Analysis of the impact of voltage control in distribution networks over energy losses

Author: Arturo Pérez Adroher

Supervisor: Pablo Frías Marín

Final Project

Bacherlor in Electromechanical Engineering

ICAI School of Engineering

26 of August 2016

Abstract

The voltage control is an essential service that the Distribution System Operator (DSO) and the Transmission System Operator (TSO) have to provide. Optimizing this ancialliary service is an important goal for both of them since the quality of the power supply is required. Moreover, the voltage level at wich a device is connected affects to the good functionality of it, and, also to the grid that will vary the current depending on the voltage level. Therefore is important to take into consideration the reaction that certain types of loads have with a variation of voltage. In this project it is analysed the repercution that a varation of voltage have over energy losses depending on the types of loads connected to the distribution network.

The main objective about this project is do a deep analysis of the impact of voltage control in different distribution networks using different load profiles over energy losses. For achieving the main one, other the objectives are needed. These are having a good mathematical model of the different types of loads that are connected to a certain grid, by means of laboratory tests. Usually, in power flow problems the loads are modelled as constant power consumption or constant impedance. However, in this project, an accurate mathematical model is used. This model, the ZIP model, anaylse the reactive and active power consumption of each load at different voltage levels. This is done because the loads consumption varies a lot depending on the load and usually they differ from the constant power /impedance model.

Also a simulation tool must be implemented in order to study the reaction that this load profiles have when they face different levels of voltage. This simulation tool

is going to be built in Matlab [1]. Besides, an important objective of this project is thinking about different load profiles for urban, semi-urban rural and industrial grids.

As it has been explained before, different load profiles are going to be simulated. The simulations are going to show their reaction to different levels of voltage. Solving the power flow problem of each network, the energy losses in the lines are easily calculated. The simulator tool uses the ladder power flow solution. This method that is deeply explained in the project consists in a series of iterations that implement the power flow problem. The iterations are done by means adjusting the voltage level each consumption point till it reach the allowed error. For the purpose of this project the allowed error is always going to be ten percent of the nominal voltage (usually 2.3 V).

The simulations are runned comparing the reaction that different levels of voltage have over energy losses. In the same place are ploted the reaction assuming the constant power consumption model, the constant impedance model and the ZIP model (mathematical model). The results show that for all types of grids the contant power model barely adjust to reality. However the Constant impedance model when lighnitng, heaters, electronic devices and low consumption induction motors have big weight in the load profile represents better their reaction. Nevertheless, it doesnt adjust completely and when there is a heavier presence of inductions motors the constant impedance model barely represents the rality when lower levels of voltage are considered. When considering different times of the year (winter and

---

[1] the programmation code is attached to the project as Appendix B

summer) there is a difference in the consumption. It has been observed that when in urban, rural and semiurban grids the presence of heaters in winter provoke more losses for lower voltages. However, the Air conditioned adjust more to the Constant Impedance Model. In an overall view in winter the nominal consumption level increases.

Improvements to the voltage control can be done. If the loads in distribution networks are modeled as ZIP loads, a better control of the energy losses can take place because the real reaction of the loads facing a variable voltage is known. If a constant model is taken, it is reccomendable to use the Constant Impendance Model as it adjust more to the reality than Constant Power model. Also few correction factors can be used for low voltage levels if a Constant Impedance Model is used.

Resumen

El control de voltaje es un servicio esencial que las empresas encargadas de la distribución y transmision eléctrica tienen que proporcionar. La optimizacion de ese servicio es un objetivo importante dado que influye directamente en la calidad de conexión que se proporciona al consumidor. Este nivel de voltaje, como es lógico, inlfuye a su vez en el buen funcionamiento de los aparators conectados a la red. Existen diferentes leyes que regulan este tema, y en ellas se establecen rangos legales en los que los niveles de voltaje se deben de encontrar. Por estos motivos es importante considerar a su vez las reacciones que ciertos tipos de cargas tienen cuando se les somete a diferentes niveles de voltaje. En este proyecto va a ser analizada la repercusión que las variaciones de voltaje en redes de distribución eléctrica tienen en las perdidas energéticas, considerando a su vez los perfiles de carga que estan conectados

El objetivo principal de este proyecto se va centrar en realizar un profundo analisis del impacto que el control de voltaje tiene en las perdidas energéticas de las redes de distribucion, usando diferentes tipos de perfiles de carga. Para alcanzar este objetivo, previos objetivos tienen que ser obtenidos. Estos objetivos son, entre otros, construir un buen modelo matemático de los differentes tipos de aparatos conectados a la red. Estos modelos matemáticos son obtenidos en el laboratorio de maquinas eléctricas de la Escuela Superior de Ingeniería de ICAI de la Universidad Pontifica de Comillas. Normalmente, en los problemas de flujos de carga las cargas son modeladas o como cargas que consumen una potencia activa constante o como cargas que poseen una impedancia constante. Sin embargo, en este proyecto, un afinado modelo

matemático es usado. Este modelo matemático, es el llamado modelo ZIP. Este modelo analiza la consumición de potencia activa y reactiva de una carga variando el nivel de voltaje. e El resultado es unas ecuaciones que representan exactamente la reacción de estas cargas en los diferentes tipos de niveles. Este modelo es usado porque usualmente lo que consume una carga varia mucho dependiendo de al nivel de voltaje que se este sometiendo la misma, y por lo tanto, un modelo que asuma una reacción fija puede no ser el mas acertado.

Para el estudio de esta reacción mencionada anteriormente es necesario crear una herramienta de simulación. Esta herramienta se contruye con el programa Matlab [2], y permite calcular las perdidas energéticas en lineas de distribución para distintos niveles de voltaje. Esta herramienta también permite seleccionar el tipo de red que se esta simulando, las cargas que hay en dicha red y la estación del año en la que se esta simulando. Esto es de suma importancia, pues otro de los objetivos es construir distribuciones de carga típicas, diferentes una de la otra, para redes urbanas, rurales, semiurbanas e industriales.

Como se ha explicado se han simulado differentes perfiles de cargas para diversas redes para distintos niveles de voltaje. Solucionando el problema de flujo de carga mediante el metodo *ladder power flow*, se consigue calcular las intensidades de la red y por ende las perdidas energéticas. Este método que se utiliza para resolver el problema de flujo de carga consiste en realizar una serie de iteraciones que ajustan el valor de la tensión de cada punto de consumo hasta que se alcanza un error permitido. Este error permitido es el 1 por ciento de el valor de la tensión nominal

---

[2] el codigo de programación esta adjuntado al projecto como Appendix B

(normalmente 2.3 V)

Los diferentes escenarios se simulan como se ha dicho antes para diferentes niveles de voltaje. En la misma grafica se pintan las perdidas originadas en la linea consierando que las cargas actuan siguiendo un modelo de potencia activa consumida constante, impedancia constante o ZIP (modelo matematico de las cargas). Los resultados muestran que para todos los tipos de redes el modelo de potencia activa consumida constante no representa en absoluto la verdadera reaccion que tienen las cargas. En cambio, el modelo de impedancia constante se ajusta mas a la realidad. Aun mas cuando la presencia de aparatos como calefactores, luminarias y maquinas de induccion de pequeño consumo. Aun asi, no se ajusta el todo, sobre todo cuando hay una presencia alta de motores de induccion de alto consumo. Por ejemplo en redes industriales, que tipicamente estan caracterizadas por tener connectadas maquinas muy grandes, el modelo de impedancia constante no se ajustaria, sobre todo, como se explica en el pryecto, para bajos niveles de voltaje. Cuando se consideran diferentes epocas del año (invierno y verano) el nivel de perdidas cambia debido tambien al cambio en presencia de calefactores por aires acondicionados, o viceversa. También, se observa que para redes parcialmente o totalmente residenciales se incrementan las perdidas energéticas en invierno.

Logicamente, despues de este proyecto se pueden plantear algunas tecnicas de optimización de reducción de perdidas. Por ejemplo, si se modelan las cargas con un modelo ZIP la reacción de las cargas se va a saber a la perfección. Una solución posible sería asumir un modelo de impedancia constante y añadir unos factores de correción para bajos niveles de tensión de redes industriales.

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Arturo Perez Adroher DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Analysis of the Impact of Voltage Control in Distribution Networks over energy Losses, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor CEDE a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.

- Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- Asignar por defecto a estos trabajos una licencia Creative Commons.
- Asignar por defecto a estos trabajos un HANDLE (URL persistente).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- Que la Universidad identifique claramente su nombre como autor de la misma
- Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- Solicitar la retirada de la obra del repositorio por causa justificada.
- Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 21 de Agosto de 2016,

***ACEPTA*: Arturo Perez Adroher**

Firmado:

A Mimi,
que su ejemplo de superacion y alegre esfuerzo ha sido inspirador

A mi padre, que sin su insistencia e interés no hubiera llegado al punto en el que estoy

A mi madre, por su cariño y apoyo

Contents

List of Figures

List of Tables

1 Introduction

1.1 State of art

Nowadays the DSO bussiness have many methods to reduce the transmission losses in the grid. Some of the most used are: the substitution of thecurrent material of the lines for another materials with less resistance, the reducction of fluxes of reactive power or increasing the nominal voltage in certain points of the grid. The last one, however, as it has been explained before is not totally true. For some load profiles raising the voltage only leads to a increase of the energy losses in the line. The reason for this is that certain types of loads change their impedance depending on the voltage level at wich they are connected, therefore, sometimes an increase of voltage lead to a increase of the current.

This problem has been studied already in diverse project of different universities. Papers like *Measurement-based Estimation of the Composite Load Model Parameters by Byoung-Ho Kim and Hongrae Kim* [Kim and Kim, 2012] begin to work in the modelation of load profiles and propose a new aproach to the study of the electrical grid. This new way focusses not only on the study of the transmission and distribution, but also in the impact that the load profile has in the grid.
One of the options is the ZIP model for loads. ZIP stands for constant impedance, current and active power. This model is quite old and it is not very effective when a lot of induction generators/motors are conected to the grid. Because of this, it is necesary also to do a study of the behaviour of the variation of the consumption of this machines when the voltage level at wich they are feeded changes. This study is done in this project. However, also there are variations of the ZIP model like the polinomic ZIP model, the exponential ZIP model or a combinaiton of both. The polinomic ZIP model

can be expressed as:

$$P_{ZIP} = P_0 \cdot \left(a_{Pzip} \cdot \left(\frac{V}{V_0} \right)^2 + b_{Pzip} \cdot \left(\frac{V}{V_0} \right) + c_{Pzip} \right) \tag{1}$$

From this equation it can be concluded that the consumption of active power depends of the voltage level at wich the device is connected. The parameters a,b and c are relevant for the study of the load profile because they determine wich type of load we have and how sensible is to voltage changes. In this project is going to be modeled the active power and also the current:

$$I_{ZIP} = I_0 \cdot \left(a_{Izip} \cdot \left(\frac{V}{V_0} \right)^2 + b_{Izip} \cdot \left(\frac{V}{V_0} \right) + c_{Izip} \right) \tag{2}$$

With this model it can be known the impact that the changes on the voltage level have in the grid. This is because, if the current that a device needs to consume in order to work at a certain voltage level point is known, the energy losses that this voltage change is making too.

Nevertheless the ZIP model doesn't fit always good when we have disturbances in the grid. This can happens frequently when we have air conditioning devides and induction motors and generators. This matter has been studied before in the paper *Load Modelling Transmission Research by the Lawrence Berkeley National Laboratory* [Bravo et al., 2010]. It proposes a new model called WCEE that fits better with reality. But for the demosntration that is needed in this project the ZIP model is equally usefull than the WCEE model.

As it is seen, the modelling of load profiles have been deeply studied, as well as the behaviour that they have with differents levels of voltage. However, there are some papers like *Modeling power system load using intelligent methods by Shengyang He*[He, 2011] y *Evaluation of Conservation Voltage Reduction (CVR) on a National Level by Schneider, Fuller,Tuffner and Singh*[Schneider et al., 2010] that goes a step further than the modelling and make a most deeply variable voltage study. This deep voltage study is necessary if it is needed to go further in this study, because it measures the modeling in different cases. It has been demonstrated that either the voltage control and the reactive power control are essential for the grid, and moreover they are very related to each other. This also the case of the active power and the phase angle.

It is understood then that the voltage level in the grid have to be in between certain values to provide a good function to all the devices. The voltage control is done by means of the control of the fluxes of reactive and active power that travel trhough the lines.

In general, the reactive power flows are controlled by capacitor banks adjustable transformers. Both of them have the main purpose of controlling the reactive current flow in different parts of the grid, reducing at the same time the voltage drop. The capacitor banks usually are able to modify their capacity in order to work at different levels to obtain a certain voltage requiered. Specially, in Spain, the adjustable bank capacitors are more used due to its simplicity and cost.

Nevertheless, not only controlling the reactive power flow it is achieved a perfect voltage control. It is necessary also taking into account the active power flows in order to achieve a reduction of te energy losses, and so, the voltage drop in the grid. An effective method for the active power flow control its the Distributed Generation (DG). It basically consists on the injection of active power on certain node of the grid from a independent generation. With this method, through the main grid there is not a lot of active power flow, and this also reduces the energy losses and the voltage drop. Thesis like *Generación Distribuida: Aspectos técnicos y su tratamiento regulatorio de Victor Méndez*[Méndez Quezada, 2005] study the impact of different levels of percentage of DG in the grid. Also projects like *Evaluación del impacto de generación distribuida en la operación y planificación de las redes de distribución eléctrica de Alberto Martín García*[García, 2006] consider the energy losses over the percentage of DG in a grid:

$$Penetration = \frac{Power_{DG}}{Power_{feeder}} \tag{3}$$

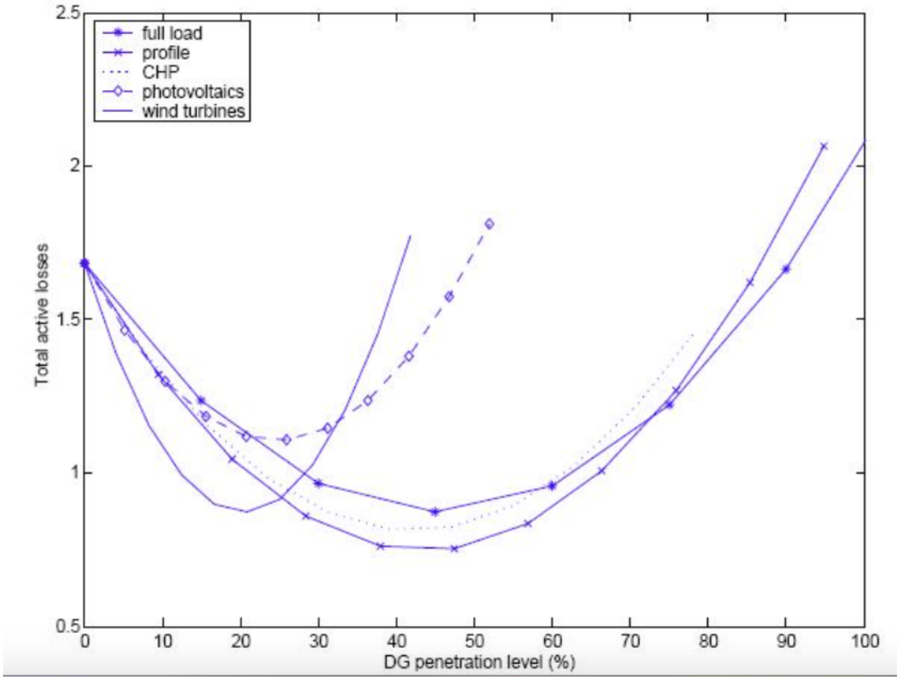


Figure 1: Penetration curve

[García, 2006]

It is demonstrated that for low levels of penetration the enrgy losses decrease and for high levels of penetration the losses incerase.

The voltage control explained before aims to achieve what is called: Optimal Power Flow. The goal is to reach for all types of scenarios a optimal power flow that lead to a saving of energy and money to the DSO.

This project using as a point of departure the ZIP model of the loads will alalyse the impact of voltage control in distribution networks over energy losses, and how, a certain voltage control can reduce them.

1.2 Objectives

The main objectives of this project are:

- **Main Objective**: Deep analysis of the impact of voltage control in different distribution networks using different load profiles over energy losses
- **Good mathematical model of the loads**: by means of measurements in the laboratory, different devices such as lamps, pc's, screens, fridges, machines and others are going to be tested at differents levels of voltage to see how they react, and wich is the current, active and reactive power that they require in those different situations.
- **Study the different types of electrical grids**: understanding the difference between the diverse types of load profiles in the electrical grids the demonstration will be more complete.
- **Build a simulation tool**: one of the main objectives of this project is building a programmable tool able to simulate the consume of a specific type of grid with an specific type of load profile.

1.3 Methodology

The resolution that is going to be taken in order to solve the problem explained before is:

- Measurement of different types of loads in the laboratory of Electrical Machines of ICAI School of Engineering

- Mathematical modelling of the loads consumption with different voltage levels. It has been measured the following devices: lamp, discharge lamp, induction motor, heater, resistor, Screen and PC.
- Study of the urban, semiurban and rural grids.
- By means of a programmable tool it is going to be simulated these different grids.
- Taken into account the results, the solution to the problem will be formulated.

  The methodology that is going to be followed to achieve a conclusion of the problem that concerns to this project is deeply explained in the section *Methodology.* After the modelling of the loads and the definition of the scenarios comes the simulation. The simulation tool has been written considering the ladder power flow method.

1.4 Structure

The project is going to follow this structure:

- Energy losses in electric power systems: This chapter will define the energy losses, how they ocur, what methods are utilized for avoiding them and how are they calculated.
- Methodology. In this chapter power flows in networks are explained as well as the way for solving them. Also, it is deeply explained the method used for this project and why
- Modeling of loads. In this chapter different devices are modeled in order to test them after in the simulations.

- Simulation and Results. In this chaprter the different scenarios are presented and simulated. The results are presented and explained
- Conclusions. After obtaining the results the conclusions of the project are explained

2 Energy losses in electric power systems

2.1 Introduction

Energy losses in electric power systems are one of the main issues to deal with. Losses always ocur, and optimizing methods for reducing them are in constant development . This is also very important since they supose an economic loss for the TSOs and DSOs. Although losses can't dissapear, several methods have been developed for reducing them. Some of this methods are explained in this section.

2.2 Modelling Energy Losses

2.2.1 Definition and Classification

An important concern for the Tranmission System Operator (TSO) and Distribution System Operator Business is the energy losses ocurring in the grids that they control. This suposse a big cost in transmission and distribution systems. Normally in european distribution networks the losses are between 2-15 percent. In Spain the distribution losses are 6 percent of the power supplied. The transmission losses are about 2-3 percent.
The energy losses in networks can be classified in to groups:

- Technical Losses: are caused by the materials of the grid (Cu), transformers, interconnections, non-optimized power-flows, non-ptimized generation

- Non-Technical Losses. This kinds of losses ocur when a fraud in the electrical measurements takes place or to the existance of ilegal connection to the grid.

