Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

PROJECTE O TESINA D'ESPECIALITAT

Títol

REAL-TIME SIMULATION OF LOAD-FLOW PROBLEMS IN ELECTRICAL GRIDS

727-TES-CA-6290

Autor/a

MARC TORNER SOLÉ

Tutor/a

PEDRO DÍEZ MEJÍA, NÚRIA PARÉS MARINÉ

Departament

MA III – DEPARTAMENT DE MATEMÀTICA APLICADA III

Intensificació

ENGINYERIA COMPUTACIONAL

Data

JUNY 2014

REAL-TIME SIMULATION OF LOAD-FLOW PROBLEMS IN ELECTRICAL GRIDS

Author: Marc Torner Solé
Tutors: Pedro Díez Mejía and Núria Parés Mariné

ABSTRACT

Key words: Real-time simulation, load-flow problems, electrical grids, Smart Grids, Distributed Generation, Principal Component Analysis.

In the present situation, the implementation of Smart Grids is one of the pressing challenges in the energy field, both for the improvement in efficiency it implies, as for its positive effects on the environment, since they facilitate the massive incorporation of renewable energy into the system.

One problem that arises in the management of these networks is to simulate in real-time their associated load-flow problems, which allow the knowledge of the grid status and to respond in real time to manage power demands.

This thesis examines how losses are evaluated in a network, that is, the difference between the generated and consumed power, when a photovoltaic generator is incorporated in one of its nodes. The needed calculations are explained, as well as the nonlinear system to solve, taking as test system a grid consisting of a three-phase generator, a transformer, and 100 equidistant nodes.

With characterizations of demand and the solar intensity as a base, annual losses simulations will be performed, by means of a calculation code which has been developed at *Laboratori de Càlcul Numèric* in UPC University. The results this code provides will be compared with the ones OpenDSS provides (specific software for the evaluation of electrical networks), in order to validate it.

The time of calculation needed to perform these simulations is high. To solve this, some ways to reduce the input data and the amount of times to solve the non-linear system of equations are studied.

A procedure for obtaining annual losses in the grid reducing the computational cost associated to the calculations is defined. Each year simulation involves solving a system of nonlinear equations 8760 times. Based on the performance of Principal Component Analysis, it will be attempted to reduce the amount of calculations to be performed to a significantly lower value, assessing how this reduction affects to the precision obtained.

SIMULACIÓ A TEMPS REAL DE PROBLEMES DE FLUXOS DE CÀRREGA EN SISTEMES ELECTRICS

Autor: Marc Torner Solé
Tutors: Pedro Díez Mejía i Núria Parés Mariné

RESUM

Paraules clau: Simulació a temps real, problemes de fluxos de càrrega, sistemes elèctrics, Smart Grids, Generació Distribuïda, Anàlisi de Components Principals.

En el context actual en què ens trobem, la implantació de les anomenades Smart Grids és un dels reptes prioritaris en el camp energètic, tant per la millora en la eficiència que aquestes suposen, com pels seus efectes positius en el medi ambient, ja que faciliten la incorporació massiva d'energies renovables en el sistema.

Un dels problemes que sorgeix en la gestió d'aquestes xarxes és el de poder simular en temps real els problemes de fluxos de càrrega associats, que permeten conèixer l'estat de la xarxa i donar una resposta a temps real per a gestionar les demandes elèctriques.

En aquesta tesina s'estudia com s'avaluen les pèrdues en una xarxa, és a dir, la diferència entre la potència generada i potència consumida, quan s'incorpora un generador fotovoltaic en un dels seus nodes. S'expliquen els càlculs a realitzar i el sistema no lineal a resoldre, prenent com a base una xarxa trifàsica formada per un generador, un transformador, i 100 nodes equidistants.

A partir de caracteritzacions de la demanda i de la intensitat solar, es faran simulacions de les pèrdues al llarg de tot un any, mitjançant un codi de càlcul desenvolupat al *Laboratori de Càlcul Numèric* de la UPC. Es compararà els resultats que aquest codi ofereix amb els que proporciona el software OpenDSS (programa específic per a l'avaluació de xarxes elèctriques àmpliament contrastat), per tal de validar-lo.

El temps de càlcul necessari per realitzar aquestes simulacions és elevat. Per resoldre això, s'estudien maneres de reduir la informació d'entrada i el número de vegades a resoldre el sistema no lineal d'equacions.

Es defineix un procediment que permeti obtenir les pèrdues anuals de la xarxa disminuint el cost computacional que impliquen els càlculs. Cada simulació anual implica resoldre un sistema no lineal d'equacions en 8760 ocasions. En base a la realització d'Anàlisi de Components Principals, es tractarà de reduir a una xifra significativament inferior la quantitat de càlculs a realitzar, valorant com aquesta reducció afecta a la precisió obtinguda.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION..1

1.1 Motivation 1

1.2 Objectives 1

1.3 Concepts 2

1.3.1 Smart Grids 2

1.3.2 Distributed Generation 2

1.3.3 The electrical system 3

CHAPTER 2. LOSSES CALCULATION IN SMART GRIDS 5

2.1 Software: OpenDSS 5

2.2 Definitions..9

2.3 Grid features 10

2.4 Equations 10

2.5 Newton Raphson Method 14

2.6 Losses calculation..15

2.7 Test system 16

2.8 Validation of the code 19

CHAPTER 3. MODEL REDUCTION TECHNIQUES 21

3.1 Principal Component Analysis 21

3.2 Reduced Basis strategy..26

CHAPTER 4. NUMERICAL EXAMPLES 27

4.1. Reduction of the input data 27

4.1.1. Analysis of the demand load curves 28

4.1.2. Demand curves reduction 32

4.2. Reduction at the solver 35

4.2.1. Avoidable losses calculations 36

4.2.2. PCA-based reduction 40

4.2.2.1. Procedure 40

4.2.2.2. Subgroups definition 43

4.2.2.3. Computation 46

4.2.2.4. Validation...47

4.2.2.5. Sample size variation..48

CHAPTER 5. CONCLUSIONS..51

CHAPTER 6. REFERENCES..53

TABLE OF FIGURES

Figure 1.1. Electricity generation, transmission and distribution system 3

Figure 2.1. OpenDSS user interface 6

Figure 2.2. Basic code example 6

Figure 2.3. Registration of COM server in Windows 7

Figure 2.4. Function DSSStartup.m to start OpenDSS from Matlab 7

Figure 2.5. Diagram of a grid 10

Figure 2.6. Test system...16

Figure 2.7. Test system diagram when a DG source is added.......................17

Figure 2.8. Example of a load curve...18

Figure 2.9. Load curve associated to the photovoltaic generator................19

Figure 2.10. Random values of DG location and power 19

Figure 2.11. Distribution of losses depending on DG position and power....20

Figure 3.1. Example of 3-dimensional data representation..........................25

Figure 3.2. Projection of data in Principal Components axis 25

Figure 4.1. Load curves along 1 January 28

Figure 4.2. Load curves along 6 January 29

Figure 4.3. Load curves evolution along Mondays at 01:00 30

Figure 4.4. Load curves evolution along Mondays at 07:00 30

Figure 4.5. Load curves evolution along Mondays at 22:00 31

Figure 4.6. Load curves evolution along Mondays at 22:00 31

Figure 4.7. New random values of DG location and power 39

TABLE OF TABLES

Table 2.1. Results using OpenDSS and the Matlab code 20

Table 4.1. Load curve associated to each node..27

Table 4.2. Load curves multiplying factors..32

Table 4.3. Errors due to load curves simplification 34

Table 4.4. Losses using the 3 ways of calculating them...............................34

Table 4.5. Absolute and relative error 35

Table 4.6. Distribution of the 2948 hours along the year............................36

Table 4.7. Hours with no need of losses calculation 37

Table 4.8. Losses using the 3 methods for 3 random pairs of data..............38

Table 4.9. Absolute and relative error 39

Table 4.10. Validation results 40

Table 4.11. Groups P01 to P10 definition 41

Table 4.12. Results of applying PCA to groups P01 to P10 42

Table 4.13. Results of applying PCA to the remaining groups 44

Table 4.14. Results of applying PCA to the 3rd classification groups 45

Table 4.15. Average error and representative hour for each group 46

Table 4.16. Results for the sample vectors 47

Table 4.17. Validation results 47

Table 4.18. Group analysis with new 20 solutions sample 49

Table 4.19. Results with 20 solutions sample 49

Table 4.20. Comparison between different sample results.........................50

Table 4.21. Results with a 2 solutions sample...50

CHAPTER 1. INTRODUCTION

1.1. Motivation

One of the outstanding topics in energetic management is the deployment of Smart Grids. This is not an easy task, since a big change in the electrical system needs to be accomplished. However, in the last years there is the wish of transforming the electrical system into a smarter one, and the overwhelming challenges that these transformations imply are showing up.

One of the main resulting problems when managing these grids is being able to simulate in real-time their associated load-flow problems, which allow the knowledge of the grid status and permits giving a real-time answer to the electrical demands.

Inclusion of Generated Distribution into a grid has affects to its behavior. Calculating these effects and the performance of the grid in an effective way is still a challenge. The procedures that are used today can be enhanced, and this will be studied in this thesis.

Managing with Smart Grids implies working with an incredibly large amount of data, so finding effective ways to handle this with reasonable computing times is essential in order to its correct development.

1.2. Objectives

In this thesis it will be explained how losses in electrical grids are nowadays calculated. An introduction to a software used with this purpose will be carried out, and also the formulation that allows the losses calculations in load-flow problems will be presented.

Thereupon, a way of coding these calculations with Matlab will be analyzed and its results will be checked by comparing them with the ones that OpenDSS (specific software for grid analysis) offers.

Once these Matlab code is checked, model reduction techniques will be discussed. The Principal Components Analysis (PCA) technique will be presented and applied to input data in order to achieve dimensionality reduction. Other ways of reducing the computational cost of the calculations will be explored.

The main objective of this report is to contribute to finding more effective ways of calculating losses in electrical grids when real-time simulations of load-flow problems are carried out.

1.3. Concepts

To start being familiar with the topics presented in this thesis, some useful concepts will be introduced:

1.3.1. Smart Grids

Smart grids have an essential role in the process of transforming the functionality of the present electricity transmission and distribution grids so that they are able to provide a user-oriented service, supporting the achievement environmental targets and guaranteeing high security, quality and economic efficiency of electricity supply in a market environment.

If smart grids are about to become present, massive renewable integration and power energy storage technologies will have to be deployed. Energy efficiency will have to be a general driving vector and demand will become an active player within the electrical system. These latter drivers will require far-reaching changes in the area of distribution networks and will determine modifications in system operation, with consequent impact on design, planning and operation of transmission networks.

Renewable generation will increasingly affect electricity networks. In particular, large wind farms (possibly offshore) will be connected to transmission networks; in addition, many distributed generation units, mainly fed by renewable energy sources (photovoltaic, small wind, biomasses, etc) will be hosted by distribution networks.

1.3.2. Distributed Generation

The implementation of smart grids in our electrical system must lead to the massive integration of renewable energy. The renewable energy generation centers which can be distributed along the electrical network are known as Distributed Generation. These centers are located close to the users of the grid, so losses of energy decrease due to lower travelling distances. Indeed, these renewable energy generators can provide useful energy at peak demand moments, avoiding the need of grid enlargement investments.

Supporting distributed generation means, in addition, having better grids from an environmental point of view. Taking into account the environmental impacts and trying to reduce them is of vital importance in order to achieve a more sustainable development.

1.3.3. The electrical system

When electricity is transferred from generation plants to customers a large structure with many components is required. This structure is made up with three sub-systems: the generation one, the transmission one and the distribution one, being each of these elements essential for the whole system to work correctly.

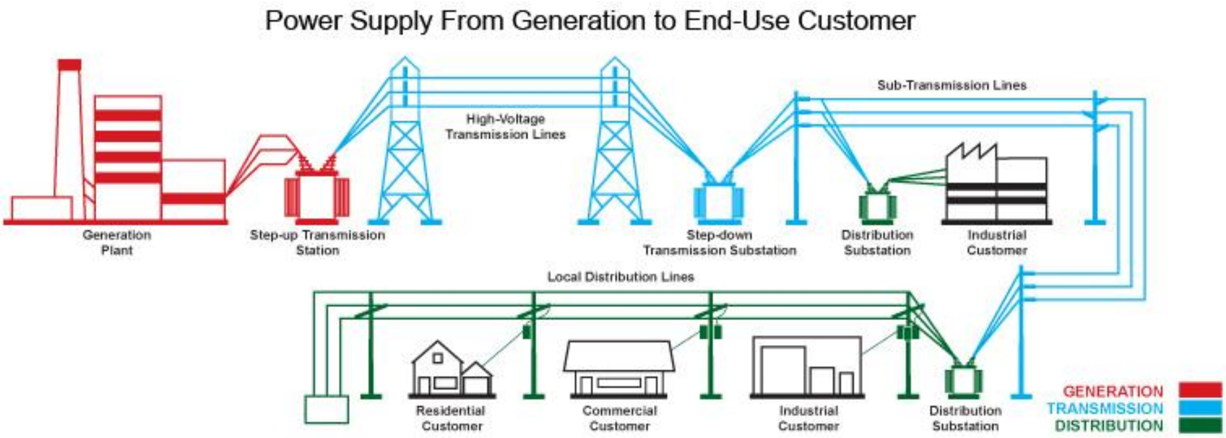


Figure1.1. Electricity generation, transmission and distribution system. [11]

Electric power is generated at the generation plant. There is wide range of different types of generation plants, depending on the energy source they use to transform mechanical power into electrical power. These generating stations are usually located far away from the places the energy will finally be used, so there is the need to transport energy large distances until it arrives to the final user.

Transformers are used to step-up voltage from generation plants to high-voltage transmission lines, which are the next element in the grid. Transmission lines are responsible for the electric power transportation. This transportation covers hundreds of kilometers, in which energy is at high voltage (220 kV or above) and three-phase AC.

When most of the distance has been covered, transmission substations will step-down voltages to values between 45 and 132 kV, and sub-transmission lines take the electricity to distribution stations, where voltages will be adapted to the final users demand voltages, which depend on the type of user, and by means of the distribution system, transported to them.

During all this process there are energy losses, which means that some of the energy which was generated at the generation plant is missed in its way to the user. In 2011, losses in the Spanish electrical system reached the 8,07% of the total energy that circulated by the grids. [7]

Some measures that can be taken in order to reduce losses are:

- Decreasing the distance of transmission and distribution lines
- Renovation and reconfiguration of old lines with low efficiency values
- Stepping-up voltages during transmission phases
- Increasing the power factor
- Increasing maintenance work in grids
- Progressing towards the deployment of Smart Grids
- Adding Distributed Generation

CHAPTER 2. LOSSES CALCULATION IN SMART GRIDS

The main objective of this chapter is to summarize the way losses are calculated in electrical grids, specifically when a distributed generator is implemented into the grid. In first place, a software which is used to calculate losses in electrical grids is presented. Then, the formulation behind losses calculations is explained. A Matlab code which has been developed in the UPC will be presented, and its validation will be carried out.

2.1. Software: OpenDSS

OpenDSS is used, among other things, to calculate losses in electrical grids.

It is an open source software which has been under development by the Electric Power Research Institute for the last 17 years. It was developed for special distribution analysis applications such as Distributed Generation analysis, providing a very flexible research platform. It can support nearly all frequency domain analysis commonly performed on electric power distribution systems. The reason why it is an open source is because it pretends to promote grid modernization and smart grid efforts by providing researchers and consultants with a tool to evaluate advanced concepts, with the aim of encouraging new advancements in distribution system analysis. [1]

Apart from allowing the calculation of a grid's losses, it has many other advanced applications such as:

- Neutral-to-earth (stray) voltage simulations
- Loss evaluations due to unbalanced loadings
- Development of DG models for the IEEE Radial Test Feeders
- High-frequency harmonic and interharmonic interference
- Losses, impedance, and circulating currents in unusual transformer back configurations
- Transformer frequency response analysis
- Distribution automation control algorithm assessment
- Impact on tankless water heaters on flicker and distribution transformers
- Wind farm collector simulation
- Wind farm impact on local transmission
- Wind generation and other DG impact on switched capacitors and voltage regulators
- Open-conductor fault conditions with a variety of single-phase and three-phase transformer connections

A very interesting feature of OpenDSS is that it allows the connection with other analysis software such as Matlab. In this thesis we will take advantadge of this feature and, by means of Matlab, we will be using results provided by OpenDSS to analyze them and try to reproduce them. We will mainly focus on the losses calculation of a certain grid we will be using as a test system. When a user is getting started with the use of the program, it is reccomended to start using it as an independent program.

