%% Losses allocated to the terminal node of each branch get reshaped into
% losses allocated to each node.

L_alloc_Loads_reorg=zeros(m*(size(Network.Buses,1)),time);
L_alloc_Gen_reorg=zeros(m*(size(Network.Buses,1)),time);

```



```

for ng=1:size(Network.Trafo,1)
    Nn2=Network.Trafo(ng,2); %Terminal node of the transformer
    L_alloc_Loads_reorg((Nn2-1)*m+1:(Nn2-1)*m+m, :)=L_alloc_Loads((ng-1)...
        *m+1:(ng-1)*m+m, :);
end

for ng=1:size(Network.Branches,1)
    nj=size(Network.Trafo,1)+ng;
    Nn2=Network.Branches(ng,2); %Terminal node of the branch
    L_alloc_Loads_reorg((Nn2-1)*m+1:(Nn2-1)*m+m, :)=L_alloc_Loads((nj-1)...
        *m+1:(nj-1)*m+m, :);
end

for ng=1:size(Network.Trafo,1)
    Nn2=Network.Trafo(ng,2); %Terminal node of the transformer
    L_alloc_Gen_reorg((Nn2-1)*m+1:(Nn2-1)*m+m, :)=L_alloc_Gen((ng-1)*m+...
        1:(ng-1)*m+m, :);
end

for ng=1:size(Network.Branches,1)
    nj=size(Network.Trafo,1)+ng;
    Nn2=Network.Branches(ng,2); %Terminal node of the branch
    L_alloc_Gen_reorg((Nn2-1)*m+1:(Nn2-1)*m+m, :)=L_alloc_Gen((nj-1)*m+...
        1:(nj-1)*m+m, :);
end

%% ABCLA Structure

MPNLA.Losses=L_br_ver_reorg.*Sbase; %[W]
MPNLA.Loads_Alloc_br=L_alloc_Loads.*Sbase; %[W]
MPNLA.Gen_Alloc_br=L_alloc_Gen.*Sbase; %[W]
MPNLA.Loads_Alloc=L_alloc_Loads_reorg.*Sbase; %[W]
MPNLA.Gen_Alloc=L_alloc_Gen_reorg.*Sbase; %[W]
MPNLA.I_Loads_primary=I_Loads_trans; %[p.u.]
MPNLA.I_Gen_primary=I_Gen_trans; %[p.u.]

```

```
end
```

A function needed to run MPNLA.m is Trans_losses.m, which calculates losses on the transformers.

```

function [LF_IL_trans, L_tr_new]=Trans_losses(Y_tr, LFI_Loads, I_brLoads_sec, ...
    time, Num_buses, N_br, m)
%The function Trans_losses.m gives in the outputs the branch currents
%(I_brLoads, I_brGen) transposed at the primary side and calculates the
%losses on the transformes for each instant in time due to each load.

%% Creation of the matrices
Ypp=Y_tr(1:4,1:4);
Yps=Y_tr(1:4,5:8);
Ysp=Y_tr(5:8,1:4);
Yss=Y_tr(5:8,5:8);
Sp=Yps*(pinv(Yss));

Z_tr=pinv(Y_tr);
Zpp=Z_tr(1:4,1:4);
Zps=Z_tr(1:4,5:8);
Zsp=Z_tr(5:8,1:4);
Zss=Z_tr(5:8,5:8);
Zb=(Sp')*Zpp*Sp+(Sp')*Zps+Zsp*Sp+Zss;

```

```

%% Initialization of the result matrices
LF_IL_trans=zeros(size(LFI_Loads));
L_tr_new=zeros(m*N_br,time);

%% Transposition of the loads currents to the primary side
for ti=1:time
    for ni=1:Num_buses
        Is=LFI_Loads((ni-1)*m+1:ni*m,ti);
        Ip=Sp*Is;
        LF_IL_trans((ni-1)*m+1:ni*m,ti)=-Ip;
    end
end

%% Calculation of the losses on the transformer from the secondary side,
% using a slightly modified approach to the BCDLA (we consider the complex
% impedance, not just the real part)

for ti=1:time
    for ni=1:N_br
        Iss=I_brLoads_sec((ni-1)*m+1:ni*m,ti);
        Sloss=(conj(Iss)).*Zb*Iss;
        L_tr_new((ni-1)*m+1:ni*m,ti)=Sloss;
    end
end

end

```

The reading of the single-phase equivalent incidence matrix inside of MPNLA is carried out with the function `finder.m`.