The control, monitoring and optimization of both losses is one of the main issues to solve in power systems.

2.2.2 Energy losses in Transmission and Distribution Systems

In the power system, as it has been explained before, energy losses is an important matter to take into account. However, in each country the percentage of losses is different. This is due to the different characteristics of each power system. The amount of losses can increase or decrease depending on various factors as the efficiency of the network, transformers and quality of the generation.

Typicaly the percentage of losses in distribution systems is around 5-6 percent while in transmission systems is about 2-3 percent. However, this values differ in each country. In Spain, for example the energy losses of distribution lines are of 8 percent, while in Denmark are 6 percent and in the Netherlands 3,8 percent.

In Spain, the cost of kWh in 2006 was about 0.182 €, one of the highest of Europe. Taking into account that the Total Energy Losses (Transmission, Transformation and Distribution) are 7 percent of the Generation and that the Consmption in 2006 was 252.600 GWh, the amount of losses were about 19.012 GWh. This suposses a cost of 346.034,838 e.

2.3 Theory of Energy Losses and variables that affect or modify its value

As it was explained before, in the power system there are some defined methods to reduce the energy losses of a specific network. Within these methods there can be found the following ones:

- Increase the efficiency of the transformers.
- Change the material of the cables for one that has lower resistance

- Increase the nominal voltage of the grid.
- Voltage / Reactive power control

Mostly, all of the methods used for reducing the energy losses in a grid are related to three main characteristics / components of the network:

- **The material of which the cables of the network are made of**. It is well known that the energy losses in a cable are defined as:

  $$S_{Losses \quad 2\to 1} = U_{2\to 1} \cdot I_{12}^{*} \tag{4}$$

  $$S_{Losses \quad 2\to 1} = U_{2\to 1} \cdot I_{12}^{*} = P_{Losses \quad 2\to 1} + jQ_{Losses \quad 2\to 1} \tag{5}$$

  $$S_{Losses \quad 2\to 1} = U_{2\to 1} \cdot I_{12}^{*} = Z_{line} \cdot I_{12} \cdot I_{12}^{*} = P_{Losses \quad 2\to 1} + jQ_{Losses \quad 2\to 1} \tag{6}$$

  The energy losses that are more relevant to the DSO's and TSO's business are the active Power Losses (W). These enery losses can be deduced from the last equation:

  $$P_{Losses \quad 2\to 1} = Re(Z_{line(21)}) \cdot |I_{21}|^2 = R_{12} \cdot |I_{21}|^2 \tag{7}$$

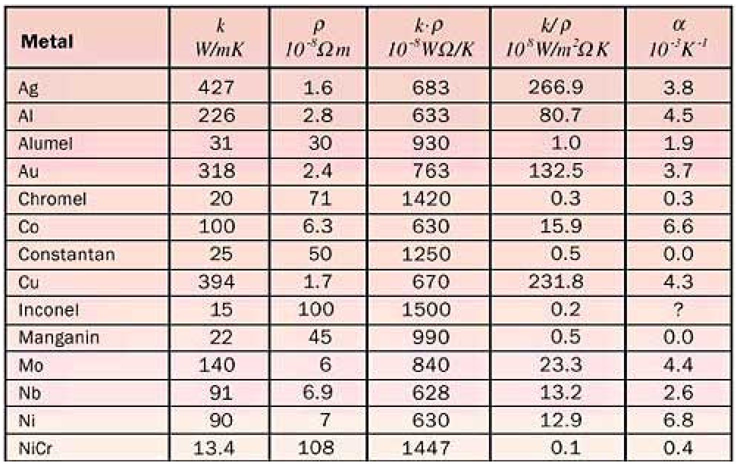
  If three phases are considered, as it has to be, the total losses will be:

  $$P_{Losses \quad 2\to 1} = 3 \cdot Re(Z_{line(21)}) \cdot |I_{21}|^2 = 3 \cdot R_{12} \cdot |I_{21}|^2 \tag{8}$$

  The equation above shows that the active power energy losses depend on the resistance and the current through it. The resistance of a cable is a value that depends on the material, the lenght and the properties of this material. It can be calculated like this:

  $$R_{cable} = \rho_{material} \cdot \frac{\ell_{cable}}{S_{cable}} \tag{9}$$

  The many materials used for cables have different resistivities:



Metal	k W/mK	ρ $10^{-8}\Omega m$	$k\cdot\rho$ $10^{-8}W\Omega/K$	k/ρ $10^{8}W/m^{2}\Omega K$	α $10^{-3}K^{-1}$
Ag	427	1.6	683	266.9	3.8
Al	226	2.8	633	80.7	4.5
Alumel	31	30	930	1.0	1.9
Au	318	2.4	763	132.5	3.7
Chromel	20	71	1420	0.3	0.3
Co	100	6.3	630	15.9	6.6
Constantan	25	50	1250	0.5	0.0
Cu	394	1.7	670	231.8	4.3
Inconel	15	100	1500	0.2	?
Manganin	22	45	990	0.5	0.0
Mo	140	6	840	23.3	4.4
Nb	91	6.9	628	13.2	2.6
Ni	90	7	630	12.9	6.8
NiCr	13.4	108	1447	0.1	0.4

Figure 2: Electrical Resistivities of various metals. Source: http://www.electronics-cooling.com

It important to mention that also the type of grid is relevant for the value of the resistance of the line. Generally, rural grids are longer than urban grids. This is logical, because, the lenght between points of consumption of both grids is largely different. In an urban grid, the distance between two points of consumption, (if we are talking about a residential area for example) can be about 2-3m. However, in a rural grid the distance between two consumption points can be 5 km if we are talking about two farms in the countryside. This is why in this project four types of grids are going to be considered:

- Rural Grid
- Semi - Urban Grid
- Urban Grid
- Industrial Grid

- **Voltage Level**: The voltage level affects to the energy losses. This can be deduced from the above definition of energy losses. The voltage level at which the loads are connected affects direclty to the current that they are demanding. There are two matters concerning the voltage level:

  - The nominal voltage level of the line: 132 kV, 230 kV, 400kV..etc.
  - The exact level at which the consumption point is. Normally, it ocurs that the voltage level is between 90-110 percent of the nominal voltage. There is a regulation about this. The IEC 50-160 states that the voltage level must be in between these two percentages. This is 0.9Un - 1.1Un.

- **Load Profile**: A load profile is a graph in which the consumption of an area (industrial, commercial...) is measured by hours [Figure 3]. This curve represent the agregate demand in Spain. It is important to consider that it represent what it ocurs individually in each grid. If the agregation level is low (few demand curves toguether), the changes in tha agregate demand are going to be higher.

  There are many things that affect the load profile. The time of the year or the type of network (urban, rural...). The characteristics of the loads connected to the network affect directly to the energy losses. Lets put a basic example. Imagine that two induction machines are connected to the grid (230V). Both of them have characteristics. One is consuming 3000W (cosphi=0.8) and the other 2000W (cosphi=0.9). The current demand of the first one is going to be higher as it can be deduced. As higher the current demand the energy losses in the transmission lines as well. If this is extrapolated further, it can be said that a residential area with high demand is going to provoke more losses than a simple grid of shops with low consumption level. Because of this when evaluating the losses of a grid refering to the load profile, some aspects have to be taken into account. These are:

- Type of grid: urban, rural, industrial, semi-urban. It can be founded out that certain types of grids such as rural grids between big house-farms are more likely to provoke higher losses. This is because each consumer is demanding more than a current house of a city.
- Hour of the day. More likely, in the morning and in the evenings, there are going to be peaks of consumption due to the working-studying timetables.

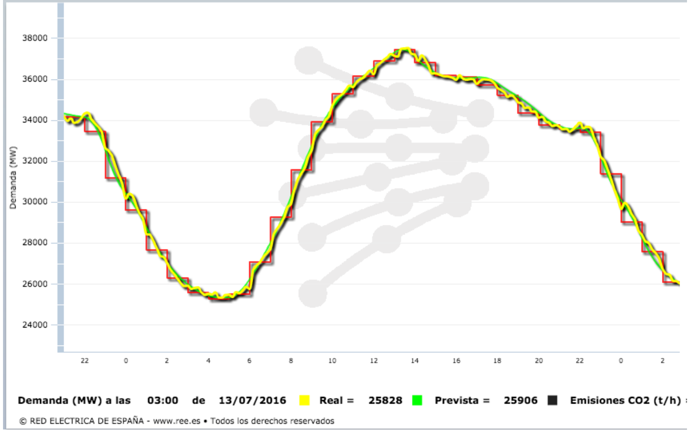


Figure 3: Spanish demand on the 12 of July 2016

- Time of the year. Depending on the time of the year the load profile is going to be different. This is basically because two factors:
    - In Winter, the hours of light are more or less in certain countries, so, more lightbulbs are going to be connected longer/shorter to the grid.
    - Depending on the time of the year more devices such as Heaters, Air Conditioning and other temperature devices are going to be used. Oftenly, these devices represent a very important part of the consumption of the installation.

For this purpose, in this project, the four types of grids above described are going to be simulated in two different times of the year:

- Spring-Summer: With an increase in the presence of Air Conditioned and fans in the network.
- Fall-Winter: With and increase in the presence of Heaters (Big and Small) in the network.

2.4 Conclusions

As it has been seen there are several methods for avoiding energy losses. Although they can't dissapear, they can be reduced. Losses depend in the characteristics of the Trasnmission, Distribution, Generation and, also, in the Consumption. The load profile influence a lot in the energy losses. This is going to be deeply studied in section 5: Modeling of loads. In this chapter the behaviour of the loads is going to be studied, and using it as a point of departure, further conclusions about how the loads influence the energy losses is going to be achieved.

3 Methodology

After doing this overview of the relevant variables that affect and modify the level of energy losses in a grid the resolution of the load flow problem used in this project in going to be described.

3.1 Power Flow in AC

In the project many networks are going to be faced. This networks operate in alternate current. In this networks there are different points of consumption that consume different values of reactive and active power. Also, each point of consumption is at a different voltage level and trough each point there is a different current.

All these values must be calculated in order to know *what* is happening in the network. This is what is called power flow solution. Once all these values are calculated, further things can be known, as the energy losses for example.

3.2 Power Flow in a distribution network

Before it has been explained what a power flow is. This power flow is going to be studied in distribution networks. A distribution network is the one that supply directly to the consumers. In the electric power systems, it exist four main steps between the generation and consumption.

- Generation: Nuclear plants, Wind farms.. etc.
- Transmission: This is the transport of power between the generation and the distribution. In each country, tipically it exists a TSO (Transmission system operator) that is in charge of this. In Spain it is called Red Electrica Española, in the Netherlands Tennet...

- Distribution: The distribution of the power takes place at a lower voltage level between the transmission and consumption. As with the case of the TSOs, there are companies in charge of the distribution. In Spain Iberdrola is one of the Business in charge of this. The distribution networks can vary a lot from one to other. Also there are different types (urban, semiurban, rural, industrial...).
- Consumption: Houses, industries, rural areas. and more.

3.3 Solving power flows

There are many methods to solve power flows. Some of them are Newton-Raphson, radial method, simplified radial method [mendez2005generacion] and ladder flow method. For this project the ladder flow method is going to be used [Liu et al., 2002]. This is an iterative method that is going to be implemented in order to solve the power flow problem for a certain grid. It is chosen because its simplicity and application to this project.

3.4 Ladder Power Flow: An Iterative Resolution Method

The resolution method is going to be explained step by step. The maximum iterative error for this simulations is going to be 0.1Un. A grid can be represented as:

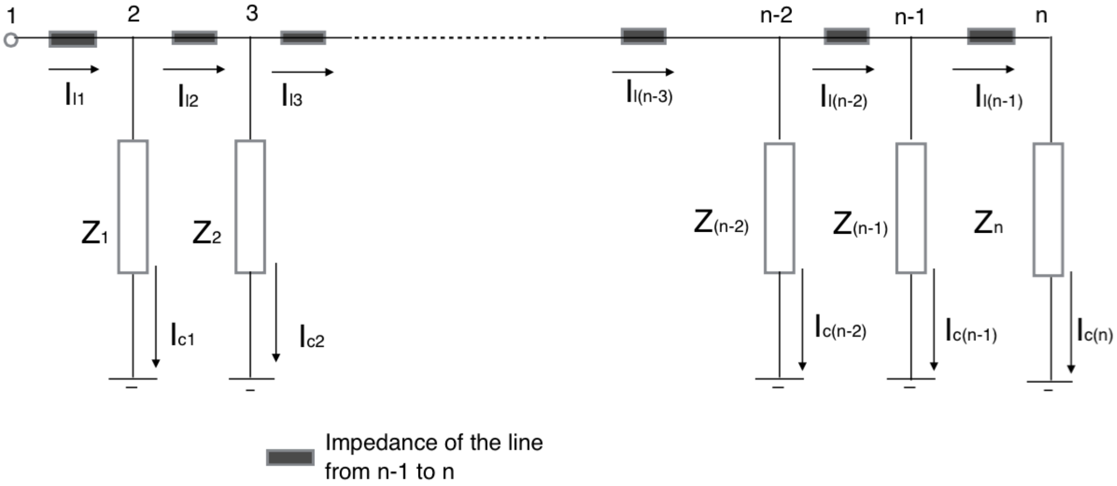


Figure 4: Representation of a grid

Being:

$$U_1, U_2, ..., U_N \quad : \quad Voltage \quad in \quad the \quad consumption \quad points \tag{10}$$

$$I_{l1}, I_{l2}, ..., I_{lN} \quad : \quad Current \quad trough \quad the \quad line \tag{11}$$

$$I_{c1}, I_{c2}, ..., I_{cN} \quad : \quad Current \quad trough \quad the \quad loads \tag{12}$$

$$Z_{l1}, Z_{l2}, ..., Z_{lN} \quad : \quad Impedance \quad of \quad the \quad line \tag{13}$$

$$Z_1, Z_2, ..., Z_N \quad : \quad Impedance \quad of \quad the \quad loads \tag{14}$$

The voltage level of all the consumption points is going to be the voltage level of the control point:

$$\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ ... \\ U_{n-2} \\ U_{n-1} \\ U_n \end{pmatrix} = \begin{pmatrix} U_1 \\ U_1 \\ U_1 \\ ... \\ U_1 \\ U_1 \\ U_1 \end{pmatrix} \tag{15}$$

Then the current through the loads is calculated:

$$\begin{pmatrix} I_{c1} \\ I_{c2} \\ I_{c3} \\ ... \\ I_{c(n-2)} \\ I_{c(n-1)} \\ I_{c(n)} \end{pmatrix} = \begin{pmatrix} Z_1 & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & Z_2 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & Z_3 & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & Z_{n-2} & 0 & 0 \\ 0 & 0 & 0 & ... & 0 & Z_{n-1} & 0 \\ 0 & 0 & 0 & ... & 0 & 0 & Z_n \end{pmatrix}^{-1} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ ... \\ U_{n-2} \\ U_{n-1} \\ U_n \end{pmatrix} \tag{16}$$

However, in the first iteration:

$$\begin{pmatrix} I_{c1} \\ I_{c2} \\ I_{c3} \\ ... \\ I_{c(n-2)} \\ I_{c(n-1)} \\ I_{c(n)} \end{pmatrix} = \begin{pmatrix} Z_1 & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & Z_2 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & Z_3 & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & Z_{n-2} & 0 & 0 \\ 0 & 0 & 0 & ... & 0 & Z_{n-1} & 0 \\ 0 & 0 & 0 & ... & 0 & 0 & Z_n \end{pmatrix}^{-1} \begin{pmatrix} U_1 \\ U_1 \\ U_1 \\ ... \\ U_1 \\ U_1 \\ U_1 \end{pmatrix} \tag{17}$$

Once obtained the currents through the loads, the current through the line can be easlily calculated by Kirchoff's law:

$$\begin{pmatrix} I_{l1} \\ I_{l2} \\ I_{l3} \\ ... \\ I_{l(n-3)} \\ I_{l(n-2)} \\ I_{(n-1)} \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} I_{ck} \\ \sum_{k=2}^{n} I_{ck} \\ \sum_{k=3}^{n} I_{ck} \\ ... \\ I_{c(n)} + I_{c(n-1)} + I_{c(n-2)} \\ I_{c(n)} + I_{c(n-1)} \\ I_{c(n)} \end{pmatrix} \tag{18}$$

After calculating the current through the line the new voltage levels of the consumption points can be obtained as:

$$U_2' = U_1 - I_{l1} \cdot Z_{l1} \tag{19}$$

Using this value of the voltage of point 2:

$$U_3' = U_2' - I_{l2} \cdot Z_{l2} \tag{20}$$

Using this value of the voltage of point 3:

$$U_4' = U_3' - I_{l3} \cdot Z_{l3} \tag{21}$$

$$. \; . \; .$$

$$U_{n-2}' = U_{n-3}' - I_{l(n-3)} \cdot Z_{l(n-3)} \tag{22}$$

$$. \; . \; .$$

$$U_n' = U_{n-1}' - I_{l(n-1)} \cdot Z_{l(n-1)} \tag{23}$$

These new values of the module of the voltage level of the consumption points are going to be compared to the previous one. These values are going to be stored in a vector called Mismatch:

$$\begin{pmatrix} M_2 \\ M_3 \\ M_4 \\ ... \\ M_{n-2} \\ M_{n-1} \\ M_n \end{pmatrix} = \left| \begin{pmatrix} U_2 \\ U_3 \\ U_4 \\ ... \\ U_{n-2} \\ U_{n-1} \\ U_n \end{pmatrix} - \begin{pmatrix} U'_2 \\ U'_3 \\ U'_4 \\ ... \\ U'_{n-2} \\ U'_{n-1} \\ U'_n \end{pmatrix} \right| \tag{24}$$

If any component of this vector is superior to the maximum allowed error (in the case of this project 0.1 Ui) another iteration has to be made. The new iteration would use the new calculated values of the voltage level of the consumption points as default values of the voltage level of the consumption points. However, if the error is lower than the maximum allowed error, the calculation is finished and the voltage level values are the last ones calculated.

Once obtained all the parameters of the grid the Energy Losses can be easily calculated, as it was explained in the section: "Theory of Energy Losses and variables that affect or modify its value".

$$S_{Total \quad Losses} = P_{Total \quad Losses} + j \cdot Q_{Total \quad Losses} \tag{25}$$

$$S_{Total \quad Losses} = \sum_{k=1}^{n-1} S_{lk_{Losses}} = \sum_{k=1}^{n-1} U_{k \to k+1} \cdot (I_{lk})^* \tag{26}$$

Applying Ohm's Law:

$$S_{Total \quad Losses} = \sum_{k=1}^{n-1} S_{lk_{Losses}} = \sum_{k=1}^{n-1} I_{lk} \cdot Z_{lk} \cdot (I_{lk})^* \tag{27}$$

Considering 3 phases:

$$S_{Total \quad Losses} = 3 \cdot \sum_{k=1}^{n-1} S_{lk_{Losses}} = 3 \cdot \sum_{k=1}^{n-1} I_{lk} \cdot Z_{lk} \cdot (I_{lk})^* \tag{28}$$

And finally from the equations above it can be deduced:

$$P_{Total \quad Losses} = \sum_{k=1}^{n-1} 3 \cdot |I_{lk}|^2 \cdot Re(Z_{lk}) = \sum_{k=1}^{n-1} 3 \cdot |I_{lk}|^2 \cdot R_k \tag{29}$$

3.4.1 Variations of the Iterative Resolution Method for this project

The iterative resolution method above explained is the one that is going to be used for solving all the power flows of all types of grids. However, some aspects have to be explained. Different types of grids are going to be simulated so the impedance of the line (Zl) is going to change depending on the grid that it is being simulated.