When OpenDSS is being used as an independent program, electrical grids are defined using text-based commands. The user writes codes to define the properties of the grid, such as the basic line parameters, the transformers properties, loads, lines, etc. These can be written in different windows or all in the same one. Once a code is written it is activated by selecting it and right-clicking the option *Do Selected* from the pop-up menu.

This is the graphic interface the user works with:

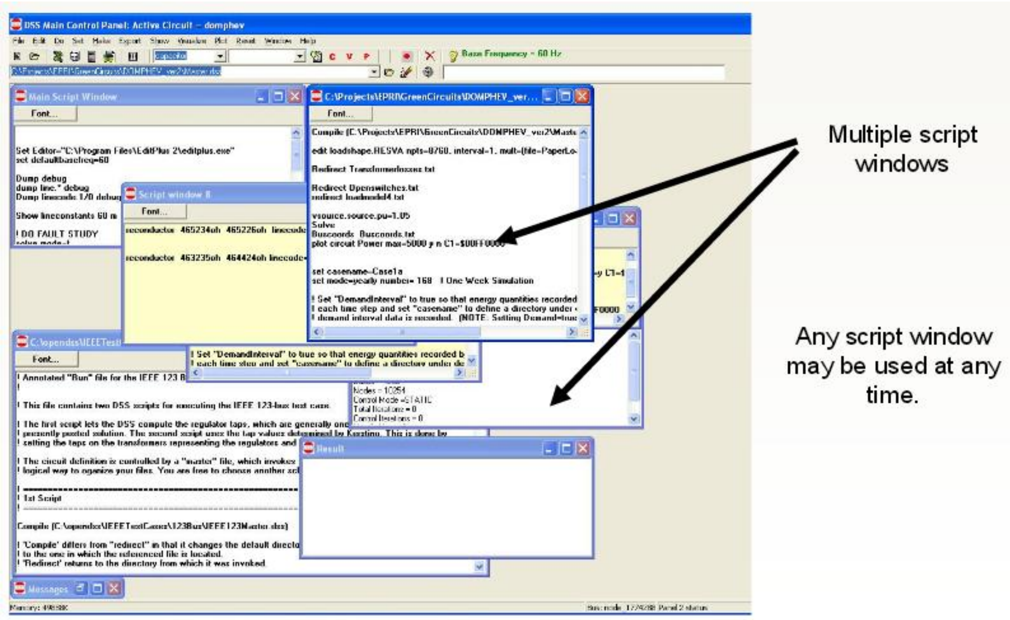


Figure 2.1. OpenDSS user interface

Codes are written with a particular syntax in order to define the grid correctly. In the next figure we can see a basic script that can be used in the program:

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

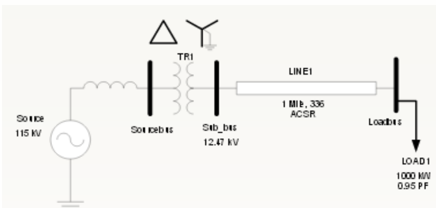


Figure 2.2. Basic code example

Writing the code *Show Losses*, losses in the defined grid are obtained.

However, it is much more comfortable to use OpenDSS through another software, such as Matlab. Matlab uses its *ActiveX* integrated server to communicate with OpenDSS' COM server, so this server turns into both programs' interface. Before doing that, the COM server must be registered in Windows. This registration is done by entering a code in DOS command window, as it is shown in the figure below:

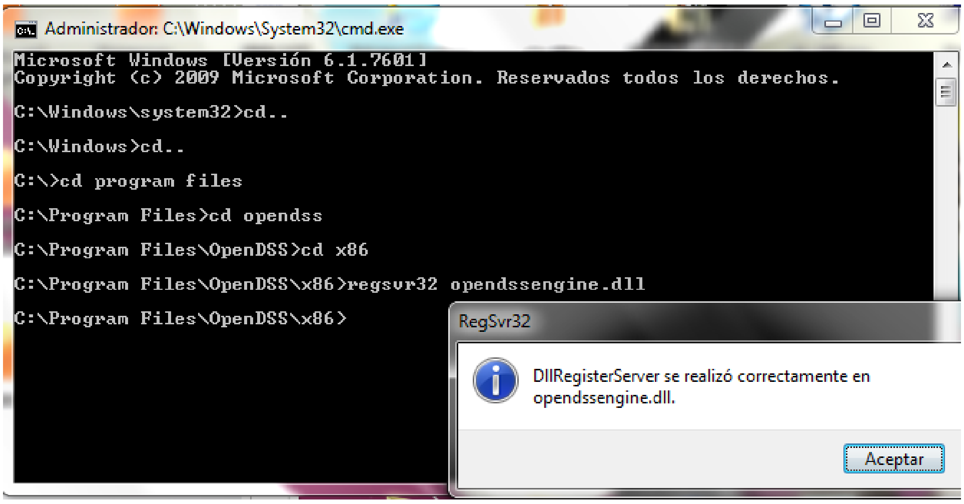


Figure 2.3. Registration of COM server in Windows

The last step is to connect both programs is the creation of a function in Matlab called *DSSStartup.m.*

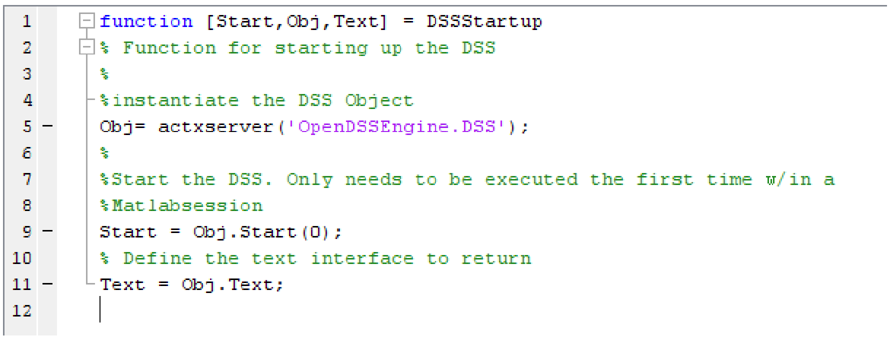


Figure 2.4. Function *DSSStartup.m* to start OpenDSS from Matlab

Once the DSSStratup function has been created and OpenDSS has been started, it can be used through Matlab with specific syntax to execute commands and define the grid. The benefits of using OpenDSS through Matlab is that all variable and results can be stored more easily and it is more comfortable regarding to further use of the data.

Another important feature about OpenDSS is that it can work in two different modes, depending on the power flow. These modes are the *Snapshot Mode* and the *Time Mode*:

Snapshot Mode

This is the default mode of OpenDSS. This mode takes a constant power demand value during an instantaneous lapse of time, and performs the simulation of the grid behavior at that precise moment. It works as if it was taking a photograph or "snapshot" of the grid at one concrete moment. This mode solves the problem of losses calculation only once, and its main advantage is its speed. However, it does not take into account variations of demand, and an only evaluation of the system is performed. This mode can be used to simulate the performance of the grid at peak demand episodes.

Time Mode

This mode takes into account that demand fluctuates with time, and is able to simulate the response of the grid to those variations. Time mode is set to calculate the behavior of the grid for a whole year and with a time lapse of 1 hour, even though these parameters can be changed.

The Time Mode pretends to be simulating with precision the behavior of the grid all along a period of time. This will happen if the demand values introduced in the program are realistic. Usually, demand is characterized by means of different load curves, each of which tries to represent a common behavior of demand along a whole year.

When working with Time Mode, not a single value of losses is obtained as solution. In this case, the solution is a losses vector, with so many components as time lapses considered.

The positive aspect about working with this mode is that results provide a lot of information about the grid. However, the computational cost it involves is quite large, so the time of calculation for complex grids is high. Indeed, if demand is not correctly characterized, results may not be loyal to reality at all.

It must be taken into account that results obtained using different modes cannot be compared because the conditions of the calculations and what they refer to are very different.

2.2. Definitions

The variables that play a part in the problem that will be considered are the following:

- Voltage $\bar{V}_i \in \mathbb{C}^{3x1}$: It is defined as the electrical potential difference between node i and the reference node or neutral node. In three-phase systems it has three complex components.

- Intensity $\bar{I}_i \in \mathbb{C}^{3x1}$: it is the flow of electric charge. As well as in the voltage, each intensity value has three components.

- Complex power $\bar{S}_i \in \mathbb{C}^{3x1}$: it is the rate of flow of energy past a given point of the circuit. It can be obtained as the product of the voltage by the conjugated intensity (when a complex vector is conjugated, the used notation to indicate it is adding an asterisk *).

$$S_i^k = V_i^k \cdot {I^*}_i^k\ ;\ \ k = 1,2,3;\ \bar{S}_i = [S_i^1, S_i^2, S_i^3]^T$$

  The complex power can also be obtained as the difference between the generated power and the demanded power.

- Local admittance matrix $[Y_i] \in \mathbb{C}^{3x3}$: it represents the nodal admittance of the nodes or buses (node or bus are words to express the same concept) in the system. Admittance is a measure of how easily the circuit will allow the current to flow. It is defined as the inverse of impedance. Due to Ohm's Law, the admittance matrix obeys the following relation:

$$[Y_i] \cdot \bar{V}_i = \bar{I}_i$$

  The local admittance matrices are used to define bigger matrices, such as line admittance matrices or global admittance matrices of the system. Later in this chapter, the way of forming these matrices will be explained.

  An admittance matrix is typically symmetric. However, there are particular grid configurations that lead to asymmetrical admittance matrices, such as the inclusion of regulation transformers.

2.3. Grid features

There are many different types of electrical grids. In this report, we are focusing on one specific kind with precise features, which are explained in the following lines. From now on, when electrical grids are mentioned, they all have these features:

- **Three-phase alternate system**

In a three-phase system, three circuit conductors carry three alternating currents (of the same frequency) which reach their instantaneous peak values at one third of a cycle from each other. Taking one current as the reference, the other two currents are delayed in time by one third and two thirds of one cycle of the electric current. It is the most common method used by electrical grids worldwide to transfer power. A three-phase system is usually more economical than an equivalent single-phase or two-phase system at the same voltage because it uses less conductor material to transmit electrical power.

- **Single-feeder distribution system**

In single feeder distribution systems there is only one feeder coming from the Transmission Substation. If, for any reason, the feeder stays out of service (failure, maintenance work, etc.) the entire grid would stay without service. In the type of grid we are considering in this thesis, only one feeder is considered, and there are no ramifications in the grid. After the feeder or generator, a transformer will be connected to the grid.

2.4. Equations

The considered electrical grids consist on a generator (0), a transformer (T), and *n* nodes or buses. They can be framed in the following diagram:

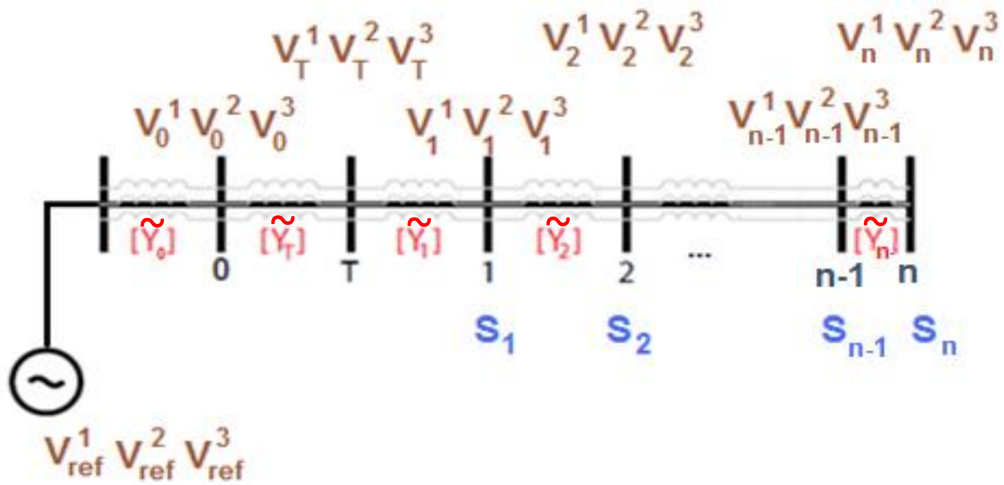


Figure 2.5. Diagram of a grid

To model the grid and perform analysis, a system of equations can be defined. There is some information that is known and some that needs to be calculated.

DATA

- Reference voltage: it is the imposed voltage at the grid start.

$$\bar{V}_{ref} = \left[V_{ref}^1, V_{ref}^2, V_{ref}^3\right]^T \in \mathbb{C}^{3x1}$$

- Line admittance matrices $\left[\widetilde{Y}_\iota\right] \in \mathbb{C}^{6x6}$: they are formed from the local admittance matrices. There are so many line admittance matrices as lines in the grid. They depend exclusively on the physical properties of the line. In general, local admittance matrices in a same grid will be proportional to each line's length. Local admittance matrices associated to the generator and the transformer are particular admittance matrices, and do not show proportionality to the rest of matrices.

  The transformer line admittance matrix $\left[\widetilde{Y_T}\right]$ is unique for each type of transformer, and depends on its physical features. It has this structure:

$$\left[\widetilde{Y_T}\right] = \begin{pmatrix} Y_T^1 & Y_T^2 \\ -Y_T^2 & Y_T^3 \end{pmatrix};\ Y_T^k \in \mathbb{C}^{3x3}; k = 1,2,3$$

  The structure of the generator admittance matrix and the rest of local admittance matrices is the following:

$$\left[\widetilde{Y_\iota}\right] = \begin{pmatrix} Y_i & -Y_i \\ -Y_i & Y_i \end{pmatrix};\ Y_i \in \mathbb{C}^{3x3}; i = 0,1,2,\dots,n$$

- Nodal complex power $\bar{S}$: The supplied power for each of the nodes is an imposed value in the problem. The generator and the transformer are not demanding power, so their associated nodal complex power is 0.

$$\bar{S} = [0,0,0,0,0,0,S_1^1,S_1^2,S_1^3,S_2^1,S_2^2,S_2^3,\dots,S_n^1,S_n^2,S_n^3]^T \in \mathbb{C}^{3(n+2)x1}$$

UNKNOWNS

- Buses voltages $\bar{V}_i \in \mathbb{C}^{3x1};\ \ i = 0, T, 1,2,3,\dots,n$
  They can be gathered in a vector:

$$\bar{V} = [V_0^1,V_0^2,V_0^3,V_T^1,V_T^2,V_T^3,V_1^1,V_1^2,V_1^3,\dots,V_n^1,V_n^2,V_n^3]^T \in \mathbb{C}^{3(n+2)x1}$$

EQUATIONS

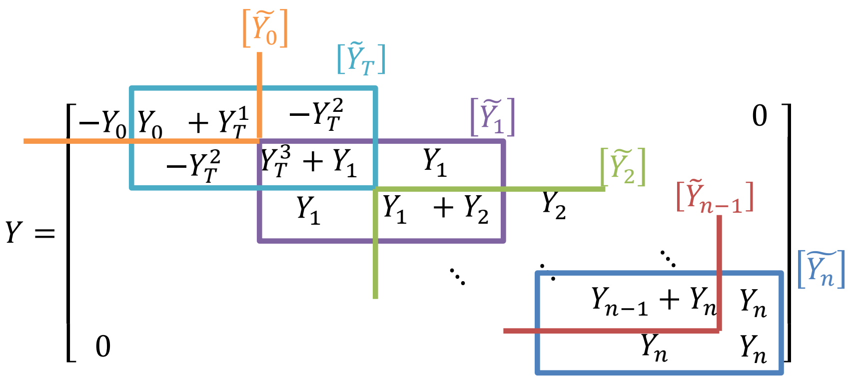
Once the type of grid we are working with has been defined, as well as the input data and the unknowns of the problem, in this section it is going to be explained how to process the calculation of losses in load-flow problems.

The obtaining of buses voltages is ruled by the following governing equation:

$$\overline{S^*} = diag(V^*) \cdot Y \cdot \bar{V}$$

where Y is the global admittance matrix of the system and $\bar{S}$ is the power vector, considering the distributed generation power.

The global admittance matrix is formed by the line admittance matrices the way shown below:



3(n+2)x3(n+3)

The previous governing equation can be transformed to an equivalent one, which will be easier to work with:

$$\overline{S^*} = diag(V^*) \cdot [Y_0 \cdot \bar{V}_{ref} + Y_{red} \cdot \bar{V}]$$

where

Y_0 is a matrix formed by the first 3 columns of Y

Y_{red} is a matrix formed by all the columns of Y except the first 3.