```

function [Nrows]=finder(Amono,nll,Nrows)
%Software interface to read the single-phase incidence matrix Amono

Ncol=find((Amono(nll,:))<0);
Nrow=find((Amono(:,Ncol))>0);
for ni=1:size(Nrow,1)
    nrow=Nrow(ni,1);
    Nrows(nrow,1)=nrow;
end

end

```

Appendix B

Solutions plotter

Since the results from both load flow and losses allocation consist in thousands of complex number arranged in big matrices, a graphical representation was much needed. This goal was obtain using the following set of function.

The first script works as a user-friendly interface: just four parameters have to be selected in order to plot the wanted portion of the grid in the desired method for a specific instant in time, being also able to choose whether to plot the values from the load flow with regulations (capability and unbalanced compensation for DG, OLTC for transformers). This plotter works for every network, but part of it (topology settings) has be written to accommodate only e-distribuzione network. Plottable portions of the grid are, aside for the whole grid, feeder 1 and feeder 3.

Feeder 1 is give by the path from the root towards node 5. It was chosen as representative of a specific condition: the only user is located at the terminal node, where a balanced load and a balanced generator are connected. The generator has higher nominal power compared to the load, which make the feeder useful to analyse the current inversion in the branches. The lines are of different length, composition and impedance, which makes another interesting point.

Feeder 3 instead fed current to node 34. It was chosen because it is by far the longest feeder, its loads are fairly spread and unbalanced, as are the generators. It represent quite the opposite condition to the one represented by feeder 1.

When the grid is analysed in its entirety, an "alert" system will run in addition to the plotting. A few vectors are built and outputted, showing nodes where the lower phase voltages and maximum neutral potential are registered and their magnitude. The system also shows which branches have the more losses and their magnitude.

This "interface" is here presented:

```
%%%%%%%%%          VOLTAGES AND LOSSES PLOTTER FOR main_CCI_3F          %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% It uses the results from the main_CCI_3F.

%% INPUTS

%%%%% TIME
% You can read in main_CCI_3F.m the start date and the step duration. Each
% unit in It is a step. It=1 correspond to the start date.

itime=1; %chosen instant in time

%%%%% VOLTAGES
% You can choose whether to use:
```

```

Reg=0;          %Regulation type:
                %0 - LF.E(without any regulation)
                %1 - LF.Enew (with regulation)

%%%%% GRAPHICS
% You can choose whether to use:
Graph=0;       %Plot type:
                %0 - Segments: on the x axis nodes are displayed accordingly
                %to the distance in [m] from node 1 (works just for feeders
                %plotting).
                %1 - Bars

%%%%% LOCATION
% You can choose whether to show:
Wh=0;          %Location:
                %0 - All the network (Works for every datafile, assuming
                % just one transformer between nodes 1 and 2).
                %1 - FEEDER 1 (Works only for the Verona network)
                %3 - FEEDER 3 (Works only for the Verona network)

%% OUTPUTS

addpath('Plot');

%Choose voltages
if Reg==0
    X = LF.E;
elseif Reg==1
    X = LF.Enew;
end

% Calculate voltage sequence components
[Ud,Ui,Uo] = sequence(X(1:m:end,:)-X(m:m:end,:),X(2:m:end,:)-X(m:m:end,...
    ,X(3:m:end,:)-X(m:m:end,:),Num_buses,1:size(Network.P.profiles,1));
LF.Ud=Ud;
LF.Ui=Ui;
LF.Uo=Uo;

Num_branches = size(Network.Branches,1);

if Wh==0
    [Nalert_V] = Voltages_Plot(X,LF.Ud,LF.Ui,LF.Uo,Itime,Num_buses,Reg)
    [Nalert_L,Nalert_L_new] = Losses_Plot(LF.Losses,LF.Losses_new,Itime,...
        Num_branches)
    Allocation_Plot(MPNLA.Loads_Alloc,MPNLA.Gen_Alloc,Itime,Num_buses)
    % Nalert are vectors. Each column is a phase (in order:a, b, c,
    % neutral). First row is the node/branch number, second row is the
    % value in the node/branch.
    % Nalert_E shows the nodes of minimum voltage per phase and maximum
    % neutral potential.
    % Nalert_L shows the branches of maximum loss per phase.
elseif Wh==1
    Voltages_Plot_1(X,LF.Ud,LF.Ui,LF.Uo,Itime,Graph,Reg);
    Losses_Plot_1(LF.Losses,LF.Losses_new,Itime);
else
    Voltages_Plot_3(X,LF.Ud,LF.Ui,LF.Uo,Itime,Graph,Reg);
    Losses_Plot_3(LF.Losses,LF.Losses_new,Itime);
end

```

Here are now presented the functions used to actually plot the results.