Also, each grid is going to be simulated considering:

- Constant Impedance Model (Z cte). This model mantains the impedance of the load constant facing variations in the voltage and the current.
- Constant Power Model. This model mantains the power of the load constant facing variations in the voltage and the current.
- ZIP Model. This model varies the power consumed by the load depending on the voltage level point at which the load is. The variation of the power follows:

$$P_{ZIP} = P_0 \cdot \left(a_{Pzip} \cdot \left(\frac{V}{V_0}\right)^2 + b_{Pzip} \cdot \left(\frac{V}{V_0}\right) + c_{Pzip} \right) \tag{30}$$

$$Q_{ZIP} = Q_0 \cdot \left(a_{Qzip} \cdot \left(\frac{V}{V_0}\right)^2 + b_{Qzip} \cdot \left(\frac{V}{V_0}\right) + c_{Qzip} \right) \tag{31}$$

The parameters of the ZIP model (a,b and c) for both the active and reactive power, are going to be obtained by a series of experimental tests in the laboratory. This is called **"Modelling of loads"** and it is explained in the following chapter.

3.5 Conclusion

As it has been demonstrated this method is good for solving power flow problems in AC. Also is the best one if we are going to work also with a ZIP model of the loads. This is because the a *ZIP load* changes its value for different levels of voltage, and an iterative method is the best because it keeps adjusting its value modifying it each time an iteration is done. This makes that the simulator doesn't get blocked because it is a long but easy calculation.

4 Modeling of loads

4.1 Introduction

An electrical grid is defined as a number elements interconnected wich have the main goal of transport the electricity from the generation to the consumption. Between this two points takes place the transmission of electricity, that trough a physical mean (copper, aluminium...) transport the electricity from one point to another. In this part of the process is where ocurrs the losses, caused by the materials wich the grid is made off or by transmission defects such as short circuits. One of the main objectives for the transmission system operator is reduce as much as possible the quantity of losses, because it is traduced in a save of money and an increment in the efficiency of the grid. Losses depend in the characteristics of the trasmission and distribution lines, generating plants and consumers .In this chapter the behaviour of the loads is going to be studied, and using it as a point of departure, further conclusions about how the loads influence the energy losses is going to be achieved.

4.2 Loads and their models

- Constant Impedance Model (Z cte). This model mantains the impedance of the load constant facing variations in the voltage and the current. Therefore if we assume that the impedance is constant, if the voltage is raised, the current demanded by the load will be proportionally higher. This can be deduced from:

  $$U_{consumption \quad point} = Z_{load} \quad (cte) \cdot I_{load} \tag{32}$$

  And if the voltage increases, the current will increase also. This fact is important to consider because, as it has been explained in previous chapters, the current demanded by the loads affect directly to the losses of the distribution and transmission lines. The reaction facing a variable voltage will be:

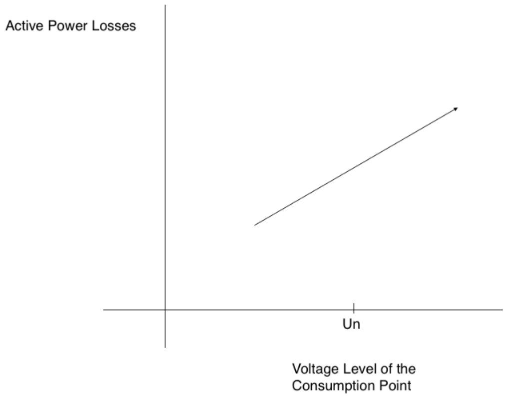


Figure 5: Energy Losses of the line supplying a Constant Impedance Model Load facing variable voltage

- Constant Power Model. This model mantains the power of the load constant facing variations in the voltage and the current. However, the reaction of this model is the opposite to the previous one. If the voltage increases, the current demanded by the load will decrease as well. this can be deduced from the following equation:

  $$P(cte) = U_{consumption \quad point} \cdot I_{load} \tag{33}$$

  This makes that the current demanded by the load will decerase when rasing the voltage. This can be seen in the following graph:

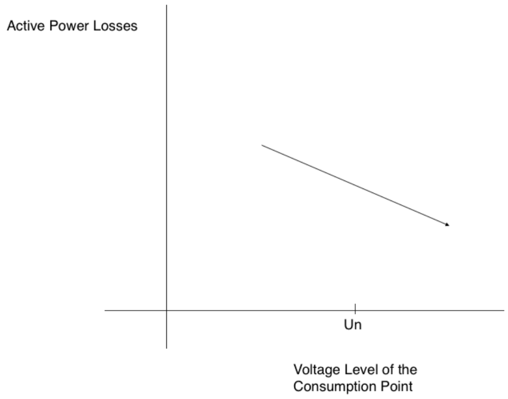


Figure 6: Energy Losses of the line supplying a Constant Power Model Load facing variable voltage

4.2.1 Alternative model: ZIP Model

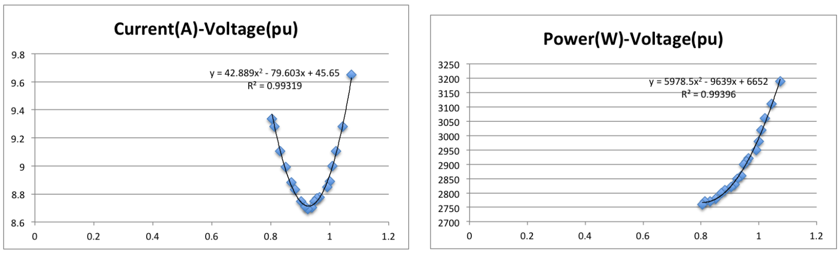
There are two load models that are usually used in power flows. These have been explained before. They are:

- Constant Power Model
- Constant Impedance Model

Nowadays, in the electricy world there is an statment (that has been introduced before) that says that for minimazing the losses is necesary to rise the voltage in certain points of consumption of the grid. This is like this because it is assumed that the active and reactive power remains constant with a variable voltage. And so, if the power is constant when we rise the voltage the current will decrease. This can be deducted from the following equation:

$$\overline{S} = \overline{I^*} \cdot \overline{U} \tag{34}$$

But this statment is not completely true. There are some existing load profiles that when the voltage increases, the current also and this leads to an increase of the losses in the line that feeds the load, and differs from the constant impedance model because the reaction is not linear as it is supposed. For example, and induction motor has the following voltage-current, voltage-power curve:



(a) Current Consumption (b) Power Consumtion

Figure 7: Test of a induction motor at variable voltage

This is because the current, active and reactive power don't remain constant at a variable voltage in this device (neither in others). Also, they don't react linearly to a variable voltage. In this load the power consumption change when we change also the voltage, therefore the current doesn't decrease when the voltage become higher. This causes the opposite objetive it is wanted when trying to reduce the losses.

This project will focuss in the different load profiles and their behaviour with a variable voltage in urban, semi-urban and rural grids of different european countries. The spanish electrical grids will be studied in detail.

For the correct study of the load curves in urban, semi-urban and rural grids it is necesary to study the curves that define the current/power consumption versus the voltage. The load curve in a specific house or industry is a sum of all the load curves of the equipments functioning at the time. For this purpose, it has been tested in a laboratory tipical equipments that operate in this kind of situations: a regular 500 W lamp, a high intensity discharge lamp, an inductor machine, two kind of PC screens, a PC, a 500 W resistor, a heater and a fridge. It is going to be studied the load curves versus a variation in the input voltage, making it vary between +/- 10 percent of its nominal value with a Variac and measuring it with a HAMER. The nominal voltage is 230 V, and the nominal frequency is 50 Hz.

The aim of these essays is model different loads in order to build an equation that gives us the power consumption for a given input voltage. For model this loads it is normally used two basic methods. The aim of these essays is model different loads in order to build an equation that gives us the power consumption for a given input voltage. This method is called Static Model wich consists in constant impedance, constant current and constant power; is also called the ZIP model. Every load then can be considered as:

$$P \quad (W) \quad = \quad P_0 \quad \cdot \quad \left(a \cdot \left(\frac{\triangle V}{V_N} \right)^2 \quad + \quad b \cdot \left(\frac{\triangle V}{V_N} \right) \quad + \quad c \right) \tag{35}$$

$$Q \quad (var) \quad = \quad Q_0 \quad \cdot \quad \left(a \cdot \left(\frac{\triangle V}{V_N} \right)^2 \quad + \quad b \cdot \left(\frac{\triangle V}{V_N} \right) \quad + \quad c \right) \tag{36}$$

$$I \quad (A) \quad = \quad I_0 \quad \cdot \quad \left(a \cdot \left(\frac{\triangle V}{V_N} \right)^2 \quad + \quad b \cdot \left(\frac{\triangle V}{V_N} \right) \quad + \quad c \right) \tag{37}$$

Being Po the nominal power, Io the nominal current, and a,b and c characteristical parameters of each load.

Using the least square adjustment for the measurement of each equipment it is obtained the polinomial expression showed before. The equipments we are going to use are showed in this table:

Equipment	**Characteristics**
Lamp	500 W
Discharge Lamp	250 W
Induction Motor	220 V 2.5/5/10 A
Heater	1/2.5 KW
Resistor	500 W
PC Screen (LG)	59 W

Table 1: Equipment tested

4.3 Results of the measurements and conclusions

As it has been shown in the measurements that are included in Appendix A, certain types of loads change their consumption at a variable voltage. The results reflect that the common saying that the power or the impedance of a load remains constant with a variable voltage is false. Therefore the statement of raising the voltage to reduce the losses is not true in some cases and it is necessary to create a control system that raise or decrease the voltage depending on the type of load profile in order to avoid high losses.

In order to do a complete study of all the ZIP models of different loads, it has been researched in previous projects like"Experimental Determination of the ZIP Coefficients for Modern Residential, Commercial, and Industrial Loads" other ZIP loads. In this paper a Refrigerator, washing machine, clothes dryier, VCR, TV and a fan are measured withing other things and the results are very interesting for our purpose. In the following table it is shown the parameters of each device. This parameters indicate the sensitivity of each load.

Power	**Active Power (W)**			**Reactive Power (var)**		
Coeficients	**a**	**b**	**c**	**a**	**b**	**c**
Lamp	334,96	150,18	45	656,999	-1086,6	492,82
Discharge Lamp	313,9	-6,7874	-32,94	440,05	-624,47	301,01
Heater 1kW	1387,4	-494,93	233,43	1142,4	-1877	811,02
Heater 2.5kW	3401,4	-1196	579,19	727,54	-1082,7	453,75
Induction mot 1	3260,2	-5052,1	2885,4	20969	-31960	13049
Induction mot 2	3511,5	-5642	4002,5	21066	-33039	13961
Induction mot 3	5978,5	-9639	6652	15603	-24002	10494
Resistor	641,83	-91,238	30,63	0	0	0
PC	-7,8886	13,653	29,124	14,185	-7,5609	36,69

Table 2: ZIP model of loads

4.3.1 Conclusions

After testing all the components, it has been reached the conclusion that netiher of them can be modelled as constant impedance or constant power consumption. It is tru that some of them are approach more one model, for example, the light bulbs and heaters, are more close to a constant impedance model. However, even these ones are not exactly following this model. Others like induction motors and PCs don't follow any of the models. Therefore, it is going to be interesting putting all of them toghether and study the reaction facing a variable voltage.

5 Modeling of Scenarios

5.1 Introduction

Till now, different mathematical ZIP equations have been obtained. The loads are ready to be simulated in a larger scenario. For this purpose different scenarios have to be built. In this chapter is going to be explained how different scenarios are proposed, and what loads are included and i which combination

5.2 Building Scenarios

5.2.1 Introduction

Now different types of grids of different countries are going to be simulated. The data of the impedance of each distribution grid is known and the normal power consumption of it also. Knowing the typical power consumtion of each point in the grid. A load profile is going to be constructed in this point. For example:

We can have that the point [1,6] (being 1 the begining of the distribution grid and 6 the 6th consumption point) is a flat of 120 m2 consuming in normal conditions 5000 W + 3000 i VAR.

Then a load profile for this point is going to be done taking into account the amount of active power consumed.

$$3000W \quad = \quad P_{fridge} + P_{heater} + P_{light \quad bulbs} + ... + P_{dryer} + P_{Television} + P_{laptop} \tag{38}$$

We can say then that this house is the sum of individual loads of certain devices whose nominal power in total sum the nominal house power consumption. As it is known how this certain devices react to different levels of voltage because we have measured them, it can be simulated.

5.2.2 Modelling of the consumption points

Taking a look to the consumption points of a grid, each of them havea different consumption. Is for this reason that a differentiation between them depending on the consumtion level is needed. Taking into account the *"Reglamento Electrotécnico de Baja Tensión"* we can clasify the different loads.
For Industries with different levels of consumption:

Active Power Consumption Level	**Industry Type**	**Spring-Summer**	**Fall-Winter**
4000-7000 W	Industry with low consumption level (A)	5 Discharge Lights 1 Air conditioned 2 Induction Motors 230 V / 2.5 A	5 Discharge Lights 1 Small Heater 2 Induction Motors 230 V / 2.5 A
7000-10000W	Industry with medium consumption level (B)	7 Discharge Lights 2 Air conditioned 3 Induction Motors 230 V / 5 A	7 Discharge Lights 3 Induction Motors 230 V / 5 A 1 Big heater
10000-15000W	Industry with high consumption level (C)	7 Discharge Lights 4 Air conditioned 2 Induction Motors 230 V / 5 A 1 Induction Motor 230 V / 10 A	7 Discharge Lights 1 Big Heater 2 Induction Motors 230 V / 5 A 1 Induction Motor 230 V / 10 A

Table 3: Classification of Inudstries by consumption level

For Ilumination of the street or buiding, that is installed in certain point of it:

- 800-1000 W : 4 Discharge Lamps (Same for Winter and Spring)

- 3500-4500 W : 16 Discharge Lamps (Same for Winter and Spring)
- 7500-8500 W : 32 Discharge Lamps (Same for Winter and Spring)

For houses and residential areas these assumtions are going to be taken:

Active Power Consumption Level	**House Type**	**Spring-Summer**	**Fall-Winter**
4000-7000 W	House with low consumption level (A)	5 Discharge Lights 2 Normal Bulbs Dryer / Washing machine PC 2 Air conditioned Television Fridge Fan	5 Discharge Lights 2 Normal Bulbs Dryer / Washing machine PC 1 Heater Television Fridge
7000-10000W	House with medium consumption level (B)	10 Discharge Lights 4 Normal Bulbs Dryer Washing machine 2 PC 2 Air conditioned 2 Television 2 Fridge 2 Fans	10 Discharge Lights 4 Normal Bulbs Dryer Washing machine 2 PC 2 Big heater 2 Television 2 Fridge 1 Small heater
10000-15000W	House with high consumption level (C)	12 Discharge Lights 6 Normal Bulbs Dryer Washing machine 2 Fans 2 PC 4 Air conditioned 2 Television 2 Fridge	12 Discharge Lights 6 Normal Bulbs Dryer Washing machine 2 Big heater 2 PC 1 Small heater 2 Television 2 Fridge

Table 4: Classification of houses by consumption level

Once defined the different types of loads that could be in different points, the grid is going to be simulated considering the loads:

- Constant Power (the one nowadays assumed),
- Constant Impedance (also assumed nowdays)
- ZIP load model (real life consumption reaction of the loads facing a variable voltage)

5.2.3 Urban, Semi-Urban, Industrial and Rural Grids, Definition

In distribution systems, various types of grids are usually distinguished. These are:

- **Urban**: Mostly residential area with some local shops and few industries.
- **Semi-urban**: Residential area with heavy presence of industry
- **Rural**: Residential Area in a rural surrounding
- **Industrial**: A network where in its majority is made of indsutries.

Normally each of these grids have a different line/cable impedance, this is taking into account when studying the energy losses of the grid.

5.2.4 Scenarios

Different energy losses from rural, semi urban and urban european grids are going to be studied and simulated in this project. the scenarios that are considered are the following

Type	**Location**	**Consumption Points**	**Description of Consumption**
Rural	France	13	6 residential consumers type A 4 residential consumers type B 2 residential consumers type C 1 lightning point 1000W
Urban	Spain	8	2 residential consumers type A 2 residential consumers type B 3 residential consumers type C 1 lightning point 1000W
Semi-Urban	Spain	8	3 residential consumers type A 3 industrial consumers type A 2 lightning points of 1000 W
Industrial	Spain	8	5 industrial consumers type C 1 industrial consumer type A 1 industrial consumer type B 1 lightning point of 4000 W

Table 5: Scenarios

6 Simulations and Results

6.1 Introduction

The scenarios that are going to be simulated have been presented in the last chapter. With a simulation tool that has been programmed and is attached to the project as *Appendix B: Simulation Tool* the power flow problem of each network will be solved. This simulation tool is based on the ladder power flow solution [Liu et al., 2002]. The input and output data of this simulation tool is:

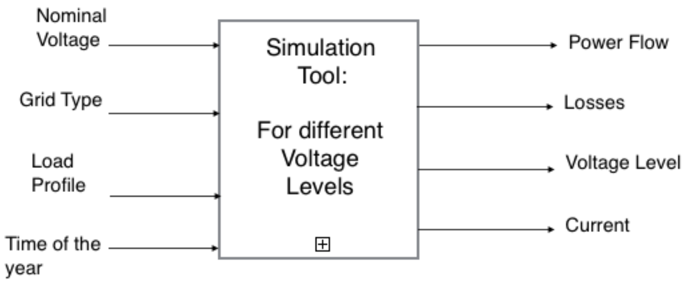


Figure 8: Input / Output Simulation Tool

6.2 Simulation of a French Rural Grid

A rural French grid is going to be simulated. This grid has 13 points of consumption. Only one of them is used for lighting and 12 are residential consumers.