This equation can be shown with the matrices representation:

$$\begin{bmatrix} {V_0^1}^* & & & & & & 0 \\ & {V_0^2}^* & & & & & \\ & & {V_0^3}^* & & & & \\ & & & {V_T^1}^* & & & \\ & & & & \ddots & & \\ & & & & & {V_n^2}^* & \\ 0 & & & & & & {V_n^3}^* \end{bmatrix} \cdot$$

3(n+2)x3(n+2)

$$\cdot\left(\begin{bmatrix} -Y_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} V_{ref}^1 \\ V_{ref}^2 \\ V_{ref}^3 \end{bmatrix} + \begin{bmatrix} Y_0 + Y_T^1 & -Y_T^2 & & & & 0 \\ -Y_T^2 & Y_T^3 + Y_1 & Y_1 & & & \\ & Y_1 & Y_1 + Y_2 & Y_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & Y_{n-1} + Y_n & Y_n \\ 0 & & & & Y_n & Y_n \end{bmatrix} \cdot \begin{bmatrix} V_0^1 \\ V_0^2 \\ V_0^3 \\ V_T^1 \\ \vdots \\ V_n^2 \\ V_n^3 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ S_1^1 \\ S_1^2 \\ S_1^3 \\ S_2^1 \\ \vdots \\ S_n^3 \end{bmatrix}^*$$

6 zeros

3(n+2)x3 3x1 3(n+2)x3(n+2) 3(n+2)x1 3(n+2)x1

The representation above has the structure of a non-linear system of equations, where the unknowns are buses voltages (V_0^1 ... V_n^3).

In order to ease the way the system is solved, matrixes are decomposed in real and imaginary part. As a consequence of that, the size of matrices is multiplied by two, but no complex numbers are inside them.

There is a way of transforming complex matrices into real matrices without affecting the result. It consists of converting each complex value in a *2x2* block, putting the real part at the diagonal and the imaginary part in the other two positions. The six matrices involved in the system will change, having this final structure:

$$diag(V) = \begin{bmatrix} \Re(V_0^1) & \Im(V_0^1) & & & & & 0 \\ -\Im(V_0^1) & \Re(V_0^1) & & & & & \\ & & \Re(V_0^2) & \Im(V_0^2) & & & \\ & & -\Im(V_0^2) & \Re(V_0^2) & & & \\ & & & & \ddots & & \\ & & & & & \Re(V_0^n) & \Im(V_0^n) \\ 0 & & & & & -\Im(V_0^n) & \Re(V_0^n) \end{bmatrix}$$

6(n+2)x6(n+2)

$$Y_0 = \begin{bmatrix} \Re(-Y_0) & \Im(Y_0) \\ -\Im(Y_0) & \Re(-Y_0) \\ 0 & 0 \\ \vdots & \vdots \\ & \\ 0 & 0 \end{bmatrix} \qquad \bar{V}_{ref} = \begin{bmatrix} \Re(V_{ref}^1) \\ \Im(V_{ref}^1) \\ \Re(V_{ref}^2) \\ \Im(V_{ref}^2) \\ \Re(V_{ref}^3) \\ \Im(V_{ref}^3) \end{bmatrix}$$

6(n+2)x6 6x1

$$Y_{red} = \begin{bmatrix} \Re(Y_0 + Y_T^1) & -\Im(Y_0 + Y_T^1) & \Re(Y_T^2) & -\Im(Y_T^2) & & 0 \\ \Im(Y_0 + Y_T^1) & \Re(Y_0 + Y_T^1) & \Im(Y_T^2) & \Re(Y_T^2) & & \\ \Re(Y_T^2) & -\Im(Y_T^2) & \Re(Y_T^3 + Y_1^1) & -\Im(Y_T^3 + Y_1^1) & & \\ \Im(Y_T^2) & \Re(Y_T^2) & \Im(Y_T^3 + Y_1^1) & \Re(Y_T^3 + Y_1^1) & \ddots & \\ & & & & & \Re(Y_n^1) \; -\Im(Y_n^1) \\ & & & & & \Im(Y_n^1) \; \Re(Y_n^1) \\ 0 & & & & & \end{bmatrix}$$

6(n+2)x6(n+2)

$$\bar{V} = \begin{bmatrix} \Re(V_0^1) \\ \Im(V_0^1) \\ \Re(V_0^2) \\ \Im(V_0^2) \\ \vdots \\ \Re(V_n^3) \\ \Im(V_n^3) \end{bmatrix} \qquad \bar{S} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Re(S_1^1) \\ \Im(S_1^1) \\ \Re(S_1^2) \\ \Im(S_1^2) \\ \vdots \\ \Re(S_n^3) \\ \Im(S_n^3) \end{bmatrix}$$

6(n+2)x1 6(n+2)x1

This way, the presented non-linear system of equations has been transformed into an equivalent one that does not include complex numbers.

2.5. Newton Raphson Method

Newton Raphson method is used to solve the presented system of equations. It is an efficient way to find zeros in a real function *F*. It is an iterative method that starts working with a given solution $\bar{V_0}$ that is improved at each step until the solution is found.

The function whose zeros are about to be found with newton Raphson method is:

$$F(\bar{V}) = diag(V^*) \cdot [Y_0 \cdot \bar{V}_{ref} + Y_{red} \cdot \bar{V}] - S^* = 0$$

Newton Raphson method's way to get each new solution is by means of the next formulation:

$$F(\overline{V_i} + \bar{h}) = F(\overline{V_i}) + J(F(\overline{V_i})) \cdot \bar{h}$$

where

- $\bar{h}$ is the step at each iteration
- $F(\bar{x}) = \begin{Bmatrix} F_1(V_1, V_2, \dots V_n) = F_1(\bar{V}) = 0 \\ \vdots \\ F_m(V_1, V_2, \dots V_n) = F_m(\bar{V}) = 0 \end{Bmatrix}$
- $J(F(\bar{V})) = \begin{bmatrix} \frac{\partial F_1(\bar{V})}{\partial V_1} & \cdots & \frac{\partial F_1(\bar{V})}{\partial V_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n(\bar{V})}{\partial V_1} & \cdots & \frac{\partial F_n(\bar{V})}{\partial V_n} \end{bmatrix}$ is the Jacobian of the function $F(\bar{V})$

When a Newton Raphson method is computed, stopping criterion must be defined. Otherwise, the algorithm would continue iterating even though the correct solution was already achieved.

In the computation of this problem, 3 stopping criterion were used:

- L^2 norm of the remainder / L^2 norm of the term independent < $1 \cdot 10^{-10}$
- Infinity norm of the remainder / Infinity norm of the term independent < $1 \cdot 10^{-10}$
- L^2 norm of the difference between the solution and the one obtained at the previous iteration / L^2 norm of the solution < $1 \cdot 10^{-6}$

2.6. Losses calculation

Once the Newton Raphson method is applied, a vector with bus voltages is obtained. At this point, all the information to calculate the losses at the grid is available. With the voltages values, the governing equation can be used to obtain the loads at each of the nodes. The components associated to the nodes of the new computed vector *S* will be used to obtain the nodes' loads. To obtain losses, only the real part of the values is taken.

The system total losses are obtained as the power injected into the grid minus the power of the loads and the distributed generator.

When the calculation of losses is computed, the generator is evaluated separately from the rest of the nodes, with the following formulation:

$$Losses = \Re(S_{gen} + S_{nodes})$$

where

$$S_{gen} = \sum_{i=1}^{3} {I_0}_i^* \cdot V_0^i \, ; \qquad S_{nodes} = \sum_{j=7}^{3(n+2)} S_j$$

$$I_0 = \underset{3x3(n+2)}{\begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,306} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,306} \\ Y_{3,1} & Y_{3,2} & \cdots & Y_{3,306} \end{bmatrix}} \cdot \underset{3(n+2)x1}{\begin{bmatrix} V_0^1 \\ V_0^2 \\ V_0^3 \\ V_T^1 \\ \vdots \\ V_n^2 \\ V_n^3 \end{bmatrix}} - \underset{3x3}{Y_0^1} \cdot \underset{3x1}{\begin{bmatrix} V_0^1 \\ V_0^2 \\ V_0^3 \end{bmatrix}} \in \mathbb{R}_{3x1}$$

2.7. Test system

The explained formulation to obtain losses has been applied into a concrete grid, which will be referred as test system. The grid is composed of 100 nodes, and a representation of it can be seen in Figure 2.6:

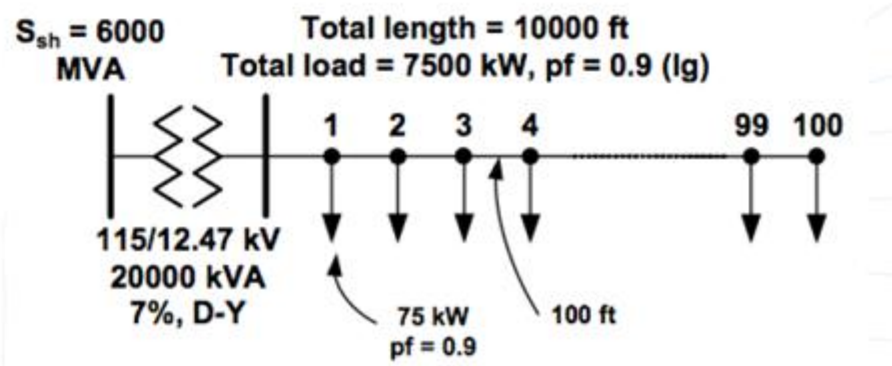


Figure 2.6. Test system

Indeed, a photovoltaic source will be added to the grid. This Distributed Generation will not be located in a fix position, neither with a fix power. Its position and power will be varied, leading to different grid configurations.

Adding the Distributed Generation will affect the vector *S*, so for any position and power of the photovoltaic source there will be a different system of equations to be solved.

The DG source can be located in any of the nodes of the grid (from 1 to 100) and with a power range up to 7.500 kW.

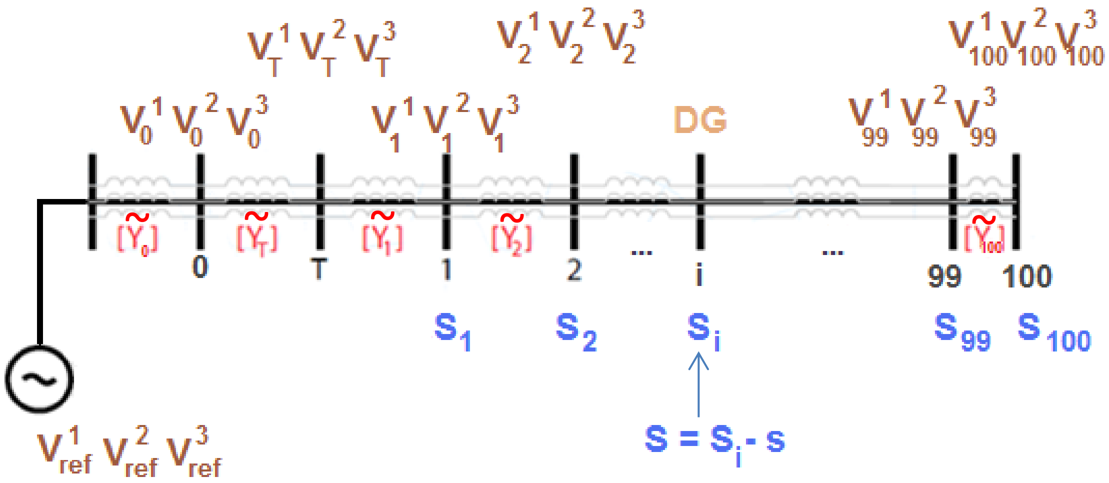


Figure 2.7. Test system diagram when a DG source is added

The way of defining the vector *S* will vary depending on the working mode (Snapshot Mode or Time Mode).

In Snapshot Mode the problem is solved only once, with a constant value of power for all nodes (except the one including the DG source). On the other hand, Time Mode implies solving the problem more times (8760 when performing a year analysis). In time mode, the power fluctuates at each hour.

The definition of the load in each mode is calculated as follows:

- Snapshot mode

$$Q = \frac{load_{kw}}{load_{pf}} \cdot \sqrt{1 - load_{pf}^2} = \frac{75}{0{,}9} \cdot \sqrt{1 - 0.9^2}$$

$$S_i^k = \left(\frac{load_{kw}}{3} + j \cdot \frac{Q}{3}\right) \in \mathbb{C};\; k = 1{,}2{,}3\; i = 1, \dots, 100$$

$load_{kw}$ and $load_{pf}$ are known values inherent to the grid.

Adding the distributed generation is reflected on three values of the vector. The three components corresponding to the DG position will see its power value modified the following way:

$$\begin{bmatrix} \widehat{S_{DG}^1} \\ \widehat{S_{DG}^2} \\ \widehat{S_{DG}^3} \end{bmatrix} = \begin{bmatrix} S_{DG}^1 - \frac{P_{DG}}{3} \\ S_{DG}^2 - \frac{P_{DG}}{3} \\ S_{DG}^3 - \frac{P_{DG}}{3} \end{bmatrix}$$

where DG is the position of the photovoltaic source $(DG \in \{1{,}100\})$ and P_{DG} is its power value $(P_{DG} \in \{0{,}7500\})$

- Time Mode

$$Q = \frac{load_{kw}}{load_{pf}} \cdot \sqrt{1 - load_{pf}^2} = \frac{75}{0{,}9} \cdot \sqrt{1 - 0.9^2}$$

$$S_i^k = \left(\frac{load_{kw} \cdot Load}{3 \cdot \max(Load)} + j \cdot \frac{load_{kw} \cdot Q}{3 \cdot \max(Load)}\right) \in \mathbb{C};\ k = 1{,}2{,}3\ i = 1, \dots{,}100$$

where the value $Load$ is determined by demand load curves. There are 4 different load curves, and each node of the grid is associated to one of them. These curves provide one load value for each hour of the year.

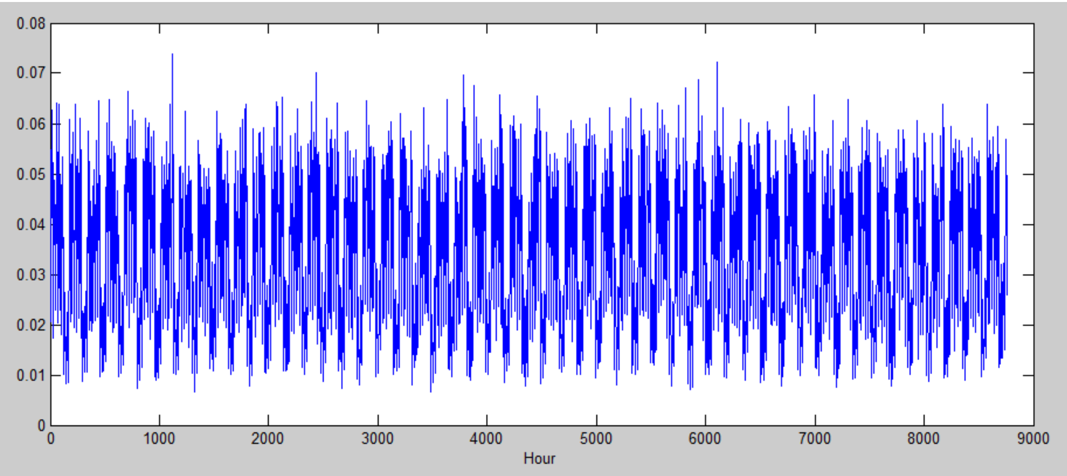


Figure 2.8. Example of a load curve

To reflect the effect of the distributed generation, its associated three components power vector is modified this way:

$$\begin{bmatrix} \widehat{S_{DG}^1} \\ \widehat{S_{DG}^2} \\ \widehat{S_{DG}^3} \end{bmatrix} = \begin{bmatrix} S_{DG}^1 - \frac{DSun \cdot P_{DG}}{3 \cdot \max(DSun)} \\ S_{DG}^2 - \frac{DSun \cdot P_{DG}}{3 \cdot \max(DSun)} \\ S_{DG}^3 - \frac{DSun \cdot P_{DG}}{3 \cdot \max(DSun)} \end{bmatrix}$$

where the value *DSun* is depending on the hour of the year. It is given by a load curve associated to the photovoltaic generator that reflects the solar intensity along a whole year.