```
function [Nalert] = Voltages_Plot(X,LFUd,LFUi,LFUo,Itime,Num_buses,Reg)
% Voltages plot for all the network.
% This function plots phase and neutral voltages and sequence decomposition.

%% Vectors initialization

% PHASE VOLTAGES

Ea=zeros(Num_buses,1);
for n=1:1:Num_buses
    Ea(n,1)=abs(X((4*(n-1)+1),Itime));
end

Eb=zeros(Num_buses,1);
for n=1:1:Num_buses
    Eb(n,1)=abs(X((4*(n-1)+2),Itime));
end

Ec=zeros(Num_buses,1);
for n=1:1:Num_buses
    Ec(n,1)=abs(X((4*(n-1)+3),Itime));
end

En=zeros(Num_buses,1);
for n=1:1:Num_buses
    En(n,1)=abs(X((4*(n-1)+4),Itime));
end

% SEQUENCES DECOMPOSITION

Ud=zeros(Num_buses,1);
for n=1:1:Num_buses
    Ud(n,1)=abs(LFUd(n,Itime));
end

Ui=zeros(Num_buses,1);
for n=1:1:Num_buses
    Ui(n,1)=abs(LFUi(n,Itime));
end

Uo=zeros(Num_buses,1);
for n=1:1:Num_buses
    Uo(n,1)=abs(LFUo(n,Itime));
end

%% ALERT SYSTEM
Nalert=zeros(2,4);

[M,I]=min(Ea-En);
Nalert(1,1)=I;
Nalert(2,1)=M;

[M,I]=min(Eb-En);
Nalert(1,2)=I;
Nalert(2,2)=M;

[M,I]=min(Ec-En);
Nalert(1,3)=I;
```

```

Nalert (2,3)=M;

[M,I]=max(En);
Nalert (1,4)=I;
Nalert (2,4)=M;

%% FIGURE PLOT

figure(5);
subplot(2,1,1);
y=[Ea-En Eb-En Ec-En];
b=bar(1:1:Num.buses,y);
xlim([0 (Num.buses+1)])
ylim([0.94 1])
xlabel('Node number');
ylabel('Nodal voltage [p.u.]');
xticks(1:1:Num.buses);
if Reg==0
    title('Voltages on all nodes without regulation')
elseif Reg==1
    title('Voltages on all nodes with regulation')
end
b(1).FaceColor = 'b';
b(2).FaceColor = 'r';
b(3).FaceColor = 'g';
grid on
legend('Phase a', 'Phase b', 'Phase c');
subplot(2,1,2)
yy=[Ui Uo];
d=bar(1:1:Num.buses,yy);
ylabel('Nodal voltage [p.u.]');
xlabel('Node number');
xticks(1:1:Num.buses);
xlim([0 (Num.buses+1)])
if Reg==0
    title('Voltages on all nodes without regulation')
elseif Reg==1
    title('Voltages on all nodes with regulation')
end
d(1).FaceColor = 'm';
d(2).FaceColor = 'c';
grid on
legend('Inverse sequence', 'Homopolar sequence');

end

function Voltages_Plot_1(X,LFUd,LFUi,LFUo,Itime,Graph,Reg)
% Voltages plot for FEEDER 1.
% This function plots phase and neutral voltages and sequence decomposition.
% FEEDER 1 includes nodes 1, 2, 3, 4, 5.