However, these 12 consumers have a different load profile. As explained before, they have been classified in three: Residential Consumers type A, B and C. As a result there are:

- 6 consumers type A
- 4 consumers type B

- 2 consumers type C
- 1 lighting point of 1000W

The result of the simulations are:

Losses (W)						
	Time of the Year					
Unitary Value		90% Un	95% Un	100% Un	105% Un	110% Un
Voltage Level (V)		207 V	218 V	230 V	242 V	253 V
ZIP Model	Summer	1232	1265	1325	1408	1515
	Winter	1662	1773	1907	2064	2243
Constant P Model	Summer	2345,3	2043,8	1800,6	1557,9	1402,7
	Winter	3511	3050,2	2680,8	2379	2128,3
Constant Z Model	Summer	1042,5	1161,5	1287	1418,9	1557,3
	Winter	1540,1	1716	1901,4	2045,4	2244,9

Table 6: Results of the simulation of a french rural grid

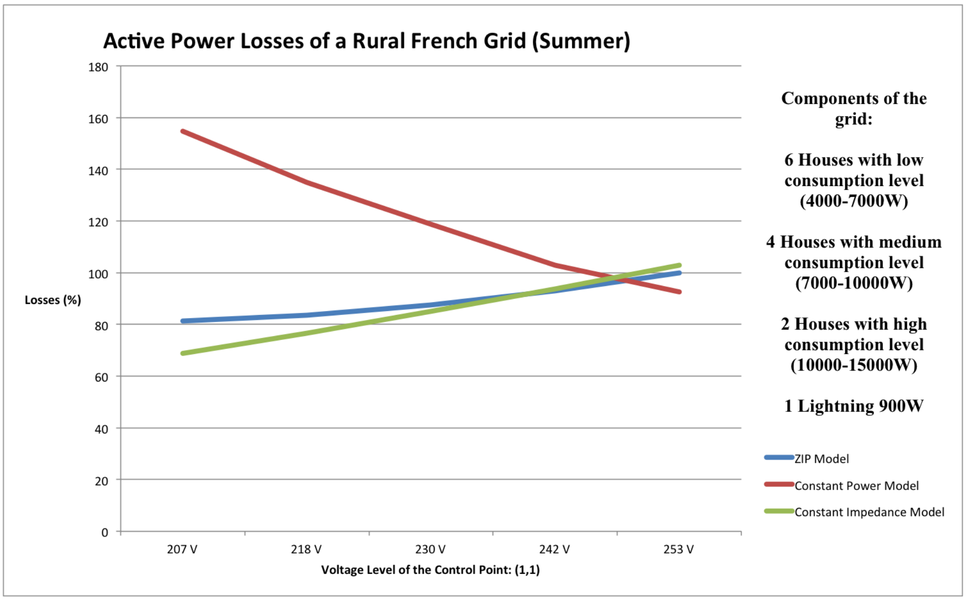


Figure 9: Results of the simulation of a french rural grid summer

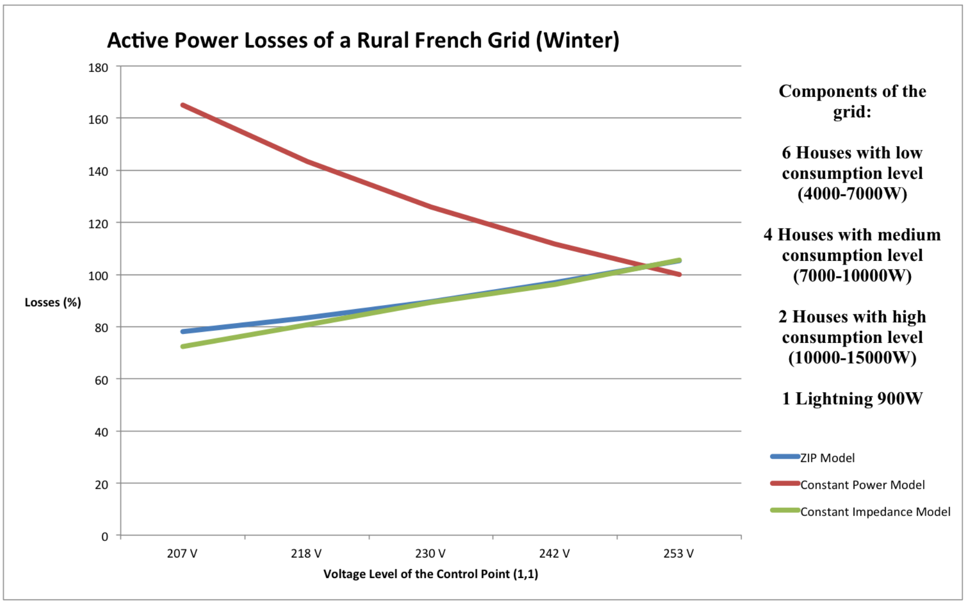


Figure 10: Results of the simulation of a french rural grid winter

As we can see in the results in winter the Active Power losses, in percentage is higher than in summer. Also, in winter the aproximation of the ZIP model to the constant impedance model is higher. this is due to the presence of heaters instead of Air conditioned in the grid. Heaters tend to have a constant impedance. This can be deduced from Firgure 27 and 28. However we can see a shift for low voltages when the test in carried out in summer. This is because the presence of Air conditioned. This devide increase the needed current for low voltages, like the induction motors. This type of reaction will be seen in the next sections, when comparing summer and winter. Assuming that the losses using the ZIP model are the real losses of the grid, the Constant Power Consumption model is far away from the reality. The Constant Impedance model would expect lower losses than the real ones.

The tendency of the Constant Power Model is always with a negative slope and it is almost linear. For low voltages it forecast almost 160 percent of the nominal losses. In this case, for 90 percent of the nominal voltage, the constant power model forecast losses of 2345 Watts while ZIP Model forecast 1042 Watts. In contrast the tendency of the Constant Impedance Model in this case and also of the ZIP Model is always with a positive slope. The reason of this have been explained before. The reason why there is a shift between the ZIP model and the Z cte model for low voltages is because the existance of induction motors (big, medium or small) in the grid. In this case we are simulating a rural grid which we assume with 12 houses. This houses have washing machines, air conditioned and fridges. These devices make the losses increase when there is an undervoltage. Nevertheless, the presence of this devices in a residential area is very low and thus, in the agregate demand the slope becomes negative. This has a deeper explanation. Lets put an example. The graph that represents the washing machine is:

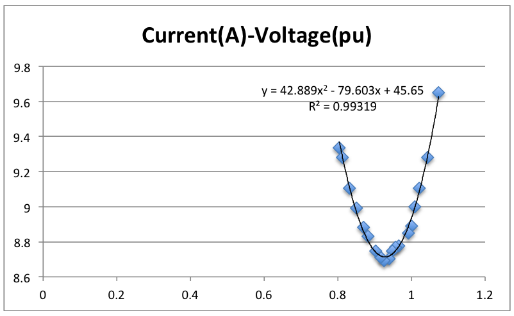


Figure 11: Induction Motor Current for different voltage levels

If this graph is combinated with this one, that is the case of a light bulb:

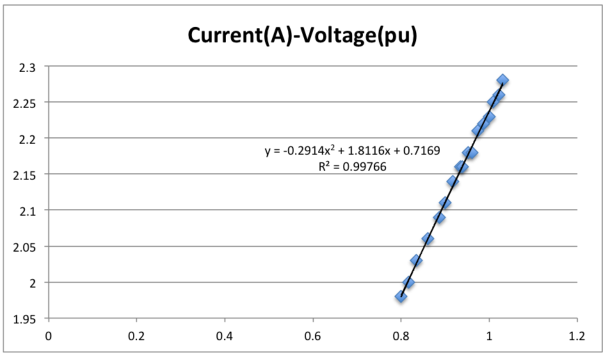


Figure 12: Light Bulb Current for different voltage levels

The resulting curve will be modified. If we take into account that the presence of the "induction motor load type" (washing machines, dryer, fridge, air conditioned) is about 20 percent of the consumption of the house. the reuslting curve is no longer going to be like a parable. Instead is going to be like a line with a slow deviation for low voltages. Of course, as higher the presence of induction motors the line will quicker change its slope and finally if we reach more that a 50 percent it will convert to a parable.

It can be seen that there is a point where all the tendency curves reach the same value. This point is also the nominal power losses of the distribution network. This value ocurs when all of the loads have approximately the nominal voltage 230. Then the Constant Impendance Model,consant power model and ZIP model have the same value of impedance and power consumed. It can be easily explained with the next equations.

Consant Power Model

The Active Power Consumed will be:

$$P(cte) \quad = \quad U_{consumption \quad point} \cdot I_{load} = \quad P_N \tag{39}$$

The Impedance of the loads will be:

$$Z_{load} = \frac{U_{consumption \quad point}}{I_{load}} = \frac{U_N}{I_{load}} = \frac{U_N \cdot U_N}{P_{consumption \quad point}} = \frac{U_N^2}{P_N} = Z_N \tag{40}$$

Consant Impedance Model

The Impedance of the loads will be:

$$Z_{load} = Z_N \tag{41}$$

The Active Power Consumed will be:

$$P_{consumed} = U_{consumption \quad point} \cdot I_{load} = U_N \cdot \frac{U_N}{Z_{load}} = \frac{U_N^2}{Z_N} = P_N \tag{42}$$

ZIP Model

In the ZIP Model the active an reactive power tested at the nominal voltage always consume the nominal active power and reactive power. This can be seen in *Appendix A: Measurements at a variable voltage.* Therefore it can be affirmed that with the ZIP Model, in nominal conditions, the load consume the nominal active and reactive power.

Conclusions

The tendencies then have been explained. The constant power model will always follow the negative slope tendency while the constant impedance will do the opposite thing. Nevertheless the ZIP model, more realistic than both of them, will get always more close to the Z cte model but wil have a shift for low voltages. this shift will be biger or not depending on the presence of induction motor load types. These conclussions can be applied to all of the Simulations that are going to be done. Probably on the Urban spanish grid and SemiUrban, the case in going to be the same but a little bit more pronounced how much is biger the presence of indutries that are characteristic for having more percentage of induciton motor load type.

6.3 Simulation of a Spanish Urban Grid

An Urban Spanish grid is going to be simulated. This grid has 8 points of consumption. Only one of them is used for lighting and 7 are residential consumers.

However, these 7 consumers have a different load profile. As explained before, they have been classified in three: Residential Consumers type A, B and C. As a result there are:

- 2 consumers type A
- 2 consumers type B
- 3 consumers type C
- 1 lighting point of 1000W

The result of the simulations are:

Losses							
	Time of the Year						
Unitary Value		90% Un	95% Un	100% Un	105% Un	110% Un	115%Un
Voltage Level		207 V	218 V	230 V	242 V	253 V	265V
ZIP Model	Summer	5690	5747	5947	6276	6734	7326
	Winter	7985	8386	8919	9574	10350	11248
Constant P Model	Summer	10670	9321	8223	7315	6556	5912
	Winter	20340	16453	14398	12722	11336	10173
Constant Z Model	Summer	4479	4991	5530	6097	6691	7313
	Winter	6893	7680	8509	9382	10297	11254

Table 7: Results of the simulation of a Spanish urban grid

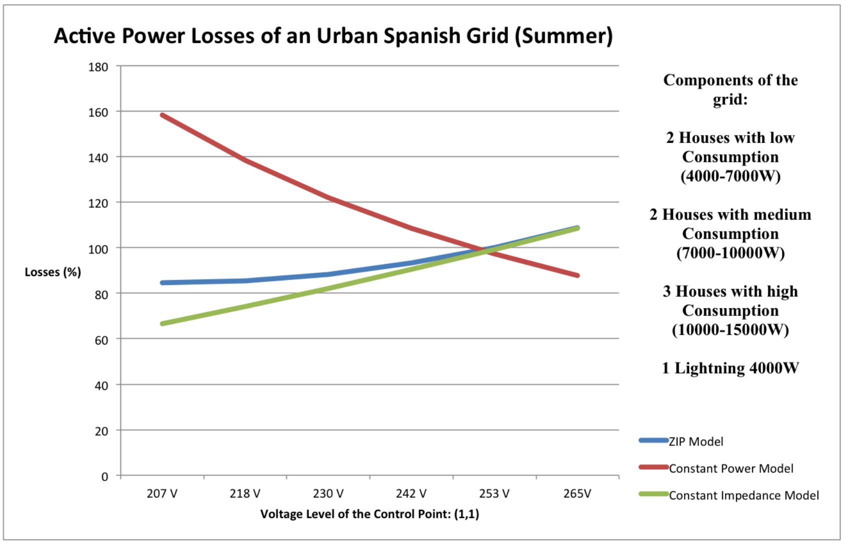


Figure 13: Results of the simulation of a Spanish urban grid summer

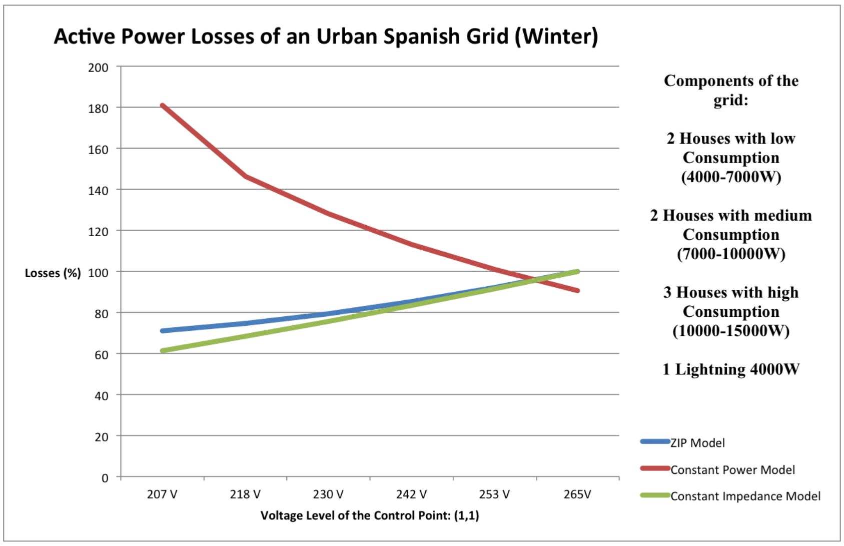


Figure 14: Results of the simulation of a Spanish urban grid winter

Similar results are obtained for an Spanish Urban grid. Again, the Constant Power Consumption model is far away from the reality. Also, the "summer shift" is obtained because of the Air Conditioned. However in this urban grid, because of the consumers type, the percentage is 20 percent higher in the Constant Power Consumption Model. This is due to the heavy loads that are being considered.

In this case, the percentage of Losses is more less the same as in the rural grid. The behaviour is almost exactly the same. This is because even with this Urban Spanish grid the load types are more less the same. However there is something that must be said in this case. The percentage of Losses in the Urban network is slightly smaller. This is due to the characteristics of the urban and rural distribution lines. Usually the Urban distribution lines are much sshorter than the Rural Distribution lines. In a Rural scenario we can face that a farm is 10 km away from the other while in a city one house can be 2 or 3 meters away from the other. Because of this, rural lines are going to have a larger impedance and are going to provoke higher losses.

However, the behaviour facing a variable voltage between the rural and the urban network is the same. And this is, as it has been exlpained before because of the load type, that are all houses.

The point where all the tendency curves join is more less in the same level of voltage. The explanation gaven before also justifies in this case.

6.4 Simulation of a Spanish Semi-Urban Grid

An Urban Spanish grid is going to be simulated. This grid has 8 points of consumption. Only one of them is used for lighting and 6 are residential consumers.

However, these 6 consumers have a different load profile. As explained before, they have been classified in three: Residential Consumers type A, B and C. As a result there are:

- 3 residential consumers type A
- 3 industrial consumers type A
- 2 lighting points of 1000W

The result of the simulations are:

Losses						
	Time of the Year					
Unitary Value		90% Un	95% Un	100% Un	105% Un	110% Un
Voltage Level		207 V	218 V	230 V	242 V	253 V
ZIP Model	Summer	1064,1	1164,4	1327,6	1561,4	1875
	Winter	1107	1233	1418	1668	1993
Constant P Model	Summer	2077	1845	1650	1485	1344
	Winter	2262	2007	1794	1613	1460
Constant Z Model	Summer	1136	1266	1403	1547	1697
	Winter	1221	1360	1507	1662	1824

Table 8: Results of the simulation of a Spanish semi-urban grid

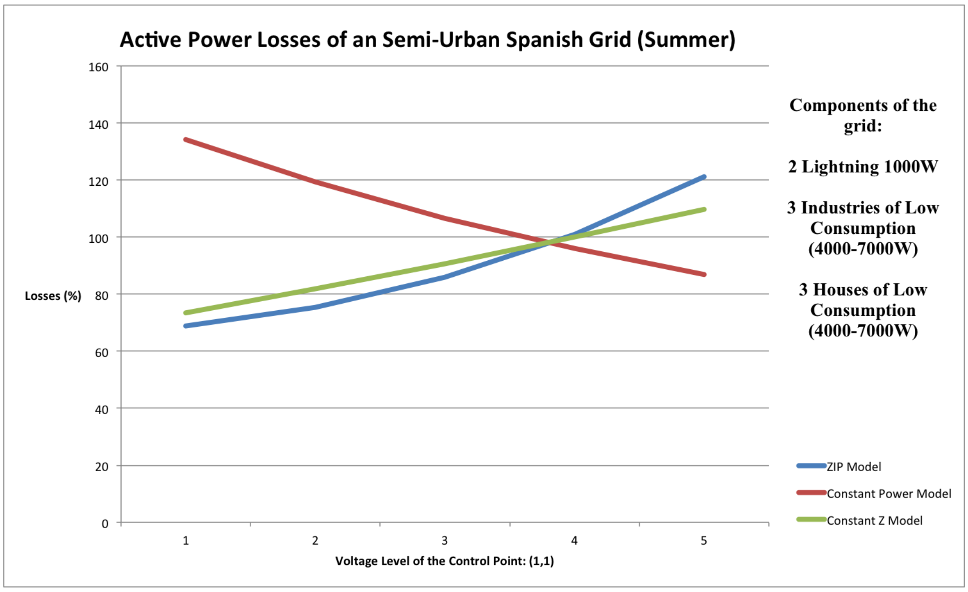


Figure 15: Results of the simulation of a Spanish semi-urban grid summer

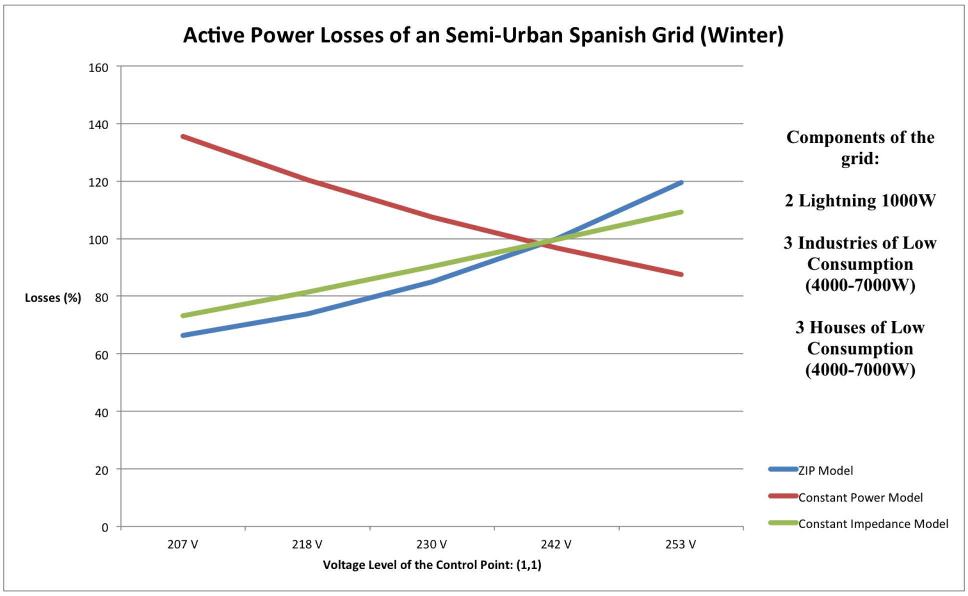


Figure 16: Results of the simulation of a Spanish semi-urban grid winter

The results of the simulations for a semi urban Spanish network are very similar to the urban. However, the presence of induction motors make that the shift appear in both winter and summer simulations. In this case, however, the expected losses of the constant impedance model are higher.