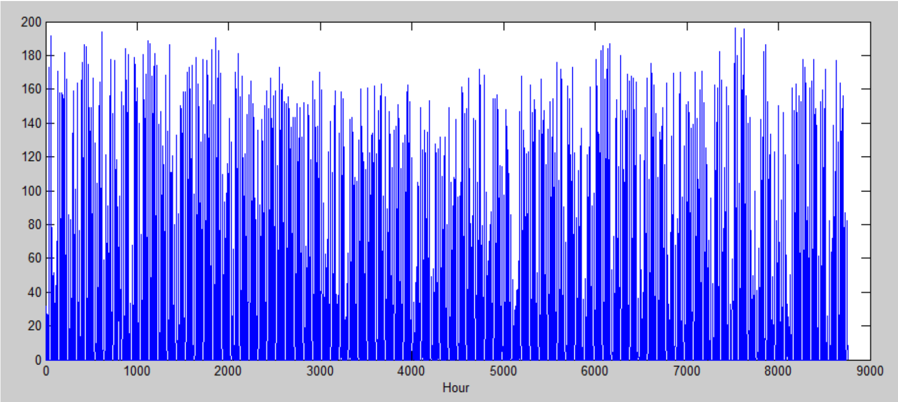


Figure 2.11. Load curve associated to the photovoltaic generator

2.8. Validation of the code

Using the previously presented formulation, a Matlab code which allows the calculation of losses in an electrical grid has been developed at UPC.

To check its reliability, a comparison between the results this model provides and the ones OpenDSS has been made. 10 losses solutions to different configurations of the grid have been analyzed. The 10 DG power values (also known as *PP)* and DG position (also known as *BP)* were randomly selected, and the solutions were obtained by the OpenDSS software. Then, the solutions for the same configurations were obtained with the Matlab code in order to validate it.

At Figure 2.12, we can see the values of the node location and power which were used to obtain the 10 solutions, as well as its representation on a mesh:

	BP	PP
1	30	3332.7
2	70	2908.1
3	68	5759.4
4	19	5979.4
5	14	1462.5
6	52	3711.5
7	98	3383.5
8	37	4873.9
9	61	5342.0
10	25	5678.5

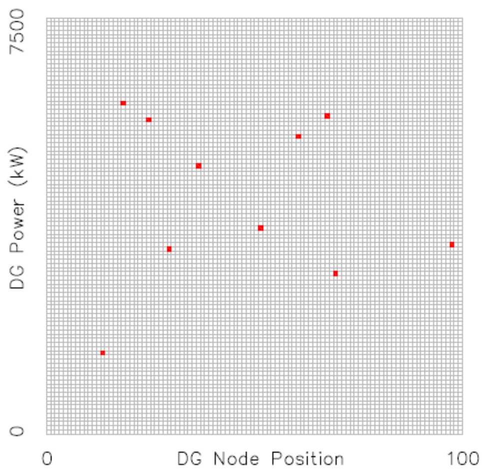


Figure 2.10. Random values of DG location and power

The results obtained using the two different ways are shown at Table 2.1. Absolute and relative errors, taking the OpenDSS value as the reference one, have been calculated.

BP	PP (kW)	Matlab Losses (kW)	OpenDSS Losses (kW)	Absolute error (kW)	Relative error
30	3332,7	154431,50	154430,10	1,272	$8{,}239{\cdot}10^{-06}$
70	2908,1	141315,75	141314,53	0,989	$6{,}999{\cdot}10^{-06}$
68	5759,4	137738,88	137737,69	1,385	$1{,}005{\cdot}10^{-05}$
19	5979,4	156840,23	156838,68	1,532	$9{,}770{\cdot}10^{-06}$
14	1462,5	175840,07	175838,58	1,657	$9{,}422{\cdot}10^{-06}$
52	3711,5	140417,61	140416,36	1,254	$8{,}931{\cdot}10^{-06}$
98	3383,5	141047,03	141045,84	1,156	$8{,}194{\cdot}10^{-06}$
37	4873,9	143614,69	143613,40	1,236	$8{,}607{\cdot}10^{-06}$
61	5342,0	135589,30	135588,10	1,129	$8{,}329{\cdot}10^{-06}$
25	5678,5	151406,06	151404,61	1,482	$9{,}791{\cdot}10^{-06}$

Table 2.1. Results using OpenDSS and the Matlab code

The difference for a year analysis is about 1 kW, and the relative error is about $1{\cdot}10^{-5}$, which is considered a small difference. It can be concluded that the Matlab code provides good results and is correctly designed to obtain losses in the test system grid.

The graphical output results that are obtained using this code for a complete analysis of the grid are the following:

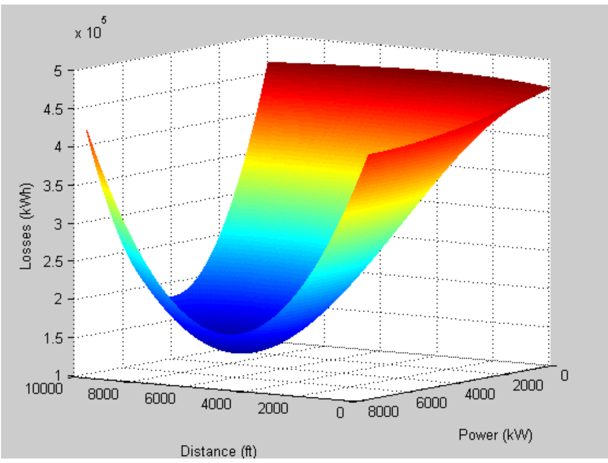


Figure 2.11. Distribution of losses depending on DG position and power

CHAPTER 3. MODEL REDUCTION TECHNIQUES

In the previous chapter, losses in a test system grid were calculated by means of OpenDSS and Matlab, obtaining really similar results. However, even the example grid had 100 nodes (a real physical grid is much larger), each year analysis takes about one hour time to be obtained. The model seems to provide good results but its computational cost is a hindrance for its effectiveness.

This fact leads to the conclusion that some techniques to achieve a model reduction have to be applied. An objective of this work is to provide information to obtain a reduced basis that is able to describe correctly the problem. A way to start finding the elements of this reduced basis and to start reducing the dimensionality of the problem is analyzing the Principal Components of the data in the problem (input data or solution data).

The problem we are taking in consideration is a load flow problem in which the position and power of a distributed generator can be varied. It can be located in 100 hundred different positions and with a range of power between 0 and 7500 kW. To characterize all the possibilities, a huge amount of data is needed. (if power was taken with entire numbers, more than 6 million solutions (7.500 x 100 x 8760 = 6.570.000.000) would be saved during a complete analysis of the grid) Being able to reproduce effectively the behavior of the net is important. Some techniques to achieve that, whether along this report or in further research, are explained in this chapter.

3.1. Principal Component Analysis

Principal components analysis is a statistical technique that linearly transforms an original set of variables into a substantially smaller set of uncorrelated variables that represents most of the information in the original set of variables. Its goal is to reduce the dimensionality of the original data set. A small set of uncorrelated variables is much easier to understand and use in further analyses than a larger set of correlated variables. This technique was conceived by Pearson in 1901 and independently developed by Hotelling in 1933.

Principal Components Analysis is useful in several ways. It allows the identification and elimination of multicolinearities in the data, by reducing the dimension of the input space leading to fewer parameters. The variance regression coefficient estimator is minimized by the PCA choice of basis.

In this report we will use the PCA method in order to try to reduce the dimensionality of the problem. We will apply it to the voltage vectors we obtained as result of applying the Newton Raphson method.

In first place, before applying it, it will be explained how the PCA method works, with general notation. [2]

Notation

- x is a vector of p random variables
- α_k is a vector of p constants
- $\alpha'_k \; x = \sum_{j=1}^{p} \alpha_{kj} \, x_j$

Procedural description

- Find a linear function of x, $\alpha'_1 \; x$ with maximum variance
- Then find another linear function of x, α'_2 , uncorrelated with $\alpha'_1 \; x$ maximum variance
- Iterate

Most of the variation in x will be accounted for by m Principal Components, where it is expected that $m \ll p$

Assumption and more notation

- $\sum$ is the known covariance matrix for the random variable x
- Foreshadowing: $\sum$ will be replaced with S, the sample covariance matrix, when $\sum$ is unknown.

First step

- Find $\alpha'_k \, x$ that maximizes $Var(\alpha'_k \, x) = \alpha'_k \sum \alpha_k$
- Choose normalization constraint, namely $\alpha'_k \, \alpha_k = 1$ (unit length vector)

Constrained maximization – method of Lagrange multipliers

To maximize $\alpha'_k \sum \alpha_k$ subject on $\alpha'_k \, \alpha_k = 1$,the technique of Lagrange multipliers is used. We maximize the function

$$\alpha'_k \sum \alpha_k - \lambda(\alpha'_k \, \alpha_k - 1)$$

This results in

$$\begin{aligned}\frac{d}{d\alpha_k}\left(\alpha_k'\mathbf{\Sigma}\alpha_k-\lambda_k(\alpha_k'\alpha_k-1)\right) &= 0\\ \mathbf{\Sigma}\alpha_k-\lambda_k\alpha_k &= 0\\ \mathbf{\Sigma}\alpha_k &= \lambda_k\alpha_k\end{aligned}$$

This is recognizable as an eigenvector equation where α_k is an eigenvector of $\sum$ and λ_k is the associated eigenvalue.

The quantity to be maximized is:

$$\alpha_k'\mathbf{\Sigma}\alpha_k=\alpha_k'\lambda_k\alpha_k=\lambda_k\alpha_k'\alpha_k=\lambda_k$$

then we should choose λ_k to be as big as possible. So, calling $\lambda_1$the largest eigenvector of $\sum$ and α_1 the corresponding eigenvector then the solution to

$$\sum\alpha_1 = \lambda_1\alpha_1$$

Is the 1st Principal Component of x.

In general will be the kth PC of x and $Var(\alpha'\, x)=\lambda_k$

The second principal component, $\alpha_2\; x$ maximizes $\alpha_2\sum\alpha_2$ subject to being uncorrelated with $\alpha_1\; x$

The uncorrelation constraint can be expressed using any of these equations:

$$\begin{aligned}\mathrm{cov}(\alpha_1'\mathbf{x},\alpha_2'\mathbf{x}) &= \alpha_1'\mathbf{\Sigma}\alpha_2=\alpha_2'\mathbf{\Sigma}\alpha_1=\alpha_2'\lambda_1\alpha_1'\\ &= \lambda_1\alpha_2'\alpha=\lambda_1\alpha_1'\alpha_2=0\end{aligned}$$

Of these, if we choose the last one, we can write a Langrangian to maximize α_2

$$\alpha_2'\mathbf{\Sigma}\alpha_2-\lambda_2(\alpha_2'\alpha_2-1)-\phi\alpha_2'\alpha_1$$

$$\begin{aligned}\frac{d}{d\alpha_2}\left(\alpha_2'\mathbf{\Sigma}\alpha_2-\lambda_2(\alpha_2'\alpha_2-1)-\phi\alpha_2'\alpha_1\right)=0\\ \mathbf{\Sigma}\alpha_2-\lambda_2\alpha_2-\phi\alpha_1=0\end{aligned}$$

If we multiply α_1 into this expression

$$\begin{aligned}\alpha_1'\mathbf{\Sigma}\alpha_2-\lambda_2\alpha_1'\alpha_2-\phi\alpha_1'\alpha_1=0\\ 0-0-\phi 1=0\end{aligned}$$

Then we can see that ϕ must be zero and this is true when

$$\boldsymbol{\Sigma}\alpha_2 - \lambda_2\alpha_2 = 0$$

This is another eigenvalue equation. The process can be repeated for $k = 1 \dots p$ yielding up to p different eigenvectors of $\sum$ along with the corresponding eigenvalues $\lambda_1 \dots \lambda_p$.

Furthermore, the variance of each of the PCs are given by

$$\text{Var}[\alpha_k' \mathbf{x}] = \lambda_k, \qquad k = 1, 2, \dots, p$$

Properties of PCA

For any integer $q, 1 \leq q \leq p$, considering the orthonormal linear transformation

$$\mathbf{y} = \mathbf{B}'\mathbf{x}$$

Where y is a q-element vector and B' is a q x p matrix, and $\sum_y = B'\sum B$ is the variance-covariance matrix for y. Then the trace of $\sum_y$ is maximized by taking $B = A_q$, where A_q consists on the first q columns of A.

This means that if a lower dimensional projection of x is chosen, then the retained variance of the resulting variables is maximized.

In fact, since the projections are uncorrelated, the percentage of variance accounted by retaining the first q Principal Components is given by:

$$\frac{\sum_{k=1}^{q} \lambda_k}{\sum_{k=1}^{p} \lambda_k} \times 100$$

PCA using the sample covariance matrix

The sample covariance matrix (an unbiased estimator for the covariance matrix of x is given by

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}'\mathbf{X}$$

Where $X_{ij} = x_{ij} - \bar{x}_j$ is a zero mean design matrix.

We construct the matrix A by combining the p eigenvectors of S (or eigenvectors of $X'X$, since they are the same)

The most straightforward way of computing the PCA loading matrix is using the singular value decomposition of $S = A'\Lambda A$, where A is a matrix consisting on the eigenvectors of S and Λ is a diagonal matrix whose elements are the eigenvalues corresponding to each of the considered eigenvectors.

Creating a reduced dimensionality projection of X is accomplished by selecting the q largest eigenvalues in Λ and retaining the q corresponding eigenvectors from A.

Simple example of dimensionality reduction using PCA

The figures below show two different views of a plotting of the same data

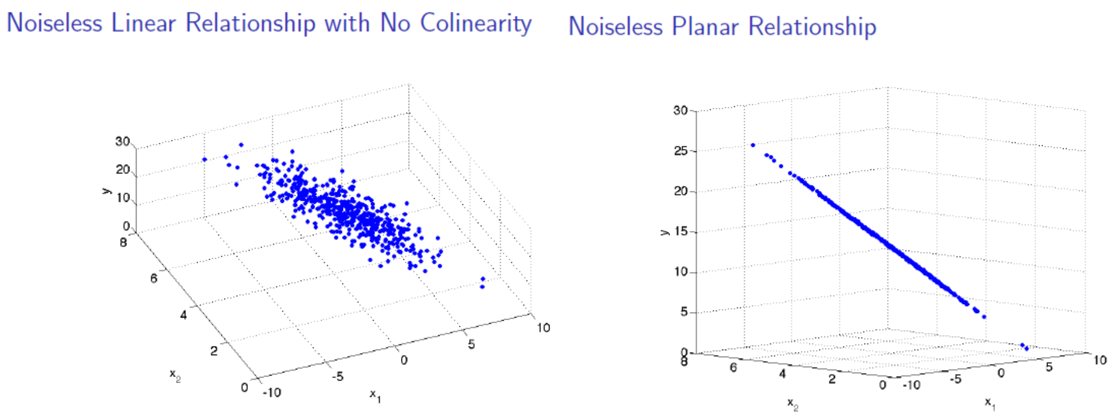


Figure 3.1. Example of 3-dimensional data representation

In this simple example, the initial data was 3-dimensional. However, by applying a rotation, it was obvious that all the information was located in a plane. If we calculate the Principal Components, the third eigenvalue will be zero, what means that the data can be reduced to two dimensions. All the variance of the data is retained in two dimensions.

After applying the method, the data can be plotted in the following way:

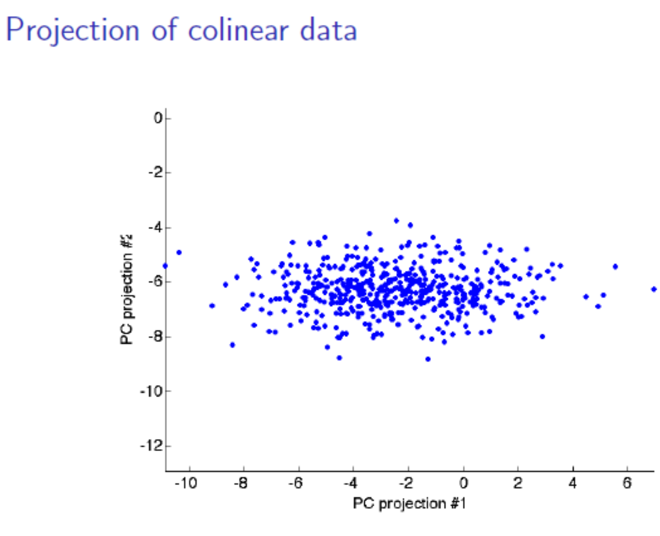


Figure 3.2. Projection of data in Principal Components axis

This is a very simple example, but summarizes the main idea of dimensionality reduction in a very visual way.

3.2. Reduced Basis strategy

Applying the Principal Components Analysis is a way of reducing the dimensionality of a problem. This can be made with different purposes. One of them is to finally obtain a reduced basis.