%% Vectors initialization

% PHASE VOLTAGES

Ea=zeros(5,1);
for n=1:5
Ea(n,1)=abs(X((4*(n-1)+1),Itime));
end

```

```

Eb=zeros(5,1);
for n=1:5
Eb(n,1)=abs(X((4*(n-1)+2),Itime));
end

Ec=zeros(5,1);
for n=1:5
Ec(n,1)=abs(X((4*(n-1)+3),Itime));
end

En=zeros(5,1);
for n=1:5
En(n,1)=abs(X((4*(n-1)+4),Itime));
end

% SEQUENCES DECOMPOSITION

Ud=zeros(5,1);
for n=1:5
Ud(n,1)=abs(LFUd(n,Itime));
end

Ui=zeros(5,1);
for n=1:5
Ui(n,1)=abs(LFUi(n,Itime));
end

Uo=zeros(5,1);
for n=1:5
Uo(n,1)=abs(LFUo(n,Itime));
end

%% FIGURE PLOT

if Graph==0
figure(1);
subplot(3,1,1);
plot([0 1 31 101 116],(Ea-En),'-dk');
%[0 1 31 101 116] are the distances in [m] between the nodes and node 1
ylabel('Nodal voltage [p.u.]');
xlabel('Node number');
xticks([0 1 31 101 116]);
xticklabels({'1','2','3','4','5'})
if Reg==0
title('Voltages on Feeder 1 without regulation')
elseif Reg==1
title('Voltages on Feeder 1 with regulation')
end
end
grid on
hold on
plot([0 1 31 101 116],(Eb-En),'-dg');
hold on
plot([0 1 31 101 116],(Ec-En),'-db');
legend('Phase a','Phase b','Phase c');
hold off
subplot(3,1,2);
plot([0 1 31 101 116],En,'-dr');
ylabel('Nodal voltage [p.u.]');
xlabel('Node number');
xticks([0 1 31 101 116]);
xticklabels({'1','2','3','4','5'})
if Reg==0

```

```

        title('Neutral potential on Feeder 3 without regulation')
elseif Reg==1
    title('Neutral potential on Feeder 3 with regulation')
end
grid on
subplot(3,1,3)
plot([0 1 31 101 116],(Ui),'-xb','LineWidth',1.5);
ylabel('Nodal voltage [p.u.]');
xlabel('Node number');
xticks([0 1 31 101 116]);
xticklabels({'1','2','3','4','5'})
if Reg==0
    title('Voltages on Feeder 1 without regulation')
elseif Reg==1
    title('Voltages on Feeder 1 with regulation')
end
grid on
hold on
plot([0 1 31 101 116],(Uo),'-dr');
legend('Inverse sequence','Homopolar sequence');
hold off
else
figure(1);
subplot(2,1,1);
y=[Ea-En Eb-En Ec-En];
b=bar([1 2 3 4 5],y);
ylim([0.98 1]);
xlabel('Node number');
ylabel('Nodal voltage [p.u.]');
if Reg==0
    title('Voltages on Feeder 1 without regulation')
elseif Reg==1
    title('Voltages on Feeder 1 with regulation')
end
b(1).FaceColor = 'b';
b(2).FaceColor = 'r';
b(3).FaceColor = 'g';
grid on
legend('Phase a','Phase b','Phase c');
subplot(2,1,2)
yy=[Ui Uo];
d=bar([1 2 3 4 5],yy);
ylabel('Nodal voltage [p.u.]');
xlabel('Node number');
if Reg==0
    title('Voltages on Feeder 1 without regulation')
elseif Reg==1
    title('Voltages on Feeder 1 with regulation')
end
d(1).FaceColor = 'm';
d(2).FaceColor = 'c';
grid on
legend('Inverse sequence','Homopolar sequence');
end
end
end

```



```

function Voltages_Plot_3(X, LFUd, LFUi, LFUo, Itime, Graph, Reg)
% Voltages plot for FEEDER 3.
% This function plots phase and neutral voltages and sequence decomposition.
% FEEDER 3 includes nodes 1, 2, 12, 20, 24, 25, 26, 28, 29, 31, 34.

%% Vectors initialization

% PHASE VOLTAGES

Ea=zeros(34,1);
for n=[1 2 12 20 24 25 26 28 29 31 34]
    Ea(n,1)=abs(X((4*(n-1)+1), Itime));
end

for n=[3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 8 10 11 11]
    if Ea(n,1)==0
        Ea(n,:)=[];
    end
end

Eb=zeros(34,1);
for n=[1 2 12 20 24 25 26 28 29 31 34]
    Eb(n,1)=abs(X((4*(n-1)+2), Itime));
end

for n=[3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 8 10 11 11]
    if Eb(n,1)==0
        Eb(n,:)=[];
    end
end

Ec=zeros(34,1);
for n=[1 2 12 20 24 25 26 28 29 31 34]
    Ec(n,1)=abs(X((4*(n-1)+3), Itime));
end

for n=[3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 8 10 11 11]
    if Ec(n,1)==0
        Ec(n,:)=[];
    end
end