In this case, it can be hardly made the difference between winter and summer. This is due to the presence of induction motors in the grid now. also the induction motors that are working are very small and they almost approach to the constant impedance model. However, for overvoltages the ZIP model forecast higher losses than the Z cte model this is because the loads that are connected for overvoltages are not exactly linear. Instead for overvoltages they have aproximately 1.4 times slope than the Z cte model. Because of this it can be predicted that the reaction for overvoltages is going to be more agressive.

6.5 Simulation of a Spanish Industrial Grid

An Urban Spanish grid is going to be simulated. This grid has 8 points of consumption. Only one of them is used for lighting and 6 are residential consumers.

However, these 6 consumers have a different load profile. As explained before, they have been classified in three: Residential Consumers type A, B and C. As a result there are:

- 5 industrial consumers type C
- 1 industrial consumer type A
- 1 industrial consumer type B
- 1 lighting points of 4000W

The result of the simulations are:

Losses							
	Time of the Year						
Unitary Value		90% Un	95% Un	100% Un	105% Un	110% Un	115%Un
Voltage Level		207 V	218 V	230 V	242 V	253 V	265V
ZIP Model	Summer	7536	7208	7227	7549	8165	9085
	Winter	9642	9541	9745	10211	10926	11896
Constant P Model	Summer	13335	11628	10242	9099	8144	7338
	Winter	21618	17389	15212	13439	11972	10742
Constant Z Model	Summer	5428	6048	6701	7388	8108	8862
	Winter	7251	8079	8952	9870	10833	11840

Table 9: Results of the simulation of a Spanish industrial grid

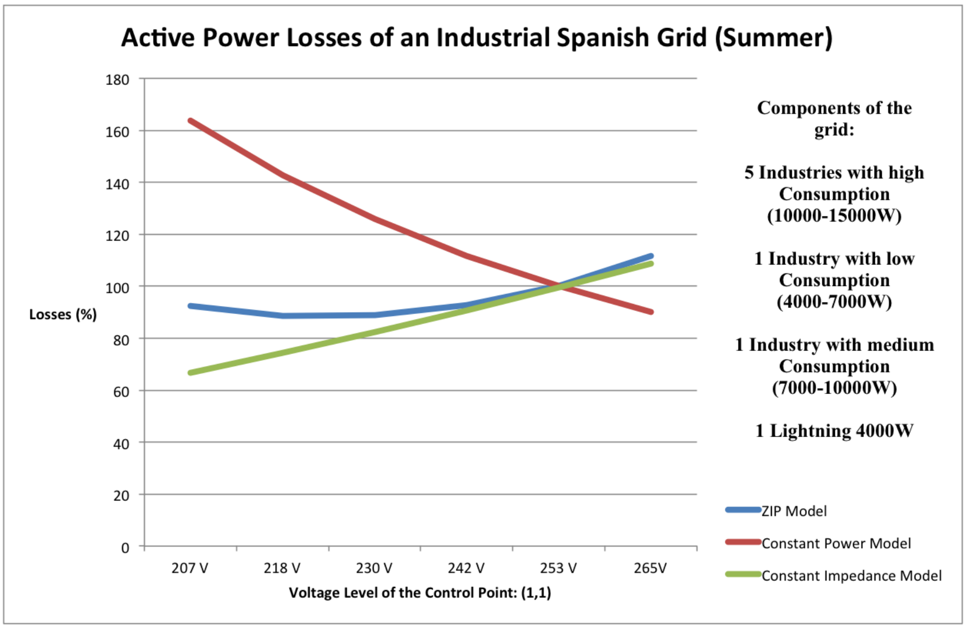


Figure 17: Results of the simulation of a Spanish industrial grid summer

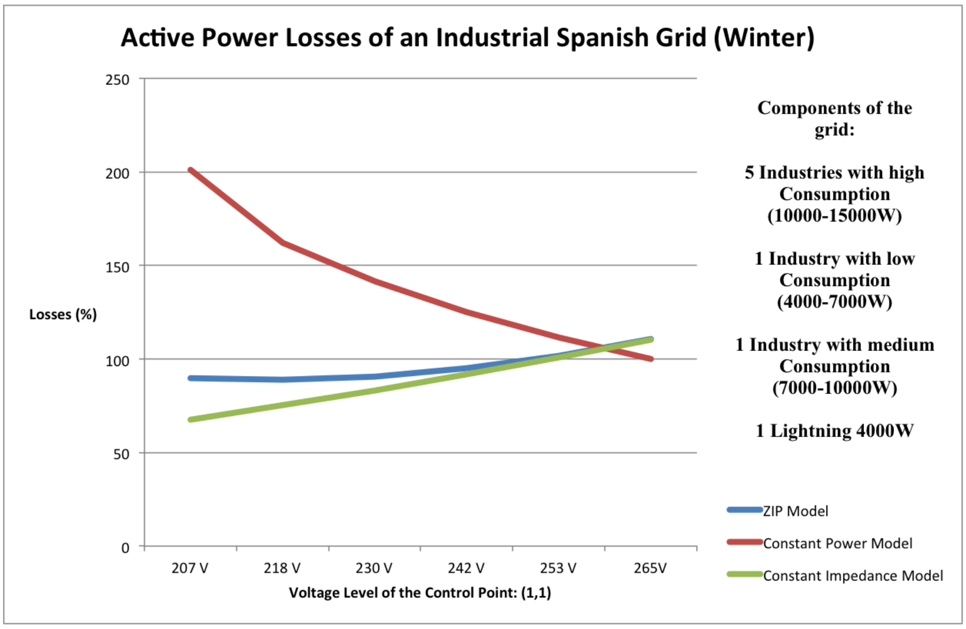


Figure 18: Results of the simulation of a Spanish industrial grid winter

The simulations of the Inudstrial network show something very interesting. As it can be apreciated in Figure 25, Figure 26 and Figure 27, the current demanded by the induction motors for lower voltages is higher when the voltage becomes lower than the nominal voltage also. In an industrial network, there is a heavy presence of induction motors. This makes that for lower voltages the losses increase. Like the induction motors, as it was said before, the Air Conditioned acts the same way. That is why the shift ins more pronounced in summer than in Winter. In this case the Consant impedance model is only applicable for voltages higher than the nominal voltage. For lower voltages the curve shifts completely apart from the Constant Impedance model.

The simulation of the industrial grid is the demonstration of how he induction motors make the model curve shift from Constant impedance model. A solution that can be applied to this problem is stuying the posibility of cominating a correction factor of presence of inudction motor load types to the constant impedance model. This is a good idea because simulating a very big network in which all the loads are a function of the voltage can be to heay for the simulator.

Also, we can observe that the percentage losses for the Industrial network are high for the constant power model This is due to the high nominal power that some of the induction motors have. This makes the current needed increase a lot when the voltage decrease. A curious thing happens with induction motors. In the undervoltage region, they approach more a constant power model, however, for overvoltage regions it approach a constant impedance model. It can be seen in this figure:

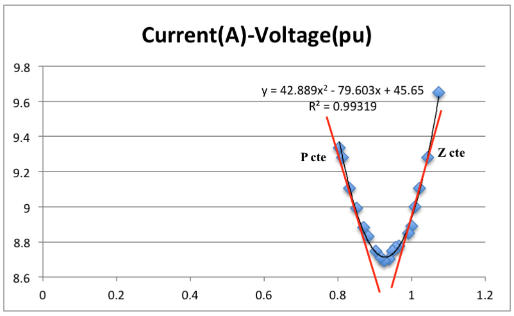


Figure 19: Induction Motor Current for different voltage levels, similarities with Z cte model and P cte model

6.6 Conclusions

The conclusions that can be carried out about the simulations are:

- The importance of the type of network regarding the behaviour of the energy losses depending on the voltage level
- The fact of how the induction motors can make the losses-voltage curve behave or shift in one way
- Differences in absolut value of the losses beween rural an urban grid

7 Replicability of the analysis and results in other EU countries

7.1 Introduction

In order to do a deep study of this matter the electrical grids of two different countries, appart from Spain, are going to be studied separately. Its is going to be explained in general aspects their characteristics and funcitonality, and then compare it to the spanish one, so and understanding of the generation trasnmission and distribution system can be achieved. Also, after explaining the main characteristic of the power system, the behaviour and the load profile of each different consumer is going to be presented

7.2 Denmark

7.2.1 Danish Power System, general aspects

The danish power system is divided in two regions, the western part, connected with Europe and eastern part, connected to Sweeden. Both regions are not synchronized an that is why they are joined by the *Great Belt Power Link (Storebælt HVDC)* a HVDC interconnection between Funen and Zealand. The direct current interconnection is needed because as it was said both regions are not synchronized. The eastern part is synchronized with Sweeden that is part the Nordic Pool Spot countries (Norway, Denmark, Sweden, Finland, Estonia, Latvia, Lithuania, Germany and the UK) as well as Denmark. The wester part, however is synchronized with Continental Europe. This characteristic make the danish power system very peculiar. Also, another singularity of the danish power system is the amount of Wind Power generated and consumed.

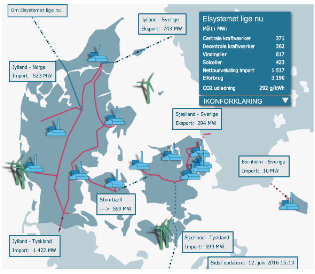


Figure 20: Source: http://energytransition.de/

The electricity market in Denmark relies on fossil, nuclear and renewables resources. Since 1970 Denmark has been investing a lot in wind power energy, being in 2009 the country with more wind power generation per person in the world. In 2015, 42,1 percent of the energy generated in this country was wind energy. This made the Danish Power system one of the countries with more security of supply of the world and more environmentally friendly. The previsions of the increase of wind power are 50 percent by 2020 and 84 percent by 2035. The danish power system is part of one of the most strong power systems of Europe. This power system is called Nord Pool Spot and is one of the markets with higher exchange trade of electrical power. This market owns the power system of Norway, Denmark, Sweden, Finland, Estonia, Latvia, Lithuania, Germany and the UK. Denmark has a interconnection ratio of 44 percent. This is very important to take into consideration.

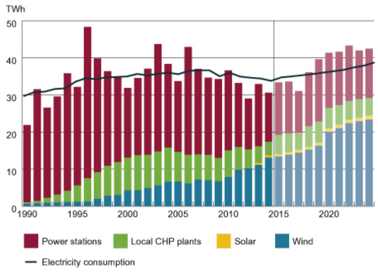


Figure 21: Power Consumption in Denmark. Spurce: http://www.energinet.dk/

A high interconneciton ratio means a high security of supply, a balance of the loads, a higher efficiency of the interconnected systems and also a better integration of alternative sources of energy like wind power or biomass. This makes Denmark a country with a solid power system. As we can see in this picture Denmark has a high percentage of Renewable Energy production compared to the Power Stations production. The eind, Hydro and Solar Energy are the 47 percent of the production in Denmark, the Biomass / Waste the 13 percent, the Natural Gas 7 percent, the Nuclear 3 percent and the Coal lingite the 30 percent. The prices of energy costs in Denmark are ridiculously low, including the renewable ones. However, it has to be taken into account that the danish citizens pay high taxes which partly are destinated to support the energy system. The consumption per capita in Denmark is 40503.303 kWh/year.

7.2.2 Consumer Behaviour and Load profile

The consumer behaviour and load profile of this country is going to be presented with the folowing images:

The proportion of the types of loads connected in residential areas and rural areas is:

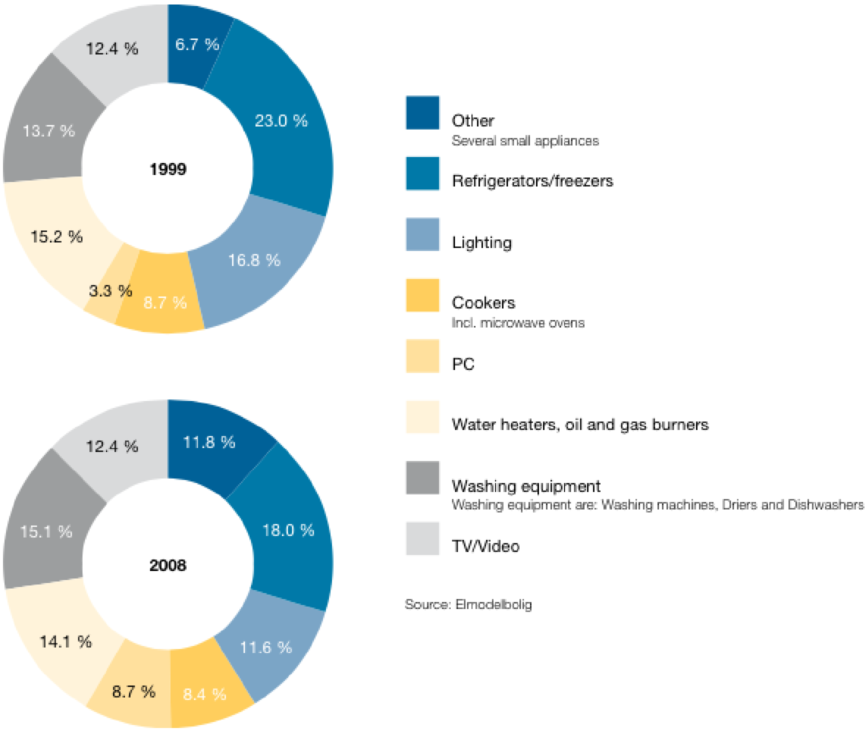


Figure 22: Danish Residential consumption according to types of appliances [Energi, 2008]

Also, the load profile of the danish citizens:

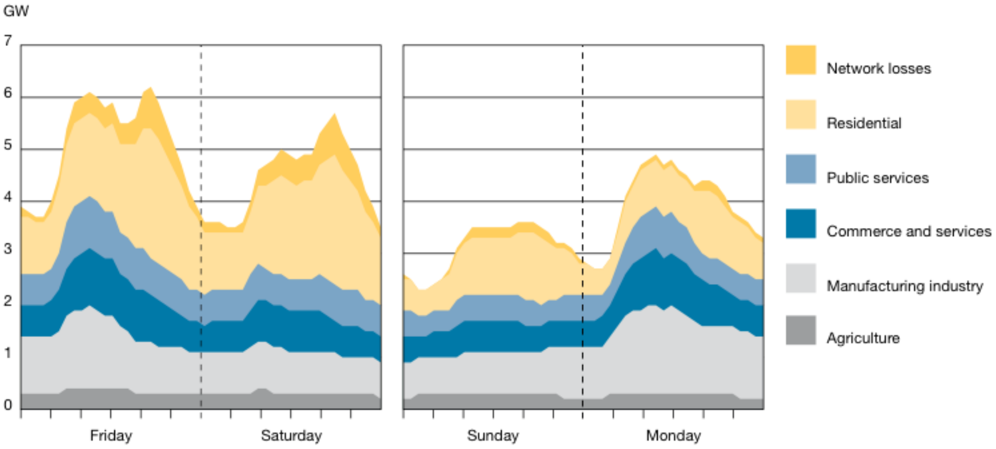


Figure 23: Electricity load curves by sectors

[Energi, 2008]

As it can be seen in the pictures the consumer behaviour there is a big demand of Water heaters and oil gas burners. However, there is not an air conditioned rate consumption. This, logically can be due to the average temperature in Denmark.

The load profile, however, is different from the Spanish one for example, the shape is more less the same one but there is a shift of few hours. This, also, can be due to the different lifestyle timetables of the danish and spanish society.

Also, the structure of the rural networks and the power generation is significant compared to other countries.
All these differences mean that the scenarios previously built for France and Spain may not be valid for northern countries of Europe. Therefore, another type of scenario may be built if an analysis of these countries want to be done. The results will carry a variation of the ZIP curve. This can means for example, that without that amount of air conditioned maybe that *'summer shift'*

that appeared in the previuos simulations doesnt appear here.

7.3 The Netherlands

7.3.1 Dutch Power Systems, general aspects

The dutch power system is synchronized in all the country. It has a strong interconnection with Germnay, being the Transmission System Operator of the Netherlands (Tennet) the same as big part of Germany. It is also interconnected with the UK and Norway by high voltage direct current interconnection and there is a future plan to connect the Netherlands and Denmark. The interconneciton with Norway is called NorNED and it was first put in use in 2008 after an agreement between Tennet and Statnett (norwegian TSO). With the UK the interconnection BritNED came 3 years later in 2011 after an angreement between Tennet and National Grid (British TSO). As a result of these interconnections then the Netherlands has a big security of Supply like Denmark.

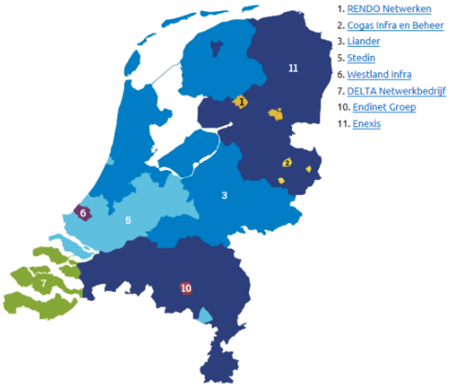


Figure 24: Structure of the dutch power system. Source: Power Point Presentation in the course of Electric Power Systems of the future EE4454 Tu Delft.

In the past 6 years, there has been a decrease of the interruptions of 10 percent in the high voltage net and 4 percent in the midle voltage net. this supposes an big increase in the reilability of the dutch power system and makes it very stable and also with a good market balance. The primary reserve is covered by Tennet and some German TSOs.

Even though the TSO (Transmission System Operator) is the same for all the country and part of Germany, in the Netherlands, there are eight different Distribution system operators. This Operators work in different geographical areas. The biggest DSO is called Lianet and it operates in the provinces of North Holland, Flevoland, Friesland and Gelderland.