In the kind of problem that is studied here, a reduced basis would allow the characterization of the grid by solving the system at some precise points. That points would be the elements of the reduced basis.

In the considered problem, one of the objectives of this report is to provide a way of obtaining a good approximation of the total year losses by calculating them only in some points along the year, reducing the amount of calculations performed. The set of representative hours in the year would be a reduced basis that would allow performing a whole year analysis providing of an approximated result of annual total losses.

CHAPTER 4. NUMERICAL EXAMPLES

The main objective of this thesis is finding ways to reduce the computational cost when real-time simulations of a specific load-flow problem are performed. This can be made by reducing the amount of input data or by trying to reduce the number of calculations made by the solver. This section is focused on Time Mode, which allows load-flow simulations for long periods of time.

A technique that will be used in order to achieve that objective is the Principal Component Analysis, which was explained in the previous chapter.

In first place, it will be discussed whether it is possible or not to reduce the amount of information that is introduced into the program to perform a whole year simulation

4.1. Reduction of the input data

When an analysis is performed, some variables are introduced into the program (OpenDSS or Matlab).

The input data consists of the admittances matrices, the load curves that represent the electrical demand along the year (*Dload curves*), and the load curve that represents the solar power along the year (*Dsun curve*).

As it was mentioned in *Chapter 2*, there are four different load curves $Dload_i$ $i=1,\dots,4$. Having different curves is a way of pretending to model realistically the behavior of the power demand on the grid. Each of the buses of the grid is ruled by one of these curves.

Load curve	Nodes ruled by this curve
Dload1	1,2,4,6,16,22,25,28,33,40,42,43,45,52,57,64,66,71,77,73,80,82,89,92,97
Dload2	7,9,12,13,15,20,24,30,32,35,39,41,47,49,50,51,55,62,65,72,74,78,81,85, 90,91,96,99
Dload3	3,10,11,18,19,21,27,29,36,37,46,53,56,61,63,68,69,75,79,87,88,95,98
Dload4	5,8,14,17,23,26,31,34,38,44,48,54,58,59,60,67,70,76,83,84,86,93,94,100

Table 4.1. Load curve associated to each node

Each curve is made up by 8760 components, since it is the amount of hours in a year (1 year = 365 days · 24 hours/day = 8760 hours).

4.1.1. Analysis of the demand load curves

A way of reducing the amount of input data would be being able to express the four load curves as a function of one of them.

The first step is creating a matrix 8760 x 4, in which every column is one of this curves, in order to see whether the rank of the matrix is 4 or less than 4.

$$Dload = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \overline{Dload1} & \overline{Dload2} & \overline{Dload3} & \overline{Dload4} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \in \mathbb{R}_{8760X4}$$

where:

$$\overline{Dload1} = [Dload1_1, Dload1_2, \dots, Dload1_{8760}\ ]^T$$
$$\overline{Dload2} = [Dload2_1, Dload2_2, \dots, Dload2_{8760}\ ]^T$$
$$\overline{Dload3} = [Dload3_1, Dload3_2, \dots, Dload3_{8760}\ ]^T$$
$$\overline{Dload4} = [Dload4_1, Dload4_2, \dots, Dload4_{8760}\ ]^T$$

The rank of this matrix is 4, which means that all the curves are linearly independent. It is not possible to express a whole curve as a linear function of the others.

The next step in order to try to reduce the input data is plotting the curves and trying to establish relations between them. Plotting the curves for a long period of time does not allow the comparison due to scale reasons, so the analysis is made in short periods of time, such as periods of 24 hours.

In the chart below we can see the four load curves along one whole day. In particular, they are the values corresponding to 1 January:

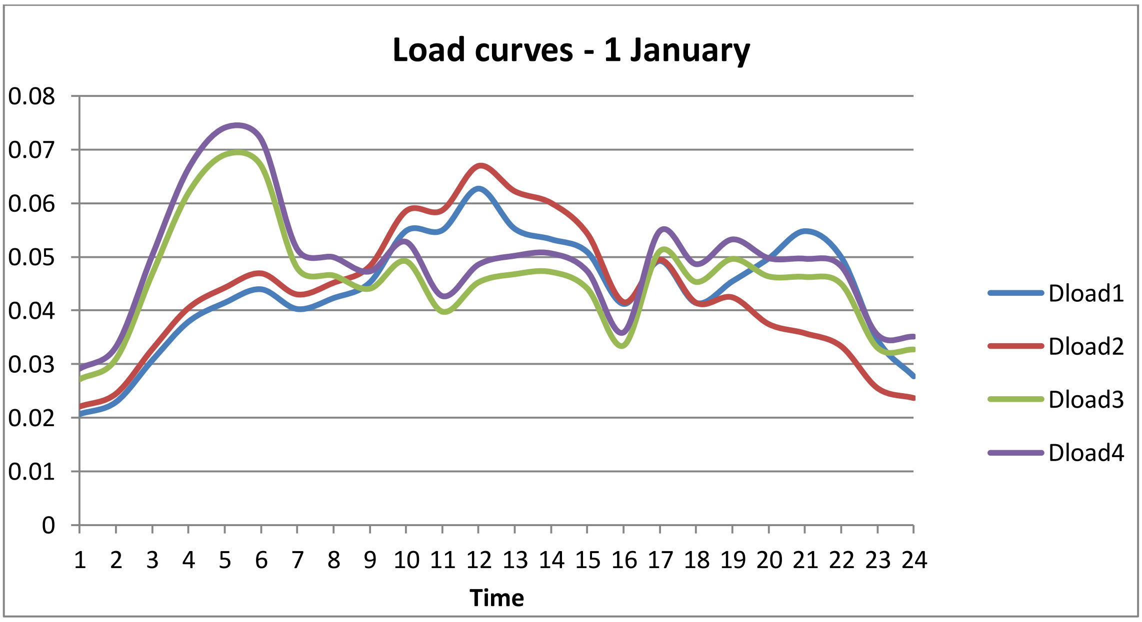


Figure 4.1. Load curves along 1 January

It can be seen that there seem to be two clear patterns, since curves *Dload1* and *Dload2* follow hardly parallel paths, and the same happens to curves *Dload3* and *Dload4*.

The chart above corresponds to a Monday. We find similar behaviors if we plot the rest of weekdays. The charts have not been included to avoid repetitive data. However, during weekend the behavior follows a different pattern, as it can be seen in the following figure, which corresponds to the first Saturday of the year:

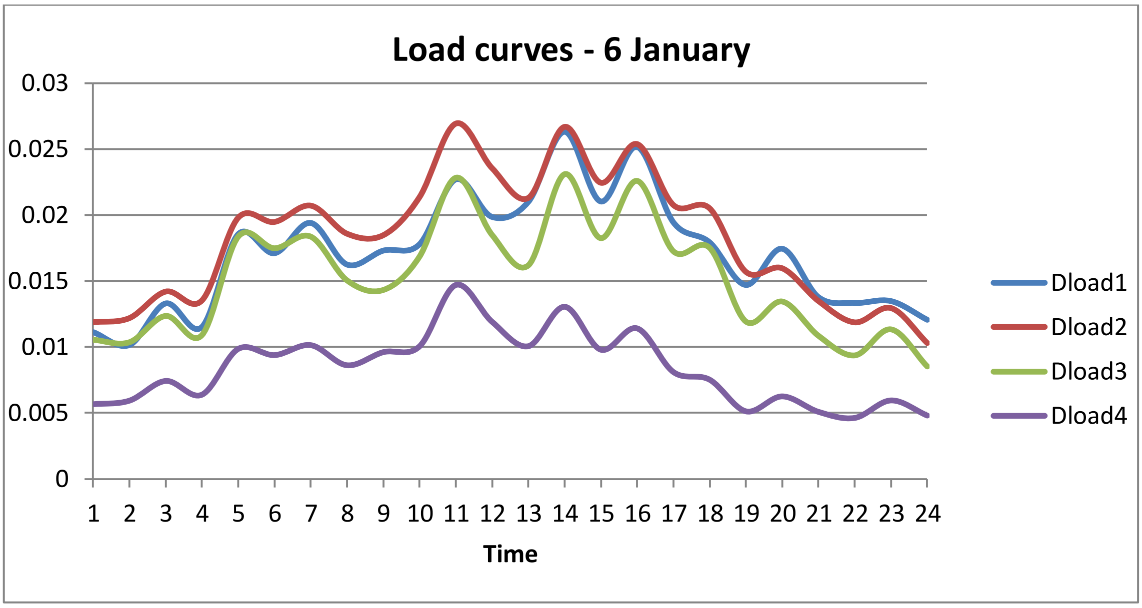


Figure 4.2. Load curves along 6 January

In this case we can see as *Dload4* takes clearly lower values to the ones of the rest of the curves. Despite this fact, it keeps still a clear parallelism with *Dload3*.

Another way of plotting load curves would be by time and day of the week. In the following chart, the values of the four load curves corresponding to the same day of the week and time are plotted, so we can see its evolution during the 52 weeks of the year (in case of Mondays 53 values will be plotted, since the considered year has 52 weeks and one day, being this extra day a Monday).

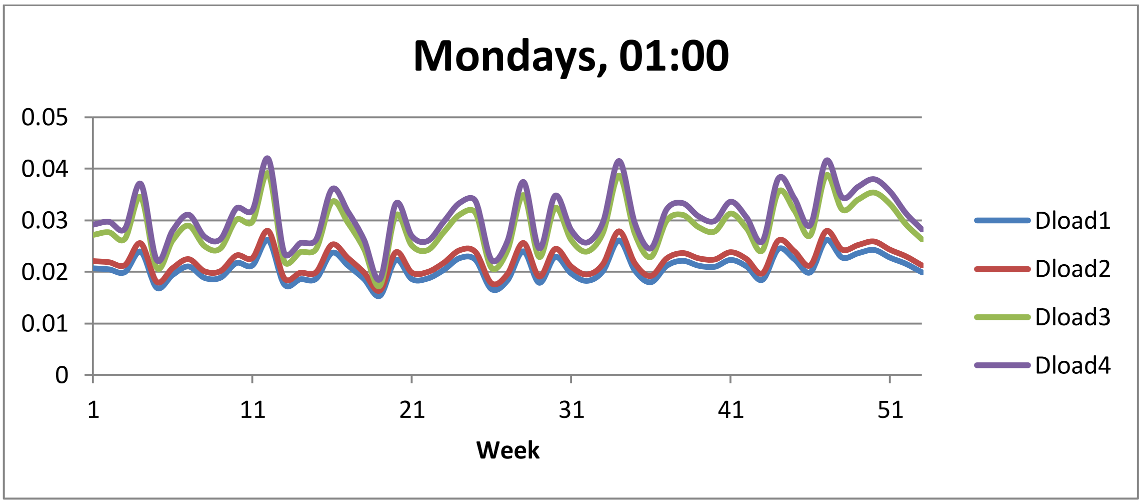


Figure 4.3. Load curves evolution along Mondays at 01:00

In this case, we can see that the four curves show parallelism, what did not happen with the previous way of representing them. The four curves are gathered in two groups, but all of them follow similar paths. This is the behavior at Mondays at 01:00. During the following hours, the behavior is similar, until at 07:00, when all the curves get closer:

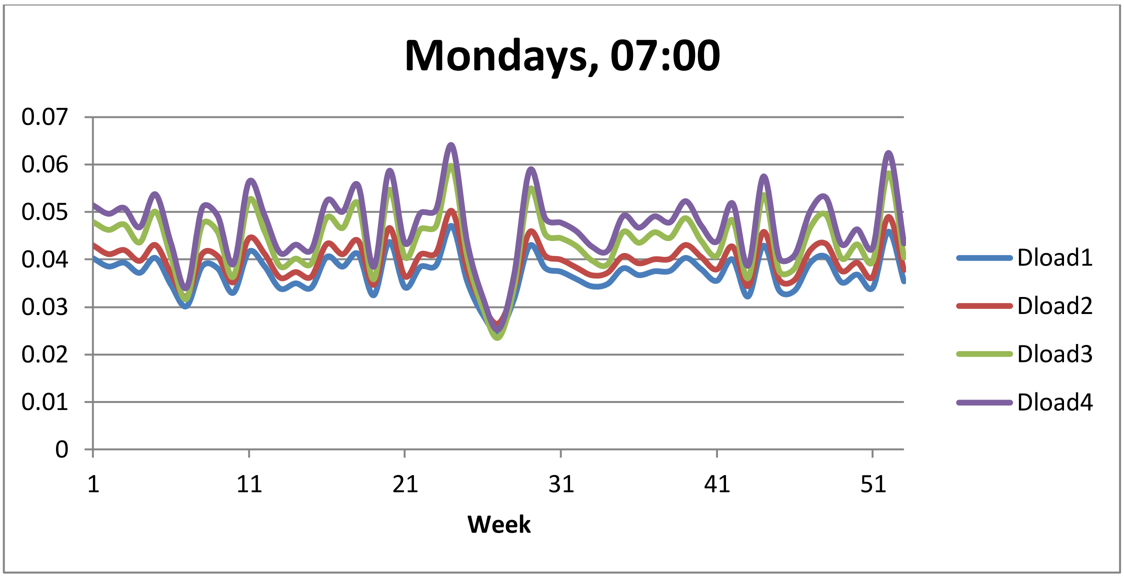


Figure 4.4. Load curves evolution along Mondays at 07:00

All along the day, the four curves show a great level of parallelism, but the distance between them changes. At some hours they look gathered in two groups like at 01:00 and at some others they are all closer to each other like at 07.00. A third behavior occurs at the last hours of the day, in which the value of Dload2 is noticeably lower to the others:

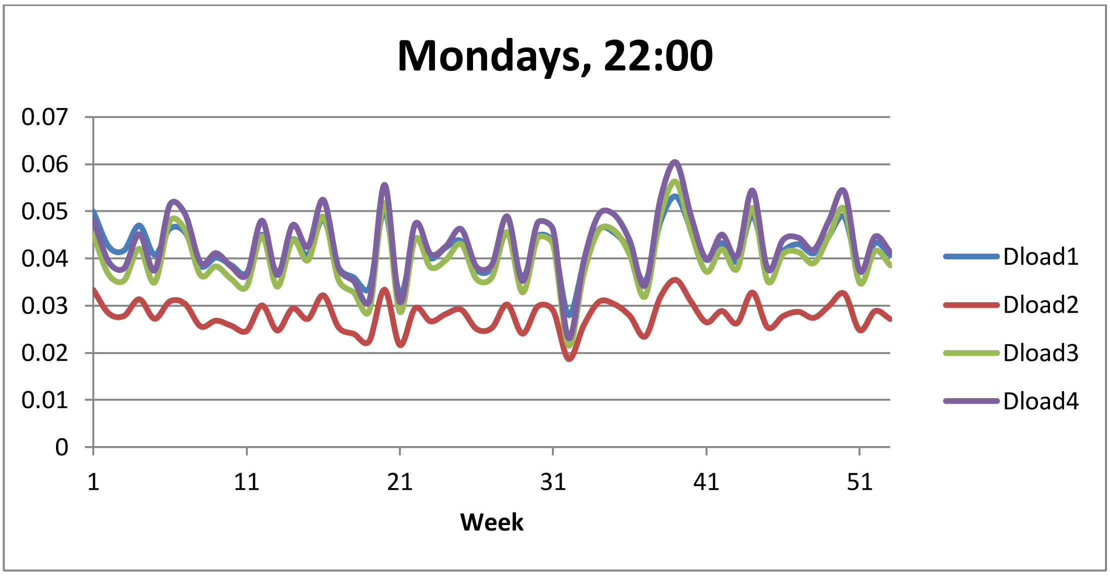


Figure 4.5. Load curves evolution along Mondays at 22:00

The previously commented behaviors of the load curves were based on Mondays' data. However, the same pattern is repeated for the rest of weekdays. For weekends, the pattern is a little different, as it can be seen in the following chart:

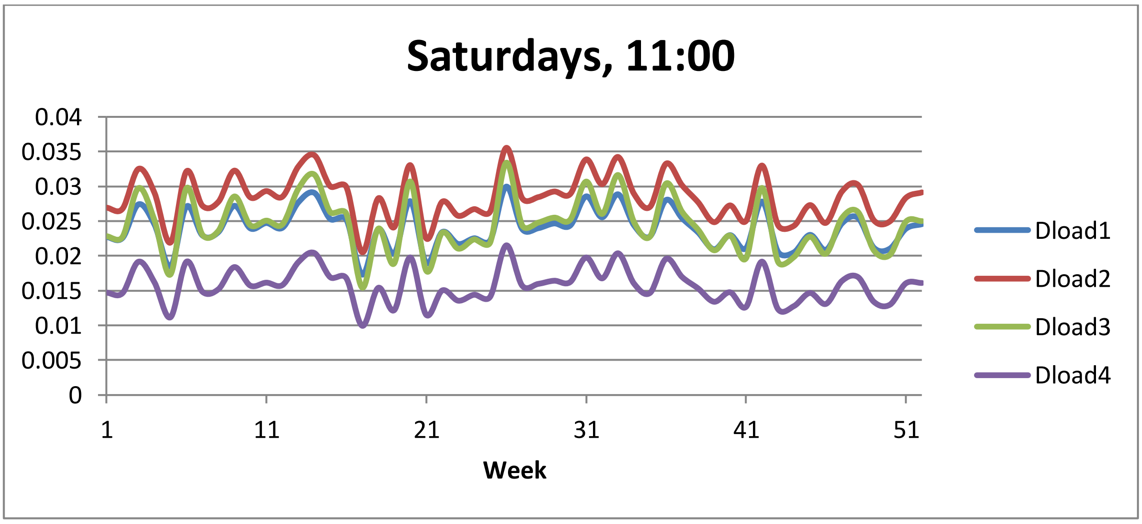


Figure 4.6. Load curves evolution along Mondays at 22:00

During weekends, at each time, there exists parallelism between the four curves, being the value of Dload4 noticeably lower to the rest.