En=zeros(34,1);
for n=[1 2 12 20 24 25 26 28 29 31 34]
    En(n,1)=abs(X((4*(n-1)+4), Itime));
end

for n=[3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 8 10 11 11]
    if En(n,1)==0
        En(n,:)=[];
    end
end

% SEQUENCES DECOMPOSITION

Ud=zeros(34,1);
for n=[1 2 12 20 24 25 26 28 29 31 34]
    Ud(n,1)=abs(LFUd(n, Itime));
end

for n=[3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 8 10 11 11]
    if Ud(n,1)==0

```

```

        Ud(n,:)=[];
    end
end

Ui=zeros(34,1);
for n=[1 2 12 20 24 25 26 28 29 31 34]
    Ui(n,1)=abs(LFUi(n,itime));
end

for n=[3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 8 10 11 11]
    if Ui(n,1)==0
        Ui(n,:)=[];
    end
end

Uo=zeros(34,1);
for n=[1 2 12 20 24 25 26 28 29 31 34]
    Uo(n,1)=abs(LFUo(n,itime));
end

for n=[3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 8 10 11 11]
    if Uo(n,1)==0
        Uo(n,:)=[];
    end
end

%% FIGURE PLOT

if Graph==0
    figure(3);
    subplot(3,1,1);
    plot([0 5 171 237 315 372 392 413 436 450 492],(Ea-En),'-dk');
    % [0 5 171 237 315 372 392 413 436 450 492] are the distances in [m]
    % between the nodes and node 1
    ylabel('Nodal voltage [p.u.]');
    xlabel('Node number');
    xticks([0 5 171 237 315 372 392 413 436 450 492]);
    xticklabels({'1','2','12','20','24','25','26','28','29','31','34'})
    if Reg==0
        title('Voltages on Feeder 3 without regulation')
    elseif Reg==1
        title('Voltages on Feeder 3 with regulation')
    end
    grid on
    hold on
    plot([0 5 171 237 315 372 392 413 436 450 492],(Eb-En),'-dg');
    hold on
    plot([0 5 171 237 315 372 392 413 436 450 492],(Ec-En),'-db');
    legend('Phase a','Phase b','Phase c');
    hold off
    subplot(3,1,2)
    plot([0 5 171 237 315 372 392 413 436 450 492],En,'-dr');
    ylabel('Nodal voltage [p.u.]');
    xlabel('Node number');
    xticks([0 5 171 237 315 372 392 413 436 450 492]);
    xticklabels({'1','2','12','20','24','25','26','28','29','31','34'})
    if Reg==0
        title('Neutral potential on Feeder 3 without regulation')
    elseif Reg==1
        title('Neutral potential on Feeder 3 with regulation')
    end
    grid on
end

```

```

subplot(3,1,3)
plot([0 5 171 237 315 372 392 413 436 450 492],(Ui),'-xb','LineWidth',1.5);
ylabel('Nodal voltage [p.u.]');
xlabel('Node number');
xticks([0 5 171 237 315 372 392 413 436 450 492]);
xticklabels({'1','2','12','20','24','25','26','28','29','31','34'})
if Reg==0
    title('Voltages on Feeder 3 without regulation')
elseif Reg==1
    title('Voltages on Feeder 3 with regulation')
end
grid on
hold on
plot([0 5 171 237 315 372 392 413 436 450 492],(Uo),'-dr');
legend('Inverse sequence','Homopolar sequence');
hold off
else
figure(3);
subplot(2,1,1);
y=[Ea-En Eb-En Ec-En];
b=bar([1 2 3 4 5 6 7 8 9 10 11],y);
ylim([0.95 1]);
xlabel('Node number');
ylabel('Nodal voltage [p.u.]');
xticklabels({'1','2','12','20','24','25','26','28','29','31','34'})
if Reg==0
    title('Voltages on Feeder 3 without regulation')
elseif Reg==1
    title('Voltages on Feeder 3 with regulation')
end
b(1).FaceColor = 'b';
b(2).FaceColor = 'r';
b(3).FaceColor = 'g';
grid on
legend('Phase 1','Phase 2','Phase3');
subplot(2,1,2)
yy=[Ui Uo];
d=bar([1 2 3 4 5 6 7 8 9 10 11],yy);
ylabel('Nodal voltage [p.u.]');
xlabel('Node number');
xticklabels({'1','2','12','20','24','25','26','28','29','31','34'})
if Reg==0
    title('Voltages on Feeder 3 without regulation')
elseif Reg==1
    title('Voltages on Feeder 3 with regulation')
end
d(1).FaceColor = 'm';
d(2).FaceColor = 'c';
grid on
legend('Inverse sequence','Homopolar sequence');
end
end
end

```

```

function [Nalert,Nalert_new] = Losses_Plot(LFLosses,LFLosses_new,Itime,Num_branches)
% Losses plot for all the network.
% This function plots losses with and without regulation.