The power demand in the Netherlands always had a strong co-depence with the Gross Domestic Product. It has grown substantially since 2000. From 2000 to 2008 it raise at an avergage growth of 1,7 percent per year, but, in 2009 because of the eocnomic crisis the demand dropped a 4.6 percent from 2008 to 2009 and remain constant from 2009 to 2013. The energy consumption per capita in the Netherlands is 58600,71 kWh/year, a 44 percent higher than the danish consumption per capita.
The power supply in the Netherlands has not changed a lot in the last 15 years. The high avialability of natural gas resources in the north of the country and under the North Sea have been supplying big part of the demand of the dutch power market. In the last ten years the Natural Gas power plants have covered 50 percent of the demand. of the country. The nuclear power generation however is not very big in the Netherlands, being onyl a 4 percent of its

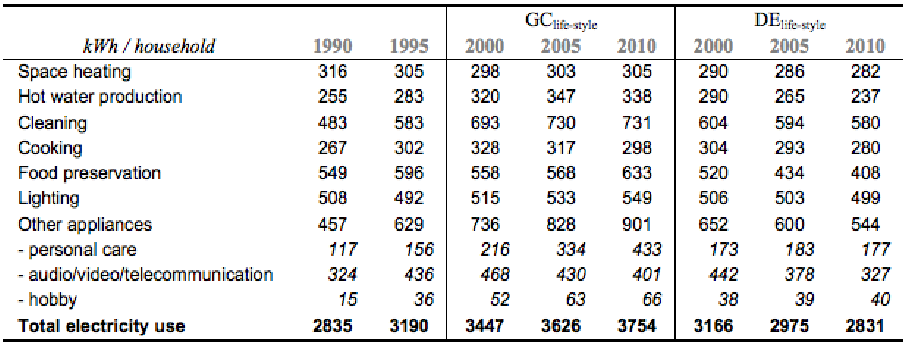
However, in the last 10 years, there has been an increase of the renewable sources of energy. Approximately 13 percent of the power generation in the Netherlands is renewable, which is not a lot compared to Spain (34 percent) and Denmark (47 percent). The dutch government in 2011 diverse targets for the upcoming decades:

- A reduction of CO2 emissios of 90 percent compared to 1990
- A 16 percent of renewable generation by 2020 with an increase of power generation of offshore and onshore wind power plants

7.3.2 Consumer Behaviour and Load Profile

The consumer behaviour and load profile of this country is going to be presented with the folowing images:

The proportion of the types of loads connected in residential areas and rural areas is:



			$GC_{life\text{-}style}$			$DE_{life\text{-}style}$		
kWh / household	1990	1995	2000	2005	2010	2000	2005	2010
Space heating	316	305	298	303	305	290	286	282
Hot water production	255	283	320	347	338	290	265	237
Cleaning	483	583	693	730	731	604	594	580
Cooking	267	302	328	317	298	304	293	280
Food preservation	549	596	558	568	633	520	434	408
Lighting	508	492	515	533	549	506	503	499
Other appliances	457	629	736	828	901	652	600	544
- personal care	*117*	*156*	*216*	*334*	*433*	*173*	*183*	*177*
- audio/video/telecommunication	*324*	*436*	*468*	*430*	*401*	*442*	*378*	*327*
- hobby	*15*	*36*	*52*	*63*	*66*	*38*	*39*	*40*
Total electricity use	**2835**	**3190**	**3447**	**3626**	**3754**	**3166**	**2975**	**2831**

Figure 25: Dutch Residential consumption according to types of appliances [Papachristos, 2014]

As it can be seen, less than in Denmark but also significantly high, the Hot water production and Space heating account for the 20-25 percent of the consumption. This makes also that the ZIP model will also may vary respect from the one built in the simulations. Also the structure of the Networks of the Netherlands is different. There is a large Semi- Urban Network in the Ranstad area, and then in the interior of the Netherlands there are mainly rural grids. In this country, also, there are many distribution systems in charge, and the Transmission system operator is the same as the one in the West of Germany. The load profile of the Netherlands is very similar to the load profile of Denmark. This is due to the similar timetables and societal way of living that both countries have.

Therefore, if a Simulation of the Dutch grid wants to be made, the consumer behaviour and the proportions of each device presence in a consumption point have to be taken into account.

7.4 Spain

7.4.1 Spanish Power System

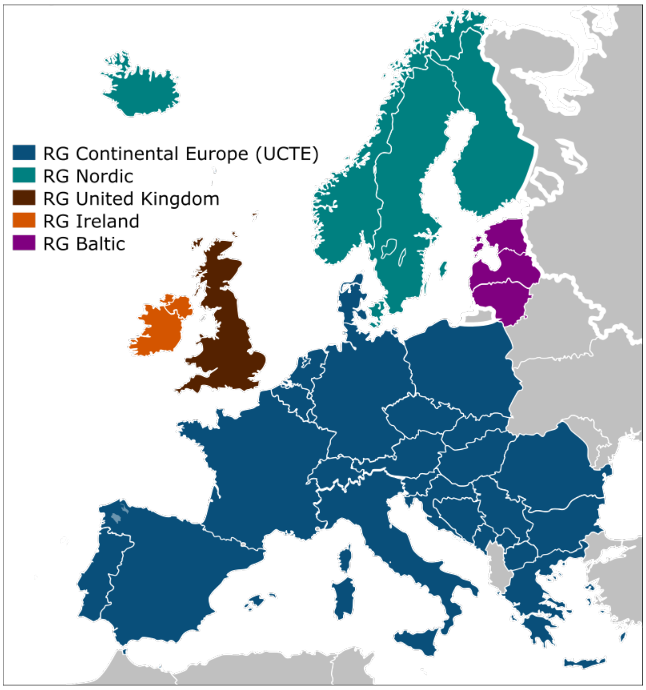


Figure 26: Source: Wikipedia.org

The Peninsula Iberica composed by the countries Andorra, España and Portugal is synchronized with the majority of countries of Europe as we can see. However, it was known till 5 years ago as the "electrical island". This is because its interconnection ratio was extremely low: 3 percent. As a result, in 2003, a new European Electrical Project came up: the electrical interconnection between France and Spain. This project had the main purpose of double the percentage of interconnection ratio from 3 to 6. On the 27 of June of 2008, both Spanish and French governments agreed in creating a company called Inelfe that would be in charge of building and managing the construction of this line. This line would connect the municipalities of Santa Llogaia (Spain) and Baixas (France). After this project was finishied the interconnection ratio between France and Spain was 6 percent. Also, it is important to mention that Spain has a unique electrical characteristic and is the electrical interconnection with Moroco wth two underground sea cables with a voltage level of 400 kV. In adition, Spain is connected to Portugal by means of 7 electrical lines 4 of 400 kv and 3 of 200 kV.

Since 1997 when the energy market liberalization ocur in Spain the power system market is owned by Red Electrica Española (REE), that the Transmission System Operator. There are five distribution system operators in Spain: Endesa, Iberdrola, Union Fenosa, E.ON and EDP who are in charge of distributing low voltage level.

The generation of electricity in spain has also changed substantially in the last years. There was a strong impulse of the renewable energies in Spain since 2000 for decresing the dependance with Europe and reach the requierments that the Kyoto Protocol established. This made that by 2013, 42 percent of energy production in Spain came from natural resources, most of them from wind and solar energy. There has been a strong decrease in the las 20 years from the production of nuclear power and carbon power.

7.4.2 Consumer Behaviour and Load Profile

The consumer behaviour of Spain and the Load Profile have been explained and developed . The main difference with the two european explained before is:

- Different ratio of use of appliances due to the temperature and the lifestyle
- Shift in the Load Profile because of the daily rythm

7.5 Conclusions and application to the project

This short introduction orientates of how is the european electrical grid. There are several regions and each of this regions have different aspects. This has to be taken into account when doing a study of certain grids because the improvments that can be done to them have to be based in their characteristics. The main ideas that has to be extracted from this chapter are:

- Structure of the different European Electrical Grids
- The Consumer Behaviour of each country is different, and it has to be considered when analyzing these grids.
- The load Profile of each country ihas to be considered when analyzing them.

The DSOs in each country are responsible for the voltage control of the low-medium level voltage grids. If any modification has to be made (as it is going to be said in this project) to the actual control system that the DSOs apply, it has also to be taken into consideration the structure of the grid of the country and also the load profiles and consumer behavior. If DSOs are making a non optimized voltage control of the grid, energy losses are going to increse. Also, this will affect to its country neighbours because if the losses are to high and alternative injection of power is needed in order to mantain the grid with a good functioning level.

As it has been explained, each country has big differences. Also, each the consumers of each country is different and this is an important thing to take into account. In this project the behaviour of the loads were deeply studied. But, of course, the summer and the winter in each country is not the same. Therefore the amount of Air Conditioned or Heaters used in each country are not going to be the same. Also, other devices are going to have a heavier presence in some countries due to temperature, way of living, activities..etc.

8 Conclusions and future work

8.1 General Conclusions

Four types of grids have been simulated before. Not only spanish but french too. There are many aspects an ideas that can be extracted from the results obtained:

- **Voltage / Reactive Power Control**: This ancilliary system is very important to consider. It has been already introduced in the chapter *Theory of Energy Losses.* It can increase the efficiency of the grid making energy losses decrease. The reactive power control can be done, for example by connecting capacitors in paralel with the circuit. However in distribution networks as it can be deduced from the previous simulations if a capacitor bank is connected far away from the consumption the current will be higher trough the line and this will increase the power losses. If we put ourselves in the case of a rural grid, in which the transformer is far away from the village, it will be better to put the capacitor bank at the entrance of the village than at the Transformation center, that usually is outside the village. This way you will reduce the current trhough the line that goes from the Transformation center to the entrance of the village. This also happens with industrial, urban and semiurban grids. Therefore reactive power control if its done good, can help to reduce energy losses.

  As it has been in the simulations the variation of voltage from one point to another is very important, because it can happen that the first point of consumption is 232 V and the last one 228 V. This is crucial when considering the loads with a ZIP model because as it can be seen in Appendix A, certain loads can vary their slope voltage curve from positive to negative when they are in a undervoltage or overgoltage situation. However, if the reactive power control can help to make a voltage profile with closer voltage level from one point to another this can help regarding the energy losses control.

Also the repercutions that a good voltage control has for consumers and distribution / transmission systems are very relevant.

  - Consumers: the devices that the consumers conect to the grid must have a safe connection. This means that most of the devices are made for being between a range of voltage. This directly affects to the lifetime of the device. For example, if a light bulb is constantly varying between 260 and 220 V, eventually it will stop working. That is why the distribution system must ensure a voltage level between two reasonable values, and, for the consumers a good voltage control will mean the big benefit of their loads to be with a safe and good connection.
  - Distribution and Transmission: The benefits of a reactive power control for the TSOs and DSOs have been explained before. Used in a good way, it can reduce the energy losses and this will make the bussines responsible for the power transmission and distribution loose less money.

- **Impact of the Distributed Generation in the Energy Losses**: As it was explained in the state of art [Méndez Quezada, 2005], Distributed generation can help to reduce energy losses. It has been demonstrated that for some penetration levels of distributed generation the energy losses decrease. Sometimes, Distributed generation have a bad impact on Energy Losses but for a certain level of penetration it helps. Therefore, for consumers can be an attractive idea to install solar panels, that adjust more to the daily demand curve (mostly on summer) in their houses. That way they will save money in terms of paying the distribution company for the power supplied and also Distribution companies will be an increase in efficiency.

This of course will be also better for the environment. If the particular consumers star also to produce green energy a better sustainable power system will be being developed. Therefore, if distributed generation is integrated in the power system with the allowed level of penetration DSOs, consumers and the environment will benefit.

- **Invalidity of the Constant Power Model**. In the simulations it has been considered the constant power model. This model assumes that the active power consumed by a device is the same at al levels of voltage. In theory, if we assume this type of model and considering the equation P = U * I, If the voltage increase the current must decrease. However, it has been demosntrated that the power consumption doesnt remain constant with a variation in voltage. Also, it is far away a good method for optimizing the energy losses in a distribution network as it has been seen.
- **Proximity of the Constant Impedance Model to the reality**. This model, as it can be seen in the results gives a better aproximation to whats happening with a variable voltage. Nevertheless for lower voltage levels this model differs from the ZIP model, while the ZIP model shifts ver slowly from a Constant Impedance model to a Constant Power Model.
- **Prensence of Induction Motors in a distribution network**. It is important also to mention how the presence of induction motors can vary for lower voltage levels the constant impendance model to the ZIP

model. This is due to the reaction that Induction Motors have with lower voltages. This can be seen in this graph:

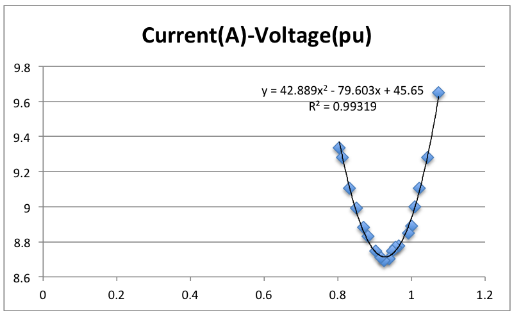


Figure 27: Current test of an 10 A / 230 V Induction motor at a variable voltage

So, it can be seen in the Simulation of an Industrial distribution network how the active power losses are always close to the nominal energy losses of the network. This is due to the shifhting above mentioned

- **Influence of the time of the year in energy losses**. It has been observed that when in urban, rural and semiurban grids the presence of heaters in winter provoke more losses for lower voltages. However, the Air conditioned adjust more to the Constant Impedance Model.

- **Optimization of the voltage control**. After this project, improvements to the voltage control can be done. If the loads in distribution networks are modeled as ZIP loads, a better control of the energy losses can take place because the real reaction of the loads facing a variable voltage is known. If a constant model is taken, it is reccomendable to use the Constant Impendance Model as it adjust more to the reality than Constant Power model. Also few correction factors can be used for low voltage levels if a Constant Impedance Model is used. These correction factors may be modelled taking into account the level of penetration of the induction motors load type in the network.

- **Consideration of Consumer Behaviour and Load Profile of the different european electrical grids**: As it has been explained in the provious chapter, each country has its own distribution and transmission structure that must be considered if a simulation of these countries wantts to be done. The influence of the lifestlyle, rythm and timetable can reach a big difference in each country.

8.2 Future work

I can be interesting to study further the impact of voltage control in these different countries taking into account the structure. Also, the study of load profiles in distribution networks of different countries of Europe can expand the conclusions that refer to an impact of the difference ”typical” load profiles and consumer behaviour of each country. Also, studying deeply the proportion of types of grids that are in each country.

9 Appendix A

In this appendix is represented the measurements carried out in the Laboratory of Electrical Machines of ICAI School of Engineering. Many devices were tested in the months of May and June of 2015. For all the tests a VARIAC, a HAMERG Grid Anlyser and a connection to the grid were used. The VARIAC allowed to vary the voltage and the HAMERG permitted to know exactly the value of the current, voltage, active power and reactive power consumed. The characteristics of the devices that are going to be tested are:

Table 10: My caption

Device	**Voltage**	**Current**	**Frequency**	**Brand**	**Power**	**rpm**
Discharge Lamp	230V	-	50 Hz	Philips	250 W	-
Lamp	220 V	-	50 Hz	ICAI	500 W	-
Programmable Power Meter HM8115-2	50 V 150 V 500 V	0.16 A 1.6 A 16 A	50-60 Hz	HAMEG	-	-
Heater 1	220 V	-	50 Hz	ICAI	1000 W	-
Heater 2	220 V	-	50 Hz	ICAI	2500 W	-
Induction Motor	230 V	8.2 A	50 Hz	ICAI	2200	1420
PC and Screen	220 V	-	50-60 Hz	Philips	-	-

Table 11: Technical Characteristics of the devices that are going to be measured

.

9.1 Measurements at a variable voltage

9.1.1 Lamp 500 W

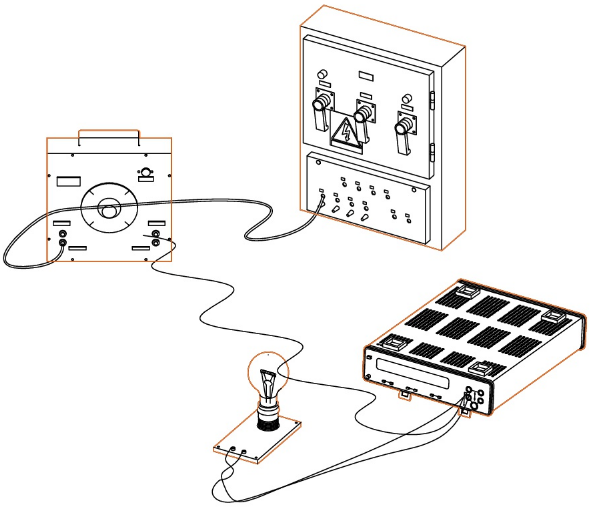
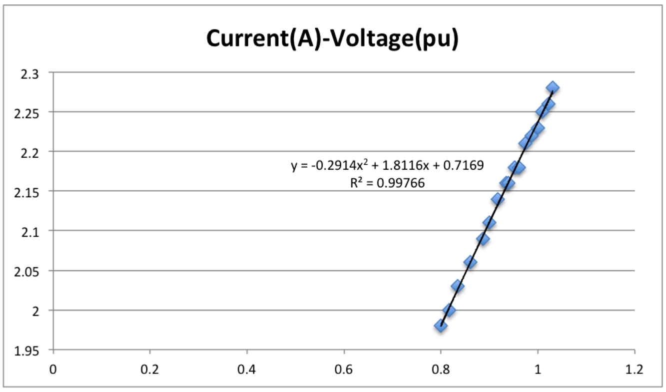


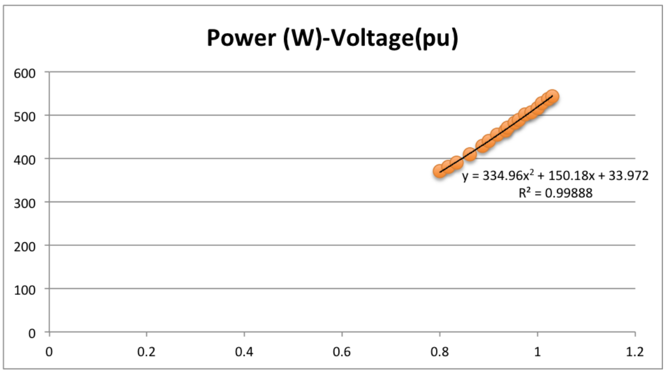
Figure 28: Test of a Lamp 500 W

Voltage	**Voltage (pu)**	**Intensidad**	**P**	**Q**	**cosfi**	**lux**
237	1,030434783	2,28	544	70	0,991822593	670
235	1,02173913	2,26	537	70	0,991610705	650
232	1,008695652	2,25	527	65	0,992479363	618
230	1	2,23	516	60	0,99330738	611
227	0,986956522	2,22	507	59	0,993296922	570
224	0,973913043	2,21	502	60	0,992932885	550
221	0,960869565	2,18	489	58	0,993039263	529
219	0,952173913	2,18	482	54	0,993782747	510
216	0,939130435	2,16	471	53	0,993728386	485
215	0,934782609	2,16	465	51	0,994039156	481
211	0,917391304	2,14	455	47	0,994707214	466
207	0,9	2,11	441	47	0,994368709	426
204	0,886956522	2,09	429	44	0,994781439	404
198	0,860869565	2,06	410	44	0,99429079	371
192	0,834782609	2,03	391	44	0,993727786	329
188	0,817391304	2	380	44	0,993363062	304
184	0,8	1,98	371	44	0,993040543	286