4.1.2. Demand curves reduction

In order to assess the similarities between load curves, we have divided them with each other. It has been found that *Dload2* is the result of multiplying *Dload1* by a factor, and the same happens with *Dload4* respect to *Dload3*. These factors, which depend on time and on the kind of day (week day or weekend), are the following:

Time	Week days		Weekends	
	Dload2/Dload1	Dload4/Dload3	Dload2/Dload1	Dload4/Dload3
1:00	WD_{01}^{2} =1.0676	WD_{01}^{4} =1.0733	WE_{01}^{2} =1.0676	WE_{01}^{4} =0.5366
2:00	WD_{02}^{2} =1.0676	WD_{02}^{4} =1.0733	WE_{02}^{2} =1.201	WE_{02}^{4} =0.5724
3:00	WD_{03}^{2} =1.0676	WD_{03}^{4} =1.0733	WE_{03}^{2} =1.0676	WE_{03}^{4} =0.601
4:00	WD_{04}^{2} =1.0676	WD_{04}^{4} =1.0733	WE_{04}^{2} =1.1743	WE_{04}^{4} =0.5854
5:00	WD_{05}^{2} =1.0676	WD_{05}^{4} =1.0733	WE_{05}^{2} =1.0676	WE_{05}^{4} =0.5366
6:00	WD_{06}^{2} =1.0676	WD_{06}^{4} =1.0733	WE_{06}^{2} =1.1388	WE_{06}^{4} =0.5366
7:00	WD_{07}^{2} =1.0676	WD_{07}^{4} =1.0733	WE_{07}^{2} =1.0676	WE_{07}^{4} =0.552
8:00	WD_{08}^{2} =1.0676	WD_{08}^{4} =1.0733	WE_{08}^{2} =1.1438	WE_{08}^{4} =0.5724
9:00	WD_{09}^{2} =1.0676	WD_{09}^{4} =1.0733	WE_{09}^{2} =1.0676	WE_{09}^{4} =0.6708
10:00	WD_{10}^{2} =1.0676	WD_{10}^{4} =1.0733	WE_{10}^{2} =1.201	WE_{10}^{4} =0.59627
11:00	WD_{11}^{2} =1.0676	WD_{11}^{4} =1.0733	WE_{11}^{2} =1.1862	WE_{11}^{4} =0.644
12:00	WD_{12}^{2} =1.0676	WD_{12}^{4} =1.0733	WE_{12}^{2} =1.1862	WE_{12}^{4} =0.644
13:00	WD_{13}^{2} =1.1269	WD_{13}^{4} =1.0733	WE_{13}^{2} =1.0142	WE_{13}^{4} =0.6214
14:00	WD_{14}^{2} =1.1269	WD_{14}^{4} =1.0733	WE_{14}^{2} =1.0142	WE_{14}^{4} =0.5649
15:00	WD_{15}^{2} =1.0676	WD_{15}^{4} =1.0733	WE_{15}^{2} =1.0676	WE_{15}^{4} =0.5366
16:00	WD_{16}^{2} =1.0083	WD_{16}^{4} =1.0733	WE_{16}^{2} =1.0083	WE_{16}^{4} =0.5051
17:00	WD_{17}^{2} =1.0048	WD_{17}^{4} =1.0733	WE_{17}^{2} =1.0676	WE_{17}^{4} =0.4696
18:00	WD_{18}^{2} =1.0009	WD_{18}^{4} =1.0733	WE_{18}^{2} =1.1438	WE_{18}^{4} =0.4293
19:00	WD_{19}^{2} =0.9341	WD_{19}^{4} =1.0733	WE_{19}^{2} =1.0676	WE_{19}^{4} =0.4293
20:00	WD_{20}^{2} =0.75359	WD_{20}^{4} =1.0733	WE_{20}^{2} =0.9151	WE_{20}^{4} =0.4651
21:00	WD_{21}^{2} =0.65241	WD_{21}^{4} =1.0733	WE_{21}^{2} =0.9786	WE_{21}^{4} =0.4683
22:00	WD_{22}^{2} =0.66724	WD_{22}^{4} =1.0733	WE_{22}^{2} =0.8897	WE_{22}^{4} =0.4937
23:00	WD_{23}^{2} =0.7391	WD_{23}^{4} =1.0733	WE_{23}^{2} =0.9608	WE_{23}^{4} =0.5247
24:00	WD_{24}^{2} =0.8541	WD_{24}^{4} =1.0733	WE_{24}^{2} =0.8541	WE_{24}^{4} =0.5635

Table 4.2. Load curves multiplying factors

It has been checked that these factors are the same during the whole year. On weekdays these factors are much more regular while on weekends they change almost at every hour.

By means of the previous table, *Dload2* can be obtained having *Dload1* or vice versa, and the same happens with *Dload4* and *Dload3*; but it is not possible to reduce the four curves just into one, since no relation has been found between curves *Dload1* and *Dload3* or *Dload2* and *Dload4*. The obtained curves will be noted as $\widehat{Dload}2$ and $\widehat{Dload}4$.

The way of expressing *Dload2* and *Dload4* as a function of *Dload1* and *Dload3* is the following:

$$i\ corresponds\ to\ a\ week\ day\ and\ hour\ k \Longrightarrow\ \widehat{Dload}2_{\iota} = WD_k^2 \cdot Dload1_i$$

$$k = 01,02, \dots ,24; i\ \epsilon\ \{1, 8760\}$$

$$i\ corresponds\ to\ a\ weekend\ and\ hour\ k \Longrightarrow\ \widehat{Dload}2_{\iota} = WE_k^2 \cdot Dload1_i$$

$$k = 01,02, \dots ,24; i\ \epsilon\ \{1, 8760\}$$

$$\overline{\widehat{Dload}2} = \left[\widehat{Dload}2_1, \widehat{Dload}2_2, \dots , \widehat{Dload2}_{8760}\ \right]^T$$

$$i\ corresponds\ to\ a\ week\ day\ and\ hour\ k \Longrightarrow\ \widehat{Dload}4_{\iota} = WD_k^4 \cdot Dload3_i$$

$$k = 01,02, \dots ,24; i\ \epsilon\ \{1, 8760\}$$

$$i\ corresponds\ to\ a\ week\ day\ and\ hour\ k \Longrightarrow\ \widehat{Dload}4_{\iota} = WD_k^4 \cdot Dload3_i$$

$$k = 01,02, \dots ,24; i\ \epsilon\ \{1, 8760\}$$

$$\overline{\widehat{Dload}4} = \left[\widehat{Dload}4_1, \widehat{Dload}4_2, \dots , \widehat{Dload4}_{8760}\ \right]^T$$

To check whether the obtained $\widehat{Dload}2$and $\widehat{Dload}4$ are good approximations or not, the differences between norms with the original curves have been calculated:

	$Dload2$	$\widehat{Dload}2$	$Dload4$	$\widehat{Dload}4$
L^1 norm	292.0002	292.0035	292.0001	292.0030
Absolute error	0.0034		0.0029	
Relative error	1.1548e-005		1.0082e-005	

L^2 norm	3.3736699	3.3737116	3.5202534	3.5202933
Absolute error	4.1676e-005		3.9982e-005	
Relative error	1.2353e-005		1.1358e-005	
Infinity norm	0.078847	0.078848	0.085944	0.085945
Absolute error	1.2386e-006		9.4881e-007	
Relative error	1.5708e-005		1.1040e-005	

Table 4.3. Errors due to load curves simplification

From the table above it can be concluded that the reduction of input data from 4 load curves to 2 load curves can be successfully performed, since the relative errors of expressing *DLoad2* and *DLoad4* as a function of *Dload1* and *Dload3* are about $1\cdot10^{-5}$.

To assess how this input data reduction affects the computation of the system, 10 grid configurations' losses have been calculated using both the original input data and the new one. The configurations that have been used to check this fact are the same that were used to validate the Matlab code in *Chapter 2.*

BP	PP (kW)	Losses with $\widehat{Dload}2$ and $\widehat{Dload}4$ (kW)	Matlab code Losses (kW)	OpenDSS Losses (kW)
30	3332.7	154431.079	154431.372	154430.10
70	2908.1	141315.271	141315.519	141314.53
68	5759.4	137738.819	137739.075	137737.69
19	5979.4	156839.92	156840.212	156838.68
14	1462.5	175839.906	175840.237	175838.58
52	3711.5	140417.502	140417.614	140416.36
98	3383.5	141046.653	141046.996	141045.84
37	4873.9	143614.374	143614.636	143613.40
61	5342.0	135588.969	135589.229	135588.10
25	5678.5	151405.832	151406.092	151404.61

Table 4.4. Losses using the 3 ways of calculating them

The absolute and relative errors that result of comparing these results with the ones obtained with the Matlab code are the following:

		Error	
BP	PP (kW)	Absolute error (kW)	Relative error
30	3332.7	0.293	$1.899 \cdot 10^{-06}$
70	2908.1	0.248	$1.75626 \cdot 10^{-06}$
68	5759.4	0.256	$1.856 \cdot 10^{-06}$
19	5979.4	0.292	$1.863 \cdot 10^{-06}$
14	1462.5	0.331	$1.881 \cdot 10^{-06}$
52	3711.5	0.112	$7.981 \cdot 10^{-07}$
98	3383.5	0.343	$2.430 \cdot 10^{-06}$
37	4873.9	0.262	$1.825 \cdot 10^{-06}$
61	5342.0	0.260	$1.920 \cdot 10^{-06}$
25	5678.5	0.260	$1.720 \cdot 10^{-06}$

Table 4.5. Absolute and relative error

The implemented simplification of the input data has a nearly negligible effect on the final result of the problem.

The average computation time of the calculations with the initial input data was 3688.18 seconds. Solving the system introducing only 2 curves took as an average 3646.18 seconds to obtain the solution.

This means that reducing the input data implied saving 1,14% of the computation time, with a low effect on the final result. This simplification implies a really low reduction of computational cost. However, it allows reducing the amount of input data with a negligible effect on the final result.

4.2. Reduction at the solver

In the previous section, it was tried to reduce the input data for year analyses.

However, the proposed reduction did not permit a significant reduction of the computational cost. In this section, an analysis of the solutions will be carried out in order to being able to reduce the number of calculations.

Solving a particular problem in Time Mode implies obtaining a solution vector made up of 8760 components, one of them for each hour of the year. This fact allows calculating the grid associated losses with 1 hour precision, and implies massive calculations. Probably some of these calculations are unnecessary. This will be discussed along this section.

4.2.1. Avoidable losses calculations

In the following lines it will be studied whether it would be possible to do without some of the large amount of calculations Time Mode implies, since the possibility that there is no need to calculate losses at some hours of the year is taken into account. In case losses were independent of the power and position of the DG, then calculating them just in one configuration would be enough, and the precision of the results would not be affected.

A priori, it can be thought that it may happen that at night hours there were no differences between locating the DG at one node or another, or with different power values, because no solar light is available during these periods.

This will be checked with the 10 losses solutions of the already analyzed configurations. Each solution is made up with 8760 components, where the first one corresponds to the losses value at the first hour of 1 January and component number 8760 to the last hour of 31 December.

It has been observed that at certain hours all the 10 vectors have exactly the same value in its corresponding component. This fact happens at 2948 hours, which are distributed along the year as follows:

From 1,00 to 5,00	Every day
6,00	314 days (from 1-Jan to 13-May and 4-Jul to 31-Dec)
7,00	66 days (from 1-Jan to 21-Feb and from 18-Dec to 31-Dec)
From 8,00 to 20,00	Never
21,00	12 days (8, 9,15,16,27,28 Nov and 6, 10, 11, 14, 22, 26 Dec)
22,00	97 days (all along the year, especially in Winter)
23,00	281 days (all along the year, especially in Winter)
24,00	353 days (except some few days, especially in Summer)

Table 4.6. Distribution of the 2948 hours along the year

It is confirmed that during many night hours it would not be necessary to calculate the grid losses, since they could be calculated just once.

We have seen that in 2948 of the 8760 hours losses calculation could be avoided because they do not depend on the case we are considering. It has also been checked that not only losses are the same for all configurations of the grid in that hours, but voltages too. We cannot forget that losses are calculated from voltages, which are the solution of the mathematical model that was previously explained in *Chapter 2*. The way of working in this case is trying to improve the model behavior by observing the final results.

Apart from these 2948 hours, there are some others where losses in different configurations are really similar. If we pay attention to the difference between the losses maximum and minimum value in the 10 considered configurations, we can see that in some cases this value is really small, that is, the 10 values are really similar.

If we accept that up to one certain value of this difference, it could be considered as negligible, then the number of hours at which losses calculation would be unnecessary would increase.

In the following table it is shown how this number of hours increases depending on the tolerance accepted:

Tolerance	Number of hours
0	2948
$1 \cdot 10^{-9}$	2970
$1 \cdot 10^{-8}$	3131
$1 \cdot 10^{-7}$	3887
$1 \cdot 10^{-6}$	4002
$1 \cdot 10^{-5}$	4049
$1 \cdot 10^{-4}$	4156
$1 \cdot 10^{-3}$	4302
$1 \cdot 10^{-2}$	4399
$1.45 \cdot 10^{-2}$	4407
$1 \cdot 10^{-1}$	4417
1	4571

Table 4.7. Hours with no need of losses calculation

Observing the data we are working with, it can be seen that the 4407 values in which the difference between losses maximum and minimum value is the lowest are exactly the same as the 4407 values in which *Dsun* (value from the photovoltaic source load curve) is equal to zero.

A criterion that can be followed is considering that it is necessary to calculate losses only when *Dsun* is non-zero. That way, instead of working with vectors made up with 8760 components, we would be working with vectors of 4353 components. That would mean saving about half the calculations.

The hours in which *Dsun* is not equal to zero are:

- 1 January – 3 January: from 08:00 to 17:00
- 4 January – 21 February: from 08:00 to 18:00
- 22 February – 13 May: from 07:00 to 18:00
- 14 May – 3 June: from 06:00 to 18:00
- 4 June – 3 July: from 06:00 to 19:00
- 4 July – 6 August: from 07:00 to 19:00
- 7 August – 19 October: from 07:00 to 18:00
- 20 October – 17 December: from 07:00 to 17:00

- 18 December – 31 December: from 08:00 to 17:00

After having seen that there are some calculations that are likely to be avoided, the first proposed reduction for calculating losses in Time Mode is to calculate losses at 4353 hours instead of 8760. The 4407 losses values when *DSun* is equal to 0 are approximated as its average value from the 10 evaluated configurations.

The hours in which *Dsun* is nonzero are known, so adapting the Matlab code to calculate the losses at that specific hours does not present many difficulties. Only one calculation of the whole year will have to be made, in order to obtain the reference values for the 4407 hours in which *DSun* is equal to zero.

The modified Matlab code does not set the variable "*hour*" between 1 and 8760, but between 1 and 4353; and a vector called *Dsunnozero* with the position of the hours that need to be taken into account is introduced.

The result obtained using this procedure is a vector made up of 4353 components for each of the cases considered and a common vector for all the cases (made up of 4407 components).