%% Vectors initialization

% PHASE LOSSES w/o regulation

Lossesa=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesa(n-1,1)=abs(LFLosses((4*(n-1)+1),Itime));
end

Lossesb=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesb(n-1,1)=abs(LFLosses((4*(n-1)+2),Itime));
end

Lossesc=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesc(n-1,1)=abs(LFLosses((4*(n-1)+3),Itime));
end

Lossesn=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesn(n-1,1)=abs(LFLosses((4*(n-1)+4),Itime));
end

% PHASE LOSSES w/ regulation

Lossesa_new=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesa_new(n-1,1)=abs(LFLosses_new((4*(n-1)+1),Itime));
end

Lossesb_new=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesb_new(n-1,1)=abs(LFLosses_new((4*(n-1)+2),Itime));
end

Lossesc_new=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesc_new(n-1,1)=abs(LFLosses_new((4*(n-1)+3),Itime));
end

Lossesn_new=zeros(Num_branches,1);
for n=2:1:(Num_branches+1)
    Lossesn_new(n-1,1)=abs(LFLosses_new((4*(n-1)+4),Itime));
end

%% ALERT SYSTEM
Nalert=zeros(2,4);

[M,I]=max(Lossesa);
Nalert(1,1)=I;
Nalert(2,1)=M;

[M,I]=max(Lossesb);
Nalert(1,2)=I;
Nalert(2,2)=M;

```

```

[M, I]=max(Lossesc);
Nalert(1,3)=I;
Nalert(2,3)=M;

[M, I]=max(Lossesn);
Nalert(1,4)=I;
Nalert(2,4)=M;

Nalert_new=zeros(2,4);

[M, I]=max(Lossesa_new);
Nalert_new(1,1)=I;
Nalert_new(2,1)=M;

[M, I]=max(Lossesb_new);
Nalert_new(1,2)=I;
Nalert_new(2,2)=M;

[M, I]=max(Lossesc_new);
Nalert_new(1,3)=I;
Nalert_new(2,3)=M;

[M, I]=max(Lossesn_new);
Nalert_new(1,4)=I;
Nalert_new(2,4)=M;

%% FIGURE PLOT

figure(6);
subplot(2,1,1);
y=[Lossesa Lossesb Lossesc Lossesn];
b=bar(1:1:Num_branches,y);
xlabel('Branch number');
ylabel('Total losses [W]');
xticks(1:1:Num_branches);
xlim([0 (Num_branches+1)])
title('Branch losses on the network without regulation')
b(1).FaceColor = 'b';
b(2).FaceColor = 'r';
b(3).FaceColor = 'g';
b(4).FaceColor = 'c';
grid on
legend('Phase a', 'Phase b', 'Phase c', 'Neutral');
subplot(2,1,2)
yy=[Lossesa_new Lossesb_new Lossesc_new Lossesn_new];
d=bar(1:1:Num_branches,yy);
ylabel('Total losses [W]');
xlabel('Branch number');
xticks(1:1:Num_branches);
xlim([0 (Num_branches+1)])
title('Branch losses on the network with regulation')
d(1).FaceColor = 'b';
d(2).FaceColor = 'r';
d(3).FaceColor = 'g';
d(4).FaceColor = 'c';
grid on
legend('Phase a', 'Phase b', 'Phase c', 'Neutral');
end

```

```

function Losses_Plot_1( LFLosses,LFLosses_new,Itime)
% Losses plot for FEEDER 1.
% This function plots losses with and without regulation.
% FEEDER 1 includes nodes 1, 2, 3, 4, 5.