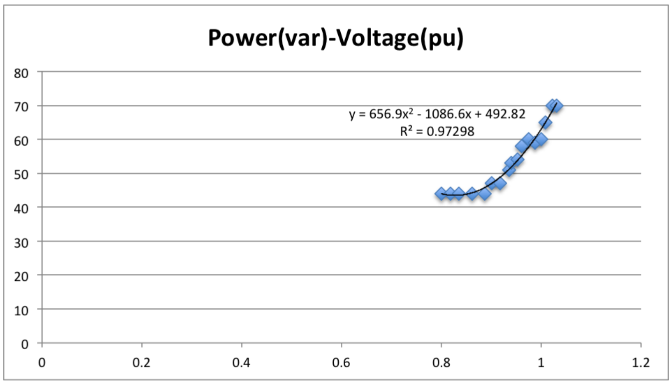
Table 12: Lamp Measurements



(a) Current Consumption



(b) Power(W) Consumption



(c) Power (var) Consumption

Figure 29: Test at variable voltage

9.1.2 Discharge Lamp

Voltage	**Voltage pu**	**Current**	**P**	**Q**	**Cosfi**	**lux**
238	1,034782609	1,357	292	137	0,905309943	1827
237	1,030434783	1,345	291,4	130	0,913242013	1791
236	1,026086957	1,334	288	123	0,919639798	1787
235	1,02173913	1,333	288	123	0,919639798	1780
234	1,017391304	1,323	285,3	120	0,921781271	1769
232	1,008695652	1,309	281,4	117	0,923367901	1734
231	1,004347826	1,302	278,8	115	0,924444445	1708
229	0,995652174	1,283	272,6	111	0,926162692	1699
226	0,982608696	1,265	266,1	106	0,929005444	1628
224	0,973913043	1,25	261,1	105	0,927788935	1586
222	0,965217391	1,236	254	104	0,925430839	1529
220	0,956521739	1,225	250,1	103	0,924655126	1507
218	0,947826087	1,211	245	103	0,921847716	1462
215	0,934782609	1,19	234,5	100	0,919853665	1390
212	0,92173913	1,17	228,5	100	0,916111321	1363
209	0,908695652	1,15	220,4	98	0,913743151	1314
207	0,9	1,137	215,3	98	0,910149194	1281
204	0,886956522	1,113	205	97	0,903917326	1187
203	0,882608696	1,105	201,8	97	0,901285962	1150
201	0,873913043	1,109	197,3	96	0,899206013	1109
198	0,860869565	1,1077	193	94	0,899036891	1064
197	0,856521739	1,067	189,4	93	0,897626703	1050
194	0,843478261	1,05	184	88	0,902134222	1006
192	0,834782609	1,041	181	87	0,901289816	988
191	0,830434783	1,03	176,6	86	0,899061812	981
188	0,817391304	1,01	172	82	0,902666414	945
185	0,804347826	0,997	167	80	0,901859867	902
183	0,795652174	0,986	163	78	0,902040548	875

Table 13: Discharge Lamp Measurments

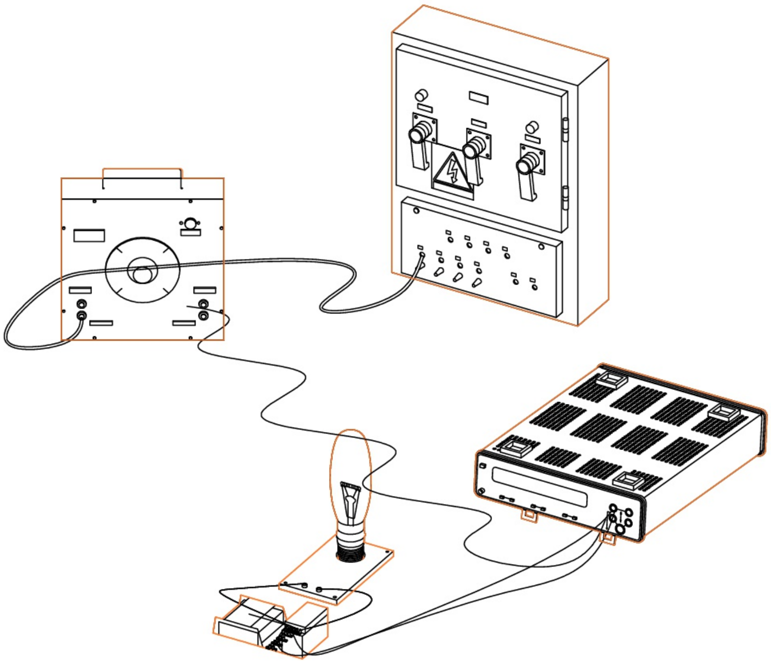
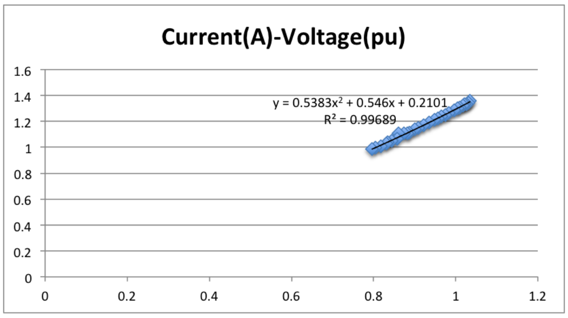
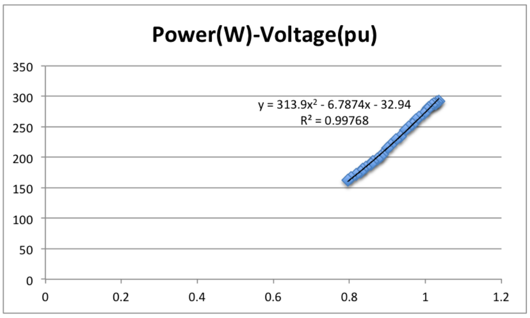


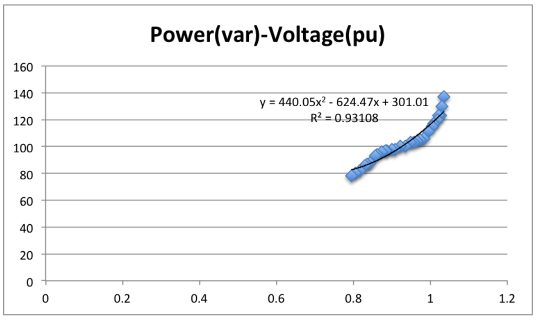
Figure 30: Test of a Discharge Lamp



(a) Current Consumption



(b) Power (W) Consumption



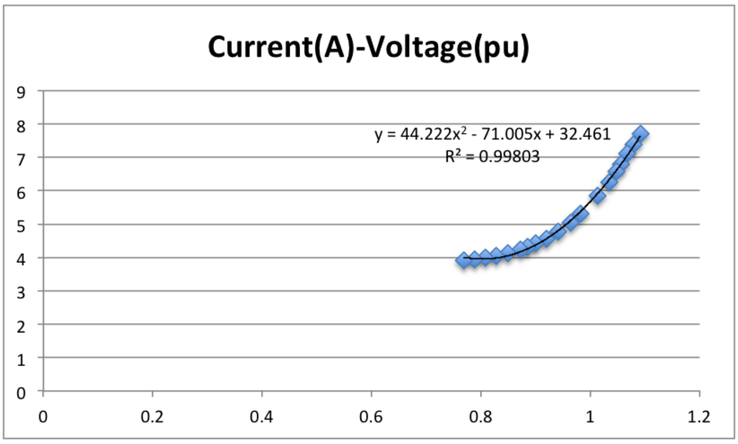
(c) Power (var) Consumption

Figure 31: Test of a discharge lamp at variable voltage

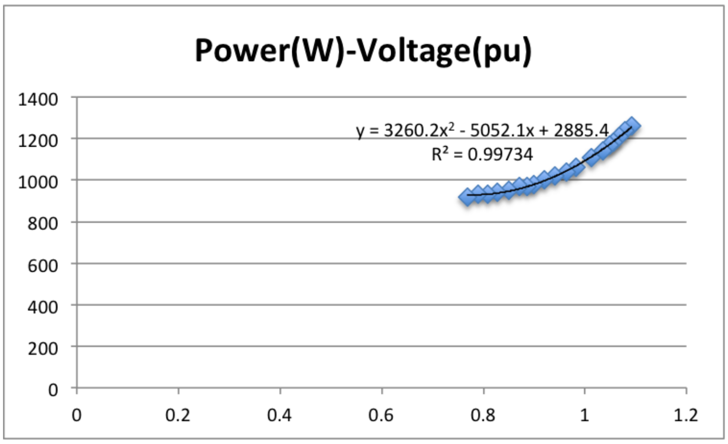
9.1.3 Induction Motor 220 V 2.5 A

Voltage	Voltage pu	Current	P	Q	cosfi
251	1,091304348	7,703	1260	3200	0,366371962
248	1,07826087	7,39	1240	3000	0,381988977
245,7	1,06826087	7,106	1210	2840	0,391963616
243	1,056521739	6,782	1180	2670	0,404230501
241	1,047826087	6,58	1170	2560	0,415675797
238	1,034782609	6,266	1140	2400	0,429056815
233	1,013043478	5,843	1110	2160	0,457068867
226	0,982608696	5,306	1060	1850	0,497148774
221,6	0,963478261	5,058	1040	1690	0,524097426
216,4	0,940869565	4,777	1020	1530	0,554700196
211,5	0,919565217	4,567	1000	1390	0,583996993
207	0,9	4,415	980	1290	0,604926747
203,6	0,885217391	4,307	970	1220	0,622344935
200,5	0,87173913	4,24	970	1170	0,638240774
195,5	0,85	4,124	950	1070	0,663929943
190,5	0,82826087	4,038	940	990	0,688556926
186	0,808695652	3,987	930	930	0,707106781
181,5	0,789130435	3,935	930	860	0,734197815
177	0,769565217	3,903	920	820	0,746513272

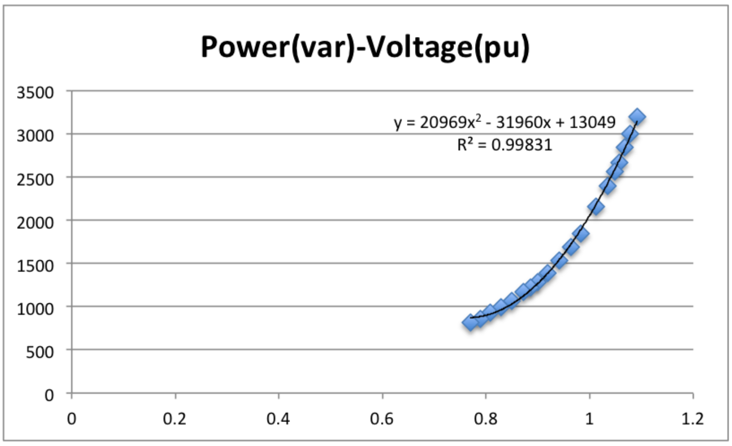
Table 14: Induction Motor 220 V 2.5 A Measurements



(a) Current Consumption



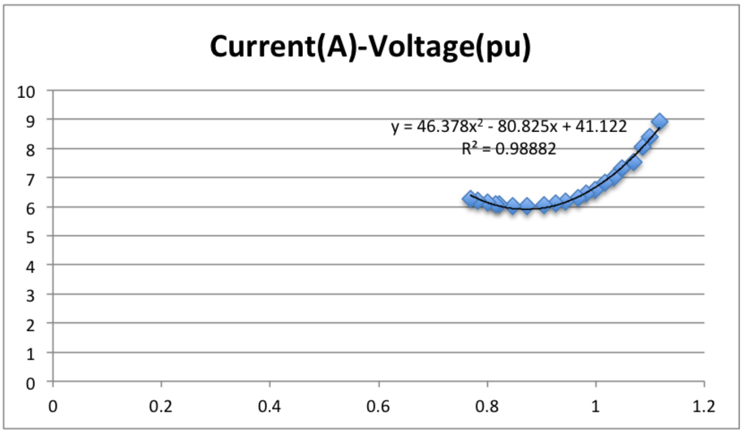
(b) Power (W) Consumption



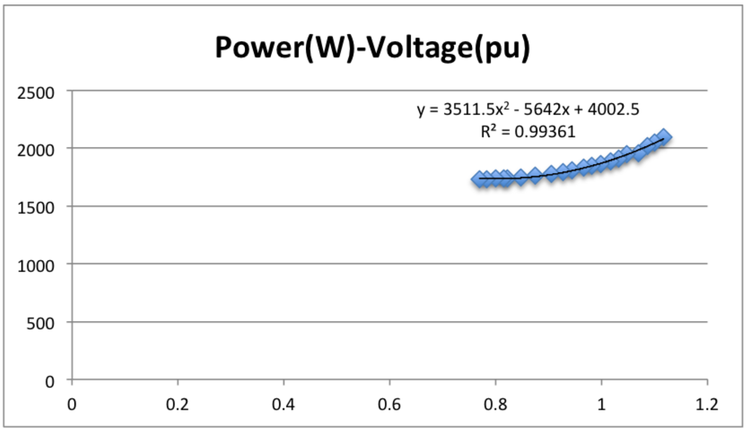
(c) Power (var) Consumption

Figure 32: Essay of a induction motor at variable voltage

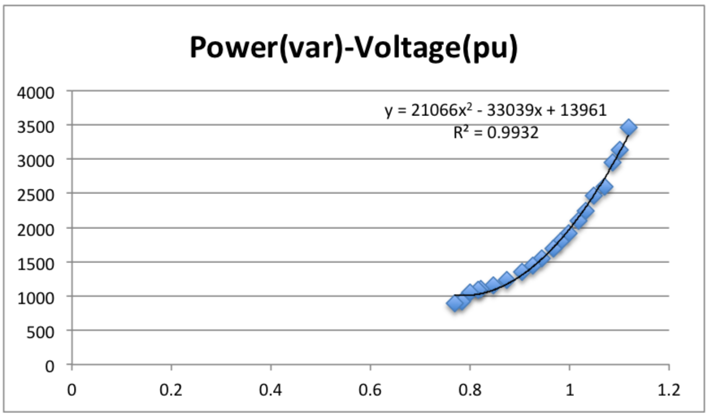
9.1.4 Induction Motor 220 V 5 A



(a) Current Consumption



(b) Power (W) Consumption



(c) Power (var) Consumption

Figure 33: Test of a induction motor at variable voltage

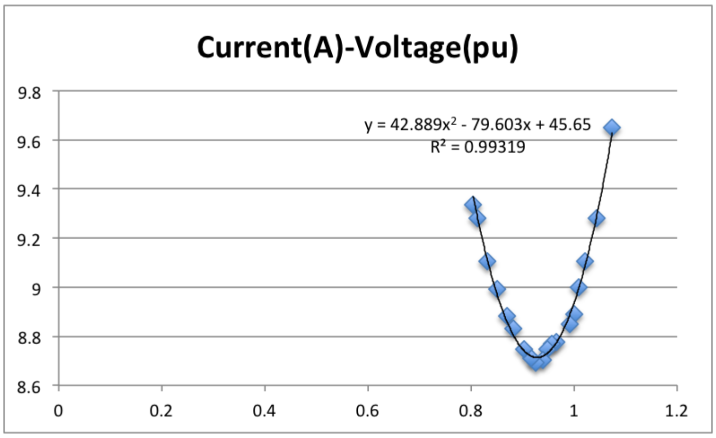
Voltage	**Voltage pu**	**Current**	**P**	**Q**	**Cosfi**
257	1,117391304	8,913	2100	3460	0,518849182
253	1,1	8,39	2050	3140	0,546674442
249,7	1,085652174	8,054	2020	2950	0,564984683
246	1,069565217	7,536	1960	2600	0,601963785
241	1,047826087	7,328	1950	2470	0,619644289
237,5	1,032608696	7	1910	2250	0,647156521
234	1,017391304	6,809	1890	2100	0,668964732
229,5	0,997826087	6,585	1860	1920	0,695795216
225,8	0,98173913	6,438	1850	1800	0,716725931
222,3	0,966521739	6,311	1830	1690	0,734649516
217	0,943478261	6,183	1810	1550	0,759552742
213	0,926086957	6,1	1790	1450	0,777042646
208	0,904347826	6,044	1780	1350	0,796765587
201	0,873913043	6,011	1760	1230	0,819669272
194,8	0,846956522	6,029	1750	1160	0,833512908
189	0,82173913	6,068	1740	1100	0,845257958
187,5	0,815217391	6,095	1740	1090	0,847450615
184	0,8	6,139	1740	1050	0,856187661
180	0,782608696	6,197	1730	920	0,882917406
177	0,769565217	6,267	1730	900	0,887132828

Table 15: Induction Motor 220 V 5 A Measurements

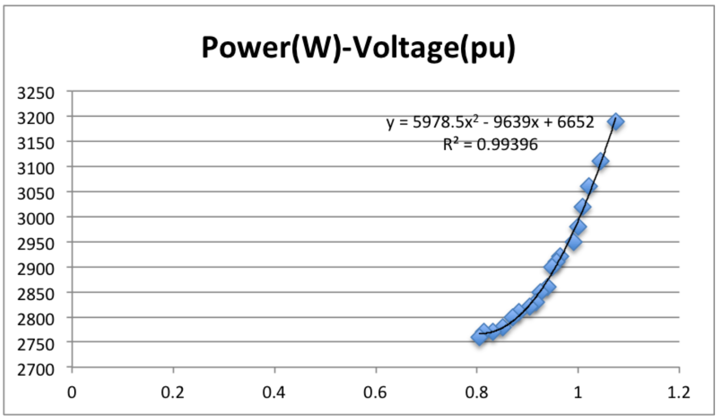
9.1.5 Induction Motor 220 V 10 A

Voltage	Voltage pu	Current	P	Q	cosfi
247	1,073913043	9,65	3190	2740	0,758584761
240	1,043478261	9,28	3110	2430	0,787985959
235	1,02173913	9,105	3060	2260	0,804394613
232	1,008695652	9	3020	2160	0,813369355
230	1	8,89	2980	2050	0,823880556
228	0,991304348	8,85	2950	1940	0,835519736
222	0,965217391	8,775	2920	1880	0,840804047
220	0,956521739	8,77	2910	1840	0,845212718
218	0,947826087	8,746	2900	1820	0,847012324
216	0,939130435	8,705	2860	1760	0,851658317
213	0,926086957	8,691	2850	1740	0,853503839
210,7	0,916086957	8,712	2830	1530	0,879671368
207,8	0,903478261	8,748	2820	1500	0,882872389
203	0,882608696	8,831	2810	1440	0,889949319
200	0,869565217	8,881	2800	1420	0,891864487
195,6	0,850434783	8,992	2780	1360	0,898271001
191,4	0,832173913	9,107	2770	1330	0,901472267
187	0,813043478	9,282	2770	1300	0,905262596
185	0,804347826	9,335	2760	1280	0,907188234