The total losses for any analyzed grid configuration will be computed as:

$$Losses = L_0 + \sum_{i=1}^{4353} L_{DSun \neq 0}\ (i)$$

where

$L_0 = \sum_{i=1}^{4407} \overline{L_{Dsun=0}}(i) = 73380.67012\ \mathrm{kW}$ is the average value of losses for the 10 evaluated configurations at hours when *DSun=0.*

To compare the results, the same random pairs of data that were used in *Chapter 2* to validate the Matlab code have been used:

BP	PP (kW)	First reduction Losses (kW)	Matlab code Losses (kW)
30	3332,7	154431,372	154431,372
70	2908,1	141315,519	141315,519
68	5759,4	137739,075	137739,075
19	5979,4	156840,212	156840,212
14	1462,5	175840,237	175840,237
52	3711,5	140417,614	140417,614
98	3383,5	141046,996	141046,996
37	4873,9	143614,636	143614,636
61	5342,0	135589,229	135589,229
25	5678,5	151406,092	151406,092

Table 4.8 Losses using the 3 methods for 3 random pairs of data

Using three decimal values, the losses values with the proposed reduction and with Matlab seem to be exactly the same.

The absolute and relative errors obtained using the Matlab code and the modified version proposed in this chapter are the following:

		Error	
BP	PP (kW)	Absolute error (kW)	Relative error
30	3332,7	$3,964\cdot10^{-07}$	$2,567\cdot10^{-12}$
70	2908,1	$8,596\cdot10^{-07}$	$6,083\cdot10^{-12}$
68	5759,4	$1,301\cdot10^{-06}$	$9,448\cdot10^{-12}$
19	5979,4	$5,222\cdot10^{-07}$	$3,330\cdot10^{-12}$
14	1462,5	$1,076\cdot10^{-06}$	$6,118\cdot10^{-12}$
52	3711,5	$1,173\cdot10^{-06}$	$8,359\cdot10^{-12}$
98	3383,5	$1,911\cdot10^{-07}$	$1,355\cdot10^{-12}$
37	4873,9	$5,611\cdot10^{-07}$	$3,908\cdot10^{-12}$
61	5342,0	$6,043\cdot10^{-07}$	$4,457\cdot10^{-12}$
25	5678,5	$5,772\cdot10^{-07}$	$3,812\cdot10^{-12}$

Table 4.9.. Absolute and relative error

Avoiding half the calculations does not affect the precision of the results. The difference between using the whole Matlab code or the proposed adaptation is extremely small in precision terms.

The absolute differences between the original Matlab code and the adapted one are never larger than $1.3\cdot10^{-6}$ (that is a relative difference of $9\cdot10^{-12}$ between them).

Using the value L_0 was useful to approximate losses at the 10 configurations considered. To confirm whether this approximation is useful or not, it will be validated in 10 new configurations of the grid. As before, these 10 new configurations will be randomly selected.

	BP	PP
1	63	7235.7
2	10	1181.9
3	28	7177.8
4	55	6001.3
5	82	1064.0
6	87	4214.3
7	48	5218.3
8	73	4041.7
9	22	3584.9
10	45	2367.7

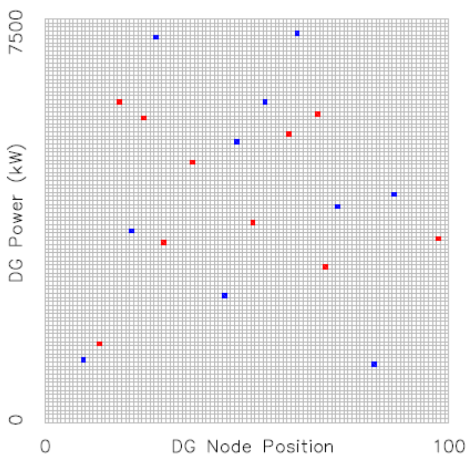


Figure 4.7. New random values of DG location and power (blue dots in the mesh)

BP	PP (kW)	First Reduction Losses (kW)	Matlab code Losses (kW)	Absolute error (kW)	Relative error
63	7235,7	147796,935	147796,935	$1{,}0328 \cdot 10^{-06}$	$6{,}9882 \cdot 10^{-12}$
10	1181,9	179141,375	179141,375	$1{,}1026 \cdot 10^{-06}$	$6{,}1551 \cdot 10^{-12}$
28	7177,8	150400,495	150400,495	$2{,}6129 \cdot 10^{-07}$	$1{,}7373 \cdot 10^{-12}$
55	6001,3	137674,659	137674,659	$6{,}6921 \cdot 10^{-07}$	$4{,}8608 \cdot 10^{-12}$
82	1064,0	163250,332	163250,332	$6{,}3033 \cdot 10^{-07}$	$3{,}8611 \cdot 10^{-12}$
87	4214,3	137526,952	137526,952	$4{,}5524 \cdot 10^{-07}$	$3{,}3102 \cdot 10^{-12}$
48	5218,3	137846,221	137846,221	$8{,}7186 \cdot 10^{-07}$	$6{,}3249 \cdot 10^{-12}$
73	4041,7	135378,646	135378,646	$5{,}1383 \cdot 10^{-07}$	$3{,}7955 \cdot 10^{-12}$
22	3584,9	159329,309	159329,309	$6{,}1179 \cdot 10^{-07}$	$3{,}8398 \cdot 10^{-12}$
45	2367,7	153406,297	153406,297	$9{,}5929 \cdot 10^{-07}$	$6{,}2533 \cdot 10^{-12}$

Table 4.10. Validation results

The table above shows that the approximation provides excellent results not only with the configurations used to obtain the L_0 value, but also in new different solutions.

Once L_0 has been obtained, each new solution is obtained with really high precision and an important reduction of computational cost. Time spent to obtain a solution with the whole Matlab code is in average 3680 seconds. With the proposed reduction, time spent in average was 1830 seconds. This means that the reduction achieves really good results and allows saving 50,27% of the time. As it was expected, the time saved when avoiding needless calculations tends to 50.3% ((8760-4353)/8760 = 0.503).

4.2.2. PCA-based reduction

In the previous section it was proved that a whole year analysis can be performed with high precision using only 4353 values instead of 8760. This reduction was made by proving that when solar intensity was 0, there was no need to calculate the losses because it was hardly the same for all the grid configurations.

The objective of the present section is to present a way of reducing to a smallest value the amount of hours needed to perform a year analysis.

4.2.2.1. Procedure

The procedure that has been followed is to analyze the losses values of the 10 grid configurations which have been previously used. The analyzed data are 10 vectors (*L)* of 4353 components, since only hours with solar intensity have been considered.

$$L_i \in \mathbb{R}_{4353x1}\,; i = 1,2,3, \dots ,10$$

These ten vectors can be put one next to the other in a matrix (*LP*).

$$LP = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \overline{L_1} & \overline{L_2} & \dots & \overline{L_{10}} \\ \vdots & \vdots & & \vdots \end{bmatrix} \in \mathbb{R}_{4353X10}$$

To start the reduction process, these ten vectors are reduced into smaller ones. Its components will be grouped depending on its *DSun* value. Ten groups are defined (P01, P02, … , P10), where P01 elements are the ones in which $\frac{DSun}{\max(DSun)} \leq 10\%$, P02 elements are the ones in which $10\% < \frac{DSun}{\max(DSun)} \leq 20\%$, etc. Each group P_i will be composed of m_i elements (*i*=1, 2, … , 10)

This way, each of the 4353 hours is placed in one group:

Group name	DSun/DSun$_{max}$	DSun value (kW)	m (number of elements)
P_{01}	0-10%	0 - 19,63	854
P_{02}	10-20%	19,63 – 39,26	577
P_{03}	20-30%	39,26 – 58,89	601
P_{04}	30-40%	58,89 – 78,52	450
P_{05}	40-50%	78,52 – 98,15	477
P_{06}	50-60%	98,15 – 117,78	396
P_{07}	60-70%	117,78 – 137,41	405
P_{08}	70-80%	137,41 – 157,04	351
P_{09}	80-90%	157,04 – 176,67	188
P_{10}	90-100%	176,67 – 196,3	54

Table 4.11. Groups P01 to P10 definition

For each of the previous groups, we can obtain its *LP* matrix:

$$LP^{P_i} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \overline{L_1}^{P_i} & \overline{L_2}^{P_i} & \dots & \overline{L_{10}}^{P_i} \\ \vdots & \vdots & & \vdots \end{bmatrix}; i = 1, 2, \dots, 10$$

Using the Principal Component Analysis, as it was explained in *Chapter 3*, these matrices can be reproduced with high precision using enough eigenvectors of the covariance matrix of the zero mean design matrix.

A particular case in the application of the Principal Component Analysis is when a single eigenvalue accounts hardly the 100% of the variance in the sample data. In this case, the vectors composing the zero mean design matrix would be all proportional to each other.

For each matrix LP^{P_i}, a Principal Component Analysis is performed.

The matrix X^{P_i} is defined for each group as:

$$X^{P_i} = \left[\overline{L_1}^{P_i} - \bar{\mu}\ \ \overline{L_2}^{P_i} - \bar{\mu}\ \ \ldots\ \overline{L_{10}}^{P_i} - \bar{\mu}\right];\ \bar{\mu} = \frac{1}{10}\sum_{j=1}^{10} \overline{L_j}^{P_i}\ ; i = 1, 2, \ldots, 10$$

The eigenvalues of the covariance matrix of X^{P_i} are obtained: $\{\lambda_1^{P_i}, \lambda_2^{P_i}, \ldots, \lambda_{m_i}^{P_i}\}$

The percentage of variance accounted by the first eigenvalue is

$$\frac{\lambda_1^{P_i}}{\sum_{j=1}^{m_i} \lambda_j^{P_i}}$$

Table 4.12 shows the results of computing this value for the ten considered groups:

Group name	DSun/DSun$_{max}$	DSun value (kW)	Variance accounted by the 1st eigenvalue	Elements
P01	0-10%	0 - 19,63	99,98%	854
P02	10-20%	19,63 – 39,26	99,93%	577
P03	20-30%	39,26 – 58,89	99,73%	601
P04	30-40%	58,89 – 78,52	99,30%	450
P05	40-50%	78,52 – 98,15	97,64%	477
P06	50-60%	98,15 – 117,78	94,21%	396
P07	60-70%	117,78 – 137,41	83,77%	405
P08	70-80%	137,41 – 157,04	66,50%	351
P09	80-90%	157,04 – 176,67	56,82%	188
P10	90-100%	176,67 – 196,3	61,38%	54

Table 4.12. Results of applying PCA to groups P01 to P10

A usual criterion to determine the dimensionality of a set of data is considering the amount of eigenvalues that represent 95% of the variance in the data. In this section, it will be considered that the analyzed data is one-dimensional when a unique eigenvalue represents more than 98% of the variance.

From the results in Table 4.12, it can be concluded that data in groups P01 to P04 is one-dimensional, since the first eigenvalue of the covariance matrix represents more than 98% of the variance in the data.

In these four groups, the data being one-dimensional can be expressed as:

$X_j^{P_i} \propto X_k^{P_i};\ j, k = 1,2,3, \ldots, 10; j \neq k\ ; i = 1, 2, 3, 4$

Dividing each of X^{P_i} columns by any of its other columns will results in a new vector whose components will be nearly constant. By fixing a column (*j*) by which all the others will be divided, we can define the *k* matrix as:

$$k^{P_i} = \begin{bmatrix} \frac{\overline{X_1}^{P_i}}{\overline{X_J}^{P_i}} & \frac{\overline{X_2}^{P_i}}{\overline{X_J}^{P_i}} & \cdots & \frac{\overline{X_J}^{P_i}}{\overline{X_J}^{P_i}} & \cdots & \frac{\overline{X_{10}}^{P_i}}{\overline{X_J}^{P_i}} \end{bmatrix} = \begin{bmatrix} k_{1,1}{}^{P_i} & k_{1,2}{}^{P_i} & & 1 & & k_{1,10}{}^{P_i} \\ k_{2,1}{}^{P_i} & k_{2,2}{}^{P_i} & \cdots & 1 & \cdots & k_{2,10}{}^{P_i} \\ \vdots & \vdots & & \vdots & & \vdots \\ k_{m_i,1}{}^{P_i} & k_{m_i,2}{}^{P_i} & & 1 & & k_{m_i,10}{}^{P_i} \end{bmatrix}$$

Column j

Then, it will be possible to find an approximation of X^{P_i}, to which we will refer as $\widehat{X^{P_i}}$:

$$\widehat{X^{P_i}} = \left[k_{rep,1}{}^{P_i} \cdot \left(\overline{L_J}^{P_i} - \bar{\mu} \right) \quad k_{rep,2}{}^{P_i} \cdot \left(\overline{L_J}^{P_i} - \bar{\mu} \right) \quad \cdots \quad k_{rep,10}{}^{P_i} \cdot \left(\overline{L_J}^{P_i} - \bar{\mu} \right) \right]$$

where

$k_{rep}{}^{P_i} = [k_{rep,1}{}^{P_i}\ k_{rep,2}{}^{P_i} \cdots k_{rep,10}{}^{P_i}]$ is the representative row of the matrix *k* in group P_i.

To find the representative row of *k*, the following procedure is carried out:

h is the representative row of *k* ⟺ $\left(\sum_{n=1}^{10} k_{h,n} - \sum_{n=1}^{10} \overline{k_n} \right)$ is minimum

Finally, we can obtain an approximation of the initial LP^{P_i} matrix, to which we will refer as $\widehat{LP^{P_i}}$:

$$\widehat{LP^{P_i}} = \left[\widehat{X_1^{P_i}} + \bar{\mu} \quad \widehat{X_2^{P_i}} + \bar{\mu} \quad \ldots \quad \widehat{X_{10}^{P_i}} + \bar{\mu} \right]$$

The previously explained process implies that for groups that satisfy the condition ($\lambda_1^{P_i}$ representing more than 98% of the variance) can be approximated just with one single value (the component corresponding to the representative row). In the group division that has been presented, only 4 of the groups satisfied the condition. Groups P_{05} to P_{10} will be divided into smaller subgroups with new criteria in order to obtain new groups that satisfy the condition. The criterion to determine the first groups was the value of *DSun.*

4.2.2.2. Subgroups definition

The input data that determines the losses value at each time are the DG node and power, the demand load at that moment and the DG load. Due to the way the analysis is performed, the DG node and power are already set for each solution vector. The demand load value is defined by the demand load curves (*DLoad1, …, DLoad4)* and the DG load is defined by the DG load curve (*DSun)*, which change hourly.

In a group with similar *DSun* values, the only parameter that can make the solution vary is the value of the demand load curves. There are 4 load curves, but it has been already seen that 2 of them follow the same pattern. In first place, the remaining groups will be divided depending on its *Dload1* value.

Group name	DLoad1/DLoad1$_{max}$	Variance accounted by the 1st eigenvalue	Elements
P0501	0-25%	96,71%	12
P0502	25-50%	62,73%	142
P0503	50-75%	99,73%	288
P0504	75-100%	99,94%	35
P0601	0-25%	99,83%	11
P0602	25-50%	76,95%	112
P0603	50-75%	99,17%	243
P0604	75-100%	99,87%	30
P0701	0-30%	99,46%	45
P0702	30-50%	95,42%	85
P0703	50-70%	97,04%	190
P0704	70-100%	99,53%	85
P0801	0-30%	99,62%	29
P0802	30-50%	98,33%	99
P0803	50-70%	90,00%	139
P0804	70-100%	98,81%	84
P0901	0-30%	99,79%	15
P0902	30-50%	99,04%	44
P0903	50-70%	88,06%	79
P0904	70-100%	96,85%	50
P1001	0-40%	99,74%	14
P1002	40-60%	98,64%	6
P1003	60-80%	89,86%	28
P1004	80-100%	97,54%	6

Table 4.13. Results of applying PCA to the remaining groups

Each of the 6 remaining groups has been divided into 4 sub-groups. For example, group P05 will be divided in sub-groups P0501, P0502, P0503 and P0504. To define the groups it has been taken into account that the higher *DSun* was, it was less frequent to have low *Dload1* values. The definition of these subgroups and the result after doing a PCA analysis is summarized in Table 4.13.

14 of de 24 cases satisfy the condition, so they will be later on reduced to one value. The definition of these groups will permit to compute 1024 values with only 14. After this second analysis, there are still some groups which cannot be reduced to one single value (the ones written in bold). In order to reduce even more the data, a third reduction criterion is included. The value of *Dload3* will be considered in this third reduction.