%% Vectors initialization

% PHASE LOSSES w/o regulation

Lossesa=zeros(3,1);
for n=2:4
Lossesa(n-1,1)=abs(LFLosses((4*(n-1)+1),Itime));
end

Lossesb=zeros(3,1);
for n=2:4
Lossesb(n-1,1)=abs(LFLosses((4*(n-1)+2),Itime));
end

Lossesc=zeros(3,1);
for n=2:4
Lossesc(n-1,1)=abs(LFLosses((4*(n-1)+3),Itime));
end

Lossesn=zeros(3,1);
for n=2:4
Lossesn(n-1,1)=abs(LFLosses((4*(n-1)+4),Itime));
end

% PHASE LOSSES w/ regulation

Lossesa_new=zeros(3,1);
for n=2:4
Lossesa_new(n-1,1)=abs(LFLosses_new((4*(n-1)+1),Itime));
end

Lossesb_new=zeros(3,1);
for n=2:4
Lossesb_new(n-1,1)=abs(LFLosses_new((4*(n-1)+2),Itime));
end

Lossesc_new=zeros(3,1);
for n=2:4
Lossesc_new(n-1,1)=abs(LFLosses_new((4*(n-1)+3),Itime));
end

Lossesn_new=zeros(3,1);
for n=2:4
Lossesn_new(n-1,1)=abs(LFLosses_new((4*(n-1)+4),Itime));
end

%% FIGURE PLOT

figure(2);
subplot(2,1,1);
y=[Lossesa Lossesb Lossesc Lossesn];
b=bar([1 2 3],y);
xlabel('Branch number');
ylabel('Total losses [W]');
xticklabels({'2 (2-3)', '3(3-4)', '4(4-5)'});
title('Branch losses on Feeder 1 without regulation')
b(1).FaceColor = 'b';

```

```

b(2).FaceColor = 'r';
b(3).FaceColor = 'g';
b(4).FaceColor = 'c';
grid on
legend('Phase a', 'Phase b', 'Phase c', 'Neutral');
subplot(2,1,2)
yy=[Lossesa_new Lossesb_new Lossesc_new Lossesn_new];
d=bar([1 2 3],yy);
ylabel('Total losses [W]');
xlabel('Branch number');
xticklabels({'2 (2-3)', '3(3-4)', '4(4-5)'});
title('Branch losses on Feeder 1 with regulation')
d(1).FaceColor = 'b';
d(2).FaceColor = 'r';
d(3).FaceColor = 'g';
d(4).FaceColor = 'c';
grid on
legend('Phase a', 'Phase b', 'Phase c', 'Neutral');
end

```

```

function Losses_Plot_3( LFLosses, LFLosses_new, Itime)
% Losses plot for FEEDER 3.
% This function plots losses with and without regulation.
% FEEDER 3 includes nodes 1, 2, 12, 20, 24, 25, 26, 28, 29, 31, 34.

%% Vectors initialization

% PHASE LOSSES w/o regulation

Lossesa=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesa(n,1)=abs(LFLosses((4*(n-1)+1), Itime));
end

for n=[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesa(n,1)==0
        Lossesa(n,:)=[];
    end
end

Lossesb=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesb(n,1)=abs(LFLosses((4*(n-1)+2), Itime));
end

for n=[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesb(n,1)==0
        Lossesb(n,:)=[];
    end
end

Lossesc=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesc(n,1)=abs(LFLosses((4*(n-1)+3), Itime));
end

for n=[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesc(n,1)==0
        Lossesc(n,:)=[];
    end
end

```

```

end

Lossesn=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesn(n,1)=abs(LFLosses((4*(n-1)+4),Itime));
end

for n=[1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesn(n,1)==0
        Lossesn(n,:)=[];
    end
end

% PHASE LOSSES w/ regulation

Lossesa_new=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesa_new(n,1)=abs(LFLosses_new((4*(n-1)+1),Itime));
end

for n=[1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesa_new(n,1)==0
        Lossesa_new(n,:)=[];
    end
end

Lossesb_new=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesb_new(n,1)=abs(LFLosses_new((4*(n-1)+2),Itime));
end

for n=[1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesb_new(n,1)==0
        Lossesb_new(n,:)=[];
    end
end

Lossesc_new=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesc_new(n,1)=abs(LFLosses_new((4*(n-1)+3),Itime));
end

for n=[1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesc_new(n,1)==0
        Lossesc_new(n,:)=[];
    end
end

Lossesn_new=zeros(33,1);
for n=[11 19 23 24 25 27 28 30 33]
    Lossesn_new(n,1)=abs(LFLosses_new((4*(n-1)+4),Itime));
end

for n=[1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 6 8 9 9]
    if Lossesn_new(n,1)==0
        Lossesn_new(n,:)=[];
    end
end