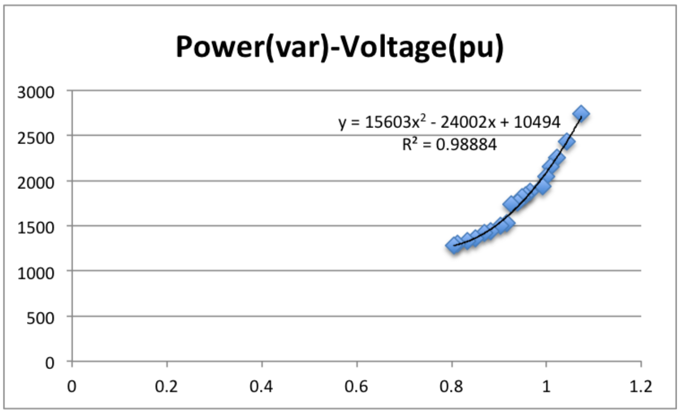
Table 16: Induction Motor 220 V 10 A Measurements



(a) Current Consumption



(b) Power (W) Consumption



(c) Power (var) Consumption

Figure 34: Test of a induction motor at variable voltage

9.1.6 Heater 1 KW

A 1000W heater is measured in different points of voltage level. In the table below it can be seen its reaction:

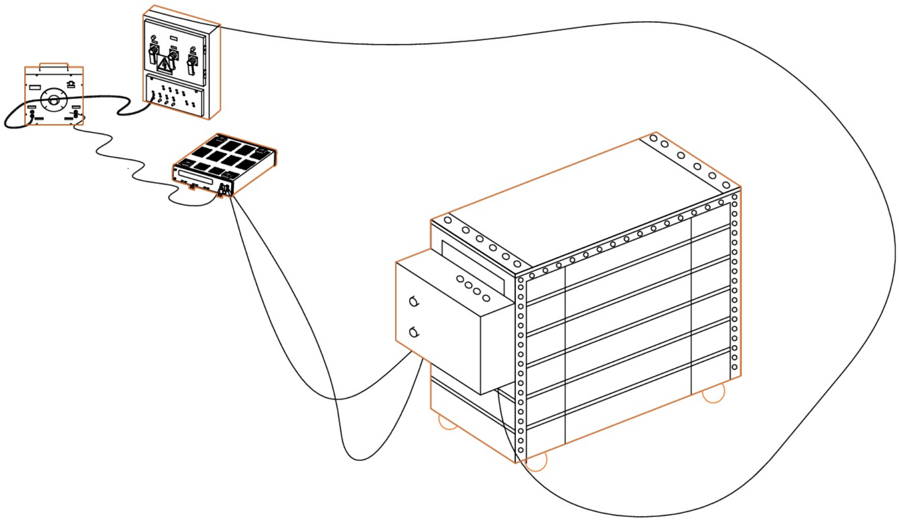
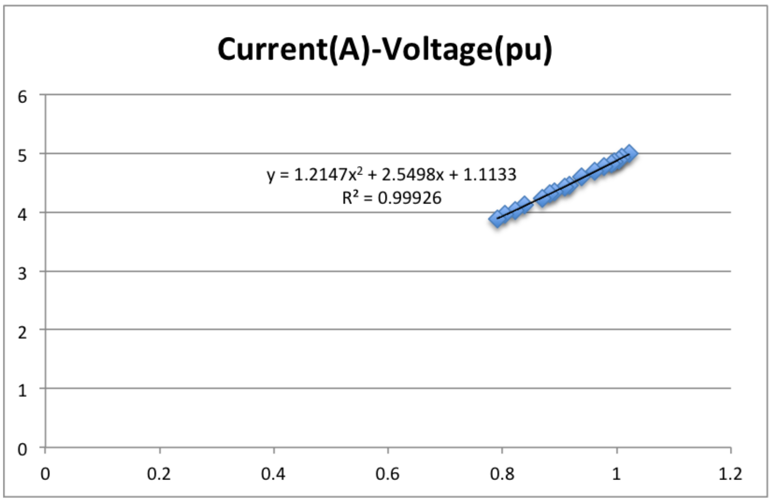


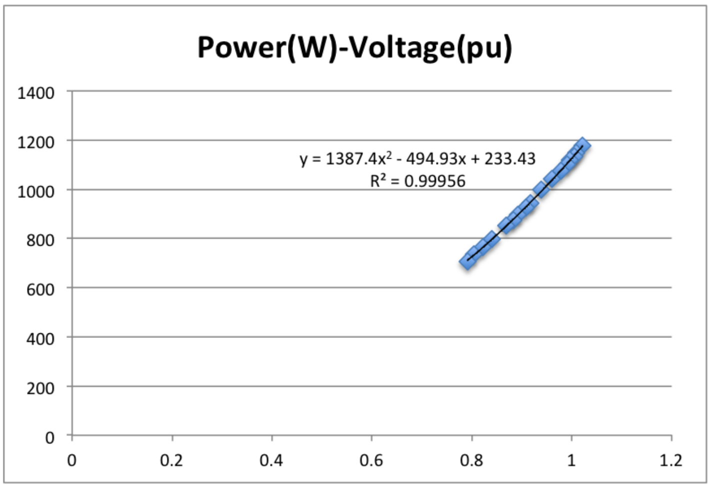
Figure 35: Test of a 1kW Heater

Voltage	**Voltage pu**	**Current**	**P**	**Q**	**Cosfi**
235	1,02173913	4,99	1176	86	0,997336734
232	1,008695652	4,93	1148	85	0,997270124
231	1,004347826	4,89	1134	82	0,997395811
229	0,995652174	4,86	1120	79	0,997521601
228	0,991304348	4,82	1100	68	0,998094715
225	0,97826087	4,77	1076	57	0,998599825
221	0,960869565	4,69	1042	57	0,998507168
216	0,939130435	4,6	998	57	0,998372962
211	0,917391304	4,46	944	52	0,998486279
209	0,908695652	4,43	925	51	0,998483515
205	0,891304348	4,35	896	49	0,998507984
203	0,882608696	4,31	876	48	0,998502152
200	0,869565217	4,24	852	40	0,998899744
193	0,839130435	4,12	796	40	0,998739792
189	0,82173913	4,03	764	40	0,998632235
185	0,804347826	3,96	737	40	0,998530408
182	0,791304348	3,88	707	40	0,998403349

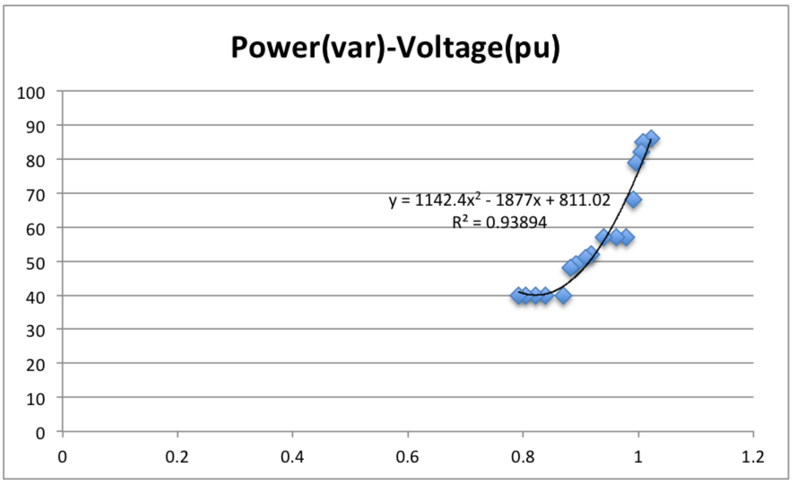
Table 17: Heater 1 KW Measurements



(a) Current Consumption



(b) Power (W) Consumption



(c) Power (var) Consumption

Figure 36: Test of heater 1KW at variable voltage

9.1.7 Heater 2.5 KW

A 2500W heater is measured in different points of voltage level. In the table below it can be seen its reaction:

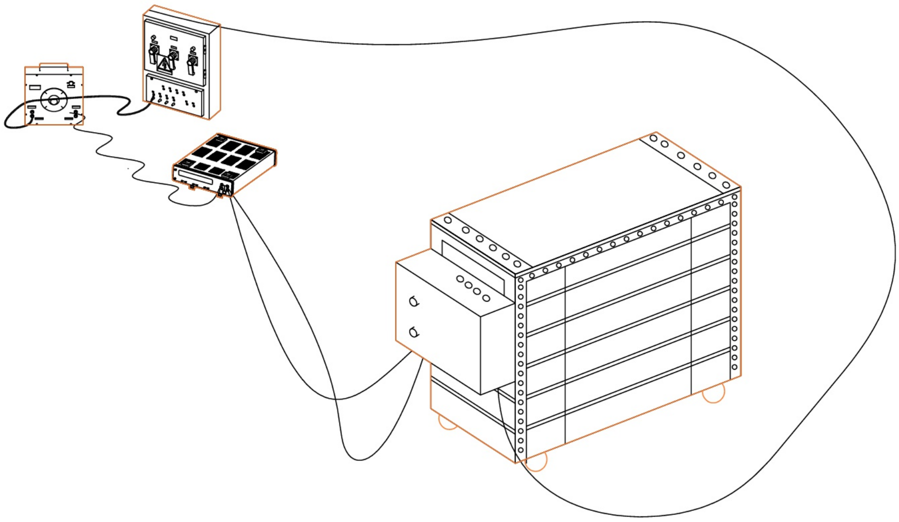
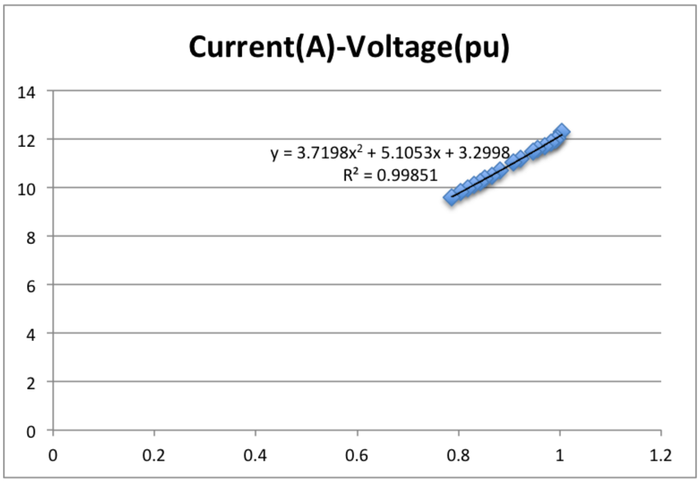


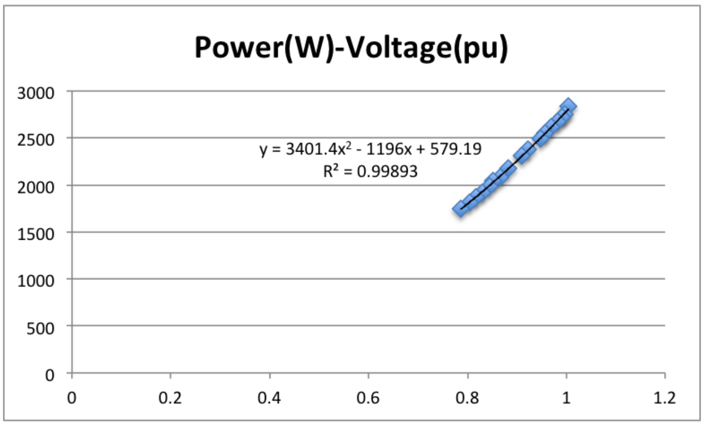
Figure 37: Test of a 2.5kW Heater

Voltage	**Voltage pu**	**Current**	**P**	**Q**	**Cosfi**
231	1,004347826	12,28	2840	105	0,999317242
229	0,995652174	12,05	2750	102	0,999312841
228	0,991304348	11,98	2724	99	0,999340224
226	0,982608696	11,86	2673	85	0,99949478
223	0,969565217	11,72	2614	83	0,999496283
220	0,956521739	11,58	2546	79	0,999518946
218	0,947826087	11,47	2492	78	0,99951051
212	0,92173913	11,19	2383	74	0,999518195
209	0,908695652	11,03	2311	73	0,99950147
203	0,882608696	10,71	2177	65	0,99955456
199	0,865217391	10,49	2083	66	0,999498406
196	0,852173913	10,39	2041	61	0,999553673
194	0,843478261	10,24	1985	60	0,999543486
191	0,830434783	10,1	1922	57	0,999560533
188	0,817391304	9,95	1871	57	0,999536265
185	0,804347826	9,82	1818	50	0,999622014
181	0,786956522	9,61	1749	50	0,99959162

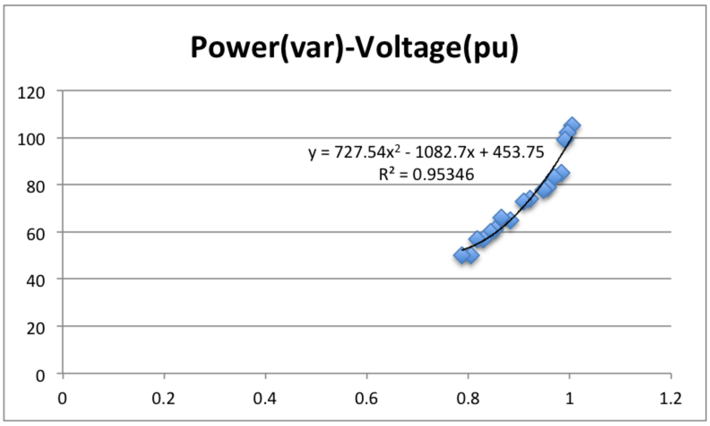
Table 18: Heater 2.5 KW Measurements



(a) Current Consumption



(b) Power (W) Consumption



(c) Power (var) Consumption

Figure 38: Test of heater 2.5 KW at variable voltage

9.1.8 Resistor 500 W

A 500W Resistor is measured in different points of voltage level. In the table below it can be seen its reaction:

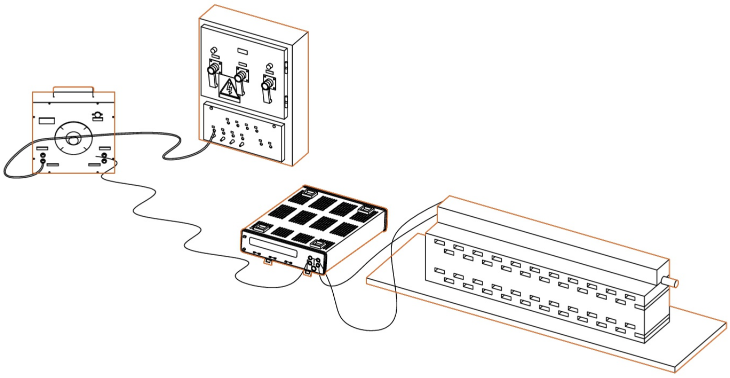
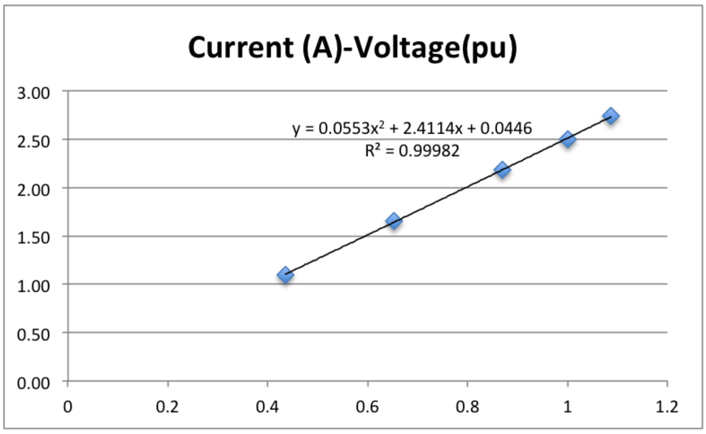


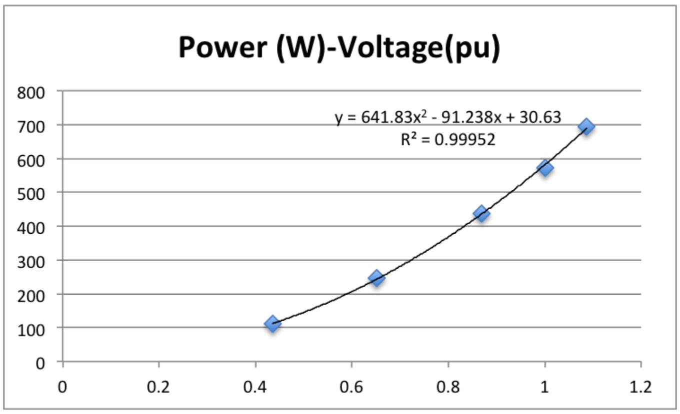
Figure 39: Test of a 0.5kW Resistor

Voltage	Voltage pu	Current	P	Q	Cosfi
100	0.434782	1,10	111	0	1
150	0.6521	1,65	247	0	1
200	0.8695	2,18	438	0	1
230	1	2,50	573	0	1
250	1.08695	2,74	695	0	1

Table 19: Resistor 500 W Measurements



(a) Current Consumption



(b) Power Consumption

Figure 40: Test of a resistor 500 W at variable voltage

9.1.9 PC and Flat Screen

A Dell PC and Flat Screen are measured in different points of voltage level. In the table below it can be seen their reaction:

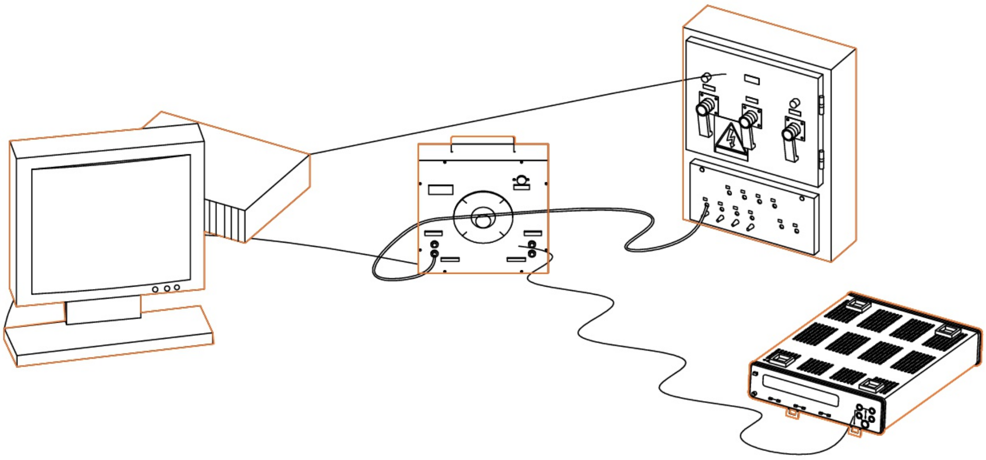