Following the same procedure as before, each remaining group is divided in smaller groups. For example, P0501 is divided in two sub-groups (P050101 and P050102):

Group name	$DLoad3/DLoad3_{max}$	Variance accounted by the 1st eigenvalue	Elements
P050101	<20%	98,27%	5
P050102	>20%	98,81%	7
P050201	<30%	90,07%	58
P050202	30-40%	85,78%	71
P050203	>40%	99,16%	13
P060201	<25%	99,49%	12
P060202	25-30%	89,40%	30
P060203	30-35%	87,45%	39
P060204	>35%	73,40%	31
P070201	<35%	98,15%	49
P070202	>35%	95,44%	36
P070301	<40%	95,54%	28
P070302	40-50%	98,34%	86
P070303	>50%	99,22%	76
P080301	<40%	94,29%	45
P080302	40-50%	95,84%	61
P080303	>50%	98,51%	33
P090301	<40%	94,73%	26
P090302	>40%	95,59%	53
P090401	<60%	98,46%	28
P090402	>60%	99,27%	22
P100301	<50%	95,39%	10
P100302	>50%	95,20%	18
P100401	<64%	99,39%	3
P100402	>64%	98,49%	3

Table 4.14. Results of applying PCA to the 3rd classification groups

After this analysis, there are 13 groups which still do not satisfy the condition of proportionality. The elements of these groups will be computed without any reduction. In 12 cases, the first eigenvalue accounts for more than 98% of the variance in the group. It supposes reducing 337 values to 12.

The proposed reduction will need 536 calculations to compute a whole year total losses value (506 from the 13 irreducible groups and 30 from the valid groups)

4.2.2.3. Computation

Once the valid groups have been defined, matrix $\widehat{LP^{P_i}}$ is obtained with the procedure explained in Section 4.2.2.1.

The losses value for each group is computed as:

$$Losses_k^{P_i} = \sum_{i=1}^{m_i} LP_{i,k}^{P_i}; \quad k = 1, 2, \ldots, 10$$

The solution whose average values were used to obtain matrix *k* was selected randomly. It was the one corresponding to node 37 with a power of 4873,9 kW (*j=8*). The following table shows the average error when obtaining the group losses for the 30 valid groups:

Group	Hour associated to representative k row	Average relative error
P01	4015	0.00062122
P02	7977	0.00379661
P03	2096	0.01329955
P04	7143	0.02117635
P0503	873	0.00255501
P0504	2243	0.0006803
P0601	1162	0.00147455
P0603	2558	0.00236231
P0604	6756	0.00030851
P0701	3010	0.00162735
P0704	5555	0.0018565
P0801	1500	0.00170573
P0802	5339	0.0085528
P0804	6396	0.00069963
P0901	301	0.04429127

Group	Hour associated to representative k row	Average relative error
P0902	3157	0.00788193
P1001	7524	0.00311641
P1002	612	0.0133707
P050101	5193	0.06799533
P050102	298	0.01904836
P050203	1713	0.01127254
P060201	7382	0.00375253
P070201	2003	0.00460558
P070302	879	0.00099835
P070303	5050	0.00074075
P080303	1858	0.00078638
P090401	900	0.00214085
P090402	1115	0.00122903
P100401	876	0.0027079
P100402	756	0.02544808

Table 4.15. Average error and representative hour for each group

Once these results are obtained, the total losses value can be obtained as follows:

$$Losses_k = L_0 + \sum_{i=1}^{30} Losses_k^{P_i} + \sum_{j=1}^{506} Losses_k^{j}$$

Where $i = 1, \ldots, 30$ refers to the 30 valid groups and $j = 1, \ldots, 506$ to the 506 irreducible hours.

The whole year computation of losses using this procedure provides these results:

L	Aproximated Losses (kW)	Matlab code Losses (kW)	Absolute Error (kW)	Relative Error (%)
1	154014,000	154431,372	417,372113	0,00270264
2	141173,218	141315,519	142,301219	0,00100698
3	138640,098	137739,075	901,023239	0,00654152
4	156635,374	156840,212	204,838235	0,00130603
5	175081,508	175840,237	758,729461	0,00431488
6	140347,011	140417,614	70,6028598	0,00050281
7	141235,345	141046,996	188,349473	0,00133537
8	143614,636	143614,636	7,5115E-05	$5{,}2303\cdot10^{-10}$
9	136189,289	135589,229	600,059999	0,00442557
10	151310,504	151406,092	95,5880603	0,00063134

Table 4.16. Results for the sample vectors

The results show that the proposed method provides quite good approximated results, taking into account that they were obtained with 536 calculations instead of 8760. The reference value is, as expected, obtained with a high precision (in fact, it is not approximated). The other vectors are approximated with different results. The maximum error is obtained for the case with the lowest value of losses, and it is around 0,65%.

4.2.2.4. Validation

To validate the results of the proposed procedure, it has been tested in 10 new configurations different to the ones in the reference sample. These new configurations are the same which were used in *Section 4.2.1* to check the results when the first reduction was carried out.

The results of this validation, assessing the absolute and relative errors, are shown in the table below:

BP	PP (kW)	Approximated Losses (kW)	Matlab Losses (kW)	Absolute error (kW)	Relative error
63	7235,7	149253,80	147796,4	1456,86	0,009857
10	1181,9	178350,27	179141,38	791,11	0,004416
28	7177,8	150710,41	150400,49	309,92	0,002060
55	6001,3	138401,39	137674,66	726,73	0,005279
82	1064,0	162635,49	163250,33	614,84	0,003766
87	4214,3	138000,80	137526,95	473,85	0,003446
48	5218,3	138155,30	137846,22	309,08	0,002242
73	4041,7	135650,95	135378,65	272,30	0,002011
22	3584,9	158846,31	159329,31	483,00	0,003031
45	2367,7	152951,18	153406,30	455,12	0,002967

Table 4.17. Validation results

It must be taken into account the reduction of computational cost implies a certain loss in precision. In the analyzed cases, the maximum error has been about 1%. Considering this error as acceptable, it has been proved that the proposed procedure can be applied in to approximate solutions out of the considered sample.

The proposed simplification requires performing a whole year calculation and 536 calculations for each new configuration that wants to be calculated. This implies saving 93.88% of time in each new solution. An analysis of many grid configurations can be performed reducing highly the computational cost of the operations.

This simplified way of calculating losses in a grid provides a good idea of how losses are when the position and power of the DG is changed, but to obtain a really precise result for any configuration, an analysis considering the 4353 hours with sunlight should be performed.

4.2.2.5. Sample size variation

The proposed procedure was applied to a sample matrix *LP* composed by 10 different solutions. When it was validated with solution that did not belong to the sample, the results showed that the approximated results were quite good. In this section, the sample size will be varied to see how does this affect to the results.

In first place, the procedure will be applied with a matrix *LP* composed by 20 different solutions. Since the groups' definition will be the same, it is expected that less groups will satisfy the condition to be reducible. The obtained error in the approximation is expected to be lower.

Then, the same procedure will be applied again with a smaller sample (only 2 solutions). It is expected that a very small sample will be not representative of the problem domain, so validating the procedure in this case with solutions out of the sample will not provide good results.

- **Sample of 20 different solutions:**

The sample that composes matrix *LP* are the initial 10 solution vectors plus the 10 solution vectors which were used to validate the procedure. After applying PCA analysis to groups P01-P10 (1^{st} classification) and groups P0501-P1004 (2^{nd} classification) the groups classified as valid were exactly the same as when the sample was smaller. For the 3^{rd} classification groups there appeared some small differences.

The following tables show the variance accounted by the 1^{st} eigenvalue in all the groups considered, and the results obtained by computing the procedure:

Group	Variance accounted by 1st eig.	Group	Variance accounted by 1st eig.	Group	Variance accounted by 1st eig.	Group	Variance accounted by 1st eig.
P01	99,98%	**P0602**	86,76%	P1001	99,83%	**P070301**	93,62%
P02	99,91%	P0603	98,54%	P1002	99,29%	**P070302**	96,83%
P03	99,63%	P0604	99,80%	**P1003**	89,30%	P070303	98,51%
P04	98,98%	P0701	99,68%	**P1004**	95,34%	**P080301**	95,28%
P05	96,26%	**P0702**	97,55%	P050101	98,95%	**P080302**	93,55%
P06	90,52%	**P0703**	94,35%	P050102	99,34%	**P080303**	97,18%
P07	74,25%	P0704	99,17%	**P050201**	94,65%	**P090301**	97,03%
P08	59,07%	P0801	99,76%	**P050202**	84,67%	**P090302**	95,44%
P09	67,35%	P0802	99,06%	P050203	98,42%	**P090401**	97,39%
P10	73,52%	**P0803**	85,20%	P060201	99,71%	P090402	98,48%
P0501	97,84%	P0804	98,03%	**P060202**	94,16%	**P100301**	96,82%
P0502	65,47%	P0901	99,86%	**P060203**	93,18%	**P100302**	93,68%
P0503	99,57%	P0902	99,43%	**P060204**	74,38%	P100401	98,93%
P0504	99,90%	**P0903**	90,18%	P070201	98,98%	**P100402**	97,08%
P0601	99,90%	**P0904**	94,07%	**P070202**	97,62%		

Table 4.18. Group analysis with new 20 solutions sample

BP	PP (kW)	Approximated Losses (kW)	Matlab Losses (kW)	Absolute error (kW)	Relative error
30	3332,7	154670,402	154431,37	239,030	0,0015478
70	2908,1	141399,029	141315,52	83,510	0,00059094
68	5759,4	137312,411	137739,08	426,663	0,00309762
19	5979,4	156970,75	156840,21	130,538	0,0008323
14	1462,5	176321,326	175840,24	481,089	0,00273594
52	3711,5	140453,869	140417,61	36,255	0,00025819
98	3383,5	140979,053	141047,00	67,942	0,0004817
37	4873,9	143614,636	143614,64	$5{,}616\cdot10^{-07}$	$3{,}9104\cdot10^{-12}$
61	5342,0	135290,958	135589,23	298,271	0,00219981
25	5678,5	151468,165	151406,09	62,073	0,00040998
63	7235,7	147125,969	147796,40	670,966	0,00453979
10	1181,9	179650,788	179141,38	509,413	0,00284364
28	7177,8	150257,315	150400,49	143,180	0,00095199
55	6001,3	137317,324	137674,66	357,336	0,00259551
82	1064,0	163629,901	163250,33	379,569	0,00232507
87	4214,3	137311,889	137526,95	215,063	0,00156379
48	5218,3	137687,077	137846,22	159,145	0,00115451
73	4041,7	135248,757	135378,65	129,889	0,00095945
22	3584,9	159613,504	159329,31	284,195	0,0017837
45	2367,7	153669,081	153406,30	262,784	0,00171299

Table 4.19. Results with 20 solutions sample

With this sample, instead of 30 valid groups there are only 26 valid groups. The non-linear system of equations needs to be solved 682 times because of that. Having this bigger sample leads to the fact that the reduction in computational cost will be lower. In precision terms, the maximum error is about 0,45%. Comparing the results of using a 10 vectors sample with a 20 vectors sample, we obtain the following differences:

	10 vectors sample	20 vectors sample
Maximum relative error	0,654%	0,454%
Average relative error (without considering the reference vector)	0,253%	0,171%
Valid groups	30	26
Number of calculations needed	536	682
Saved time of calculation	93,88%	92,21%

Table 4.20. Comparison between different sample results

- **Sample of 2 different solutions:**

In this case it is being checked what the result is when only the first two solution vectors are taken to form matrix *LP.* When PCA is performed, the first classification groups satisfy in all cases the condition of proportionality, and in all ten cases the variance accounted by the first eigenvalue is equal to 100,00%. Computing the procedure with this sample basis provides the following results:

BP	PP (kW)	Approximated Losses (kW)	Matlab Losses (kW)	Absolute error (kW)	Relative error
30	3332,7	154 431,37	154431,37	0,000291	$1{,}881\cdot10^{-09}$
70	2908,1	141 315,52	141315,52	0,000187	$1{,}321\cdot10^{-09}$
68	5759,4	39560,04	137739,08	98179,037	0,713
19	5979,4	128895,55	156840,21	27944,663	0,178
14	1462,5	170864,27	175840,24	4975,964	0,028
52	3711,5	133227,072	140417,61	7190,542	0,051
98	3383,5	109523,39	141047,00	31523,609	0,223
37	4873,9	121380,03	143614,64	22234,608	0,155
61	5342,0	70807,26	135589,23	64781,968	0,478
25	5678,5	122917,89	151406,09	28488,120	0,188

Table 4.21. Results with a 2 solutions sample

It is immediate to see that in this case the sample was too small to provide good results. The approximation for the two vectors in the sample was excellent, but it was not able to approximate solutions out of the sample. Errors up to 71,3% appear.

CHAPTER 5. CONCLUSIONS

The development of this thesis introduces the concept of Smart Grids. It has been seen how the implementation of these grids will lead to more efficient electrical systems and will facilitate the generalization of Distributed Generation.

It was proved that, with the explained governing equation as a basis, the calculation code developed at UPC was able to reproduce the behavior of a system with incorporated DG obtaining the same results as the ones highly assessed software provided. The code is able to perform with precision annual simulations of the grid's behavior, for each considered position and power of the photovoltaic source.

Performing these analyses took about 1 hour for each grid configuration. Trying to reproduce all the possibilities would lead to spending 750.000 hours in computation time (if power was taken with entire values). As a consequence of that, it is considered important to find ways to reduce the computational cost.

The reduction in the input data permitted reducing the 5 input curves to 3. This fact did not permit any improvement in the calculations efficiency at first, but later on was used to determine the 3 criteria to develop the PCA-based reduction procedure.

The first proposed way to reduce the time of calculation is performing annual analysis without calculating the losses at those hours in which the photovoltaic generator is not providing power to the system. It has been proved that, in those hours, losses in all grid configurations can be considered the same. This first reduction permits saving 50,27% of computation time providing excellent results (relative errors were about $1{\cdot}10^{-12}$). After the first reduction, performing a year analysis of losses in the grid requires solving the presented non-linear system 4353 times instead of 8760.

Then, a second way to reduce the computational cost is proposed. In this case, the obtained result is an approximation that provides errors about 1% as maximum. The procedure consist is gathering the 4353 hours in groups so that losses solution vectors present a high proportionality level between them when subtracting the sample average value (in a sample of 10 solutions). The level of proportionality is given by the performance of a Principal Component Analysis to each group of solutions. Each group which satisfies the set proportionality condition is approximated with a single losses value.

The proposed procedure permits a reduction of 93,88% in computational cost (the non-linear system needs to be computed only 536 times), providing errors lower than 0.65% when reproducing the sample vectors, and lower than 1% when new solutions were approximated.

When the sample size is increased to 20, some precision is gained (maximum error goes from 0,65% to 0,45%) but the reduction of computational time is also lower (an annual analysis is performed with 682 hour calculations). When the sample is too small (only 2 solutions), the procedure is not valid to approximate solutions out of the sample.

The objective of being able to reduce the computational cost for annual losses analysis in electrical grids has been satisfied. The first proposed reduction allows obtaining results with excellent precision, while the second one allows obtaining an approximation of the results with a low time of calculation.

CHAPTER 6. REFERENCES

[1] Dugan, R., EPRI, K. T., & McDermott, T. (2011). *An open source platform for collaborating on smart grid research.* San Diego, CA: Power and Energy Society General Meeting, 2011 IEEE, 1-7

[2] Dunteman, George (1989) *Principal Component Analysis.* Sage Publications, Chapters 1-6

[3] EU Comission Task Force for Smart Grids (2010). *Functionalities of smart grids and smart meters. Final deliverable.* European Comission, 4-14

[4] Florentin, Eric and Díez, Pedro (2012) *Adaptive reduced basis strategy based on goal oriented error assessment for stochastic problems.* Computer Methods in Applied Mechanics and Engineering, Elsevier, 225-228

[5] García Domínguez, Alejandro (2014). *Optimal location of distributed generators in electrical grids.* Universitat Politècnica de Catalunya, 5-15

[6] Guerra, Luis Gerardo. (2012) *Localización Óptima de Generación Distribuida en Sistemas de Distribución Trifásicos con Carga Variable en el Tiempo Utilizando el Método de Monte Carlo.* Universitat de Barcelona, 21-31

[7] Iberdrola Distribución Eléctrica (n.d.) Retrieved from http://iberdroladistribucionelectrica.com/

[8] Jolliffe, I.T. (2002) *Principal Component Analysis.* University of Aberdeen. Springer Series in Statistics, 10-59

[9] LaCàN, UPC. (2012). *Simulating and optimizing electrical grids: problem statement and potential applications of ROM and PGD to Smart Grids.* Universitat Politècnica de Catalunya, 1-10

[10] Martínez, J. and Guerra, L.G. *Optimum Placement of Distributed Generation in Three-Phase Distribution Systems with Time Varying Load Using a Monte Carlo Approach.* Power and Energy Society General Meeting, 2012 IEEE, 1-7.

[11] Nebraska Public Power District (n.d.) Retrieved from http://nppd.com/