%% FIGURE PLOT

```



```

figure(4);
subplot(2,1,1);
y=[Lossesa Lossesb Lossesc Lossesn];
b=bar([1 2 3 4 5 6 7 8 9],y);
xlabel('Branch number');
ylabel('Total losses [W]');
xticklabels({'11 (2-12)', '19 (12-20)', '23 (20-24)', '24 (24-25)', ...
            '25 (25-26)', '27 (26-28)', '28 (28-29)', '30 (29-31)', '33 (31-34)'});
title('Losses on Feeder 3 without regulation')
b(1).FaceColor = 'b';
b(2).FaceColor = 'r';
b(3).FaceColor = 'g';
b(4).FaceColor = 'c';
grid on
legend('Phase a', 'Phase b', 'Phase c', 'Neutral');
subplot(2,1,2)
yy=[Lossesa.new Lossesb.new Lossesc.new Lossesn.new];
d=bar([1 2 3 4 5 6 7 8 9],yy);
ylabel('Total losses [W]');
xlabel('Branch number');
xticklabels({'11 (2-12)', '19 (12-20)', '23 (20-24)', '24 (24-25)', ...
            '25 (25-26)', '27 (26-28)', '28 (28-29)', '30 (29-31)', '33 (31-34)'});
title('Losses on Feeder 3 with regulation')
d(1).FaceColor = 'b';
d(2).FaceColor = 'r';
d(3).FaceColor = 'g';
d(4).FaceColor = 'c';
grid on
legend('Phase a', 'Phase b', 'Phase c', 'Neutral');

end

```

```

function Allocation_Plot (MPNLA_L,MPNLA_G,Itime,Num_buses)
% Losses allocation plot for all the network.

%% Vectors initialization

% For loads
Alloc_aL=zeros(Num_buses,1);
for n=1:1:Num_buses
    Alloc_aL(n,1)=MPNLA_L((4*(n-1)+1),Itime);
end

Alloc_bL=zeros(Num_buses,1);
for n=1:1:Num_buses
    Alloc_bL(n,1)=MPNLA_L((4*(n-1)+2),Itime);
end

Alloc_cL=zeros(Num_buses,1);
for n=1:1:Num_buses
    Alloc_cL(n,1)=MPNLA_L((4*(n-1)+3),Itime);
end

Alloc_nL=zeros(Num_buses,1);
for n=1:1:Num_buses
    Alloc_nL(n,1)=MPNLA_L((4*(n-1)+4),Itime);
end

% For generators
Alloc_aG=zeros(Num_buses,1);

```

```

for n=1:1:Num_buses
    Alloc_aG(n,1)=MPNLA_G((4*(n-1)+1),Itime);
end

Alloc_bG=zeros(Num_buses,1);
for n=1:1:Num_buses
    Alloc_bG(n,1)=MPNLA_G((4*(n-1)+2),Itime);
end

Alloc_cG=zeros(Num_buses,1);
for n=1:1:Num_buses
    Alloc_cG(n,1)=MPNLA_G((4*(n-1)+3),Itime);
end

Alloc_nG=zeros(Num_buses,1);
for n=1:1:Num_buses
    Alloc_nG(n,1)=MPNLA_G((4*(n-1)+4),Itime);
end

%% FIGURE PLOT

figure(7);
subplot(2,1,1);
y=[Alloc_aL Alloc_bL Alloc_cL Alloc_nL];
b=bar(1:1:Num_buses,y);
xlabel('Node number');
ylabel('Total losses allocated [W]');
xticks(1:1:Num_buses);
xlim([0 (Num_buses+1)])
title('Losses allocated on loads')
b(1).FaceColor = 'b';
b(2).FaceColor = 'r';
b(3).FaceColor = 'g';
b(4).FaceColor = 'c';
grid on
legend('Phase a', 'Phase b', 'Phase c', 'Neutral');
subplot(2,1,2);
y=[Alloc_aG Alloc_bG Alloc_cG Alloc_nG];
d=bar(1:1:Num_buses,y);
xlabel('Node number');
ylabel('Total losses allocated [W]');
xticks(1:1:Num_buses);
xlim([0 (Num_buses+1)])
title('Losses allocated on generators')
d(1).FaceColor = 'b';
d(2).FaceColor = 'r';
d(3).FaceColor = 'g';
d(4).FaceColor = 'c';
grid on
legend('Phase 1', 'Phase 2', 'Phase3', 'Neutral');
end

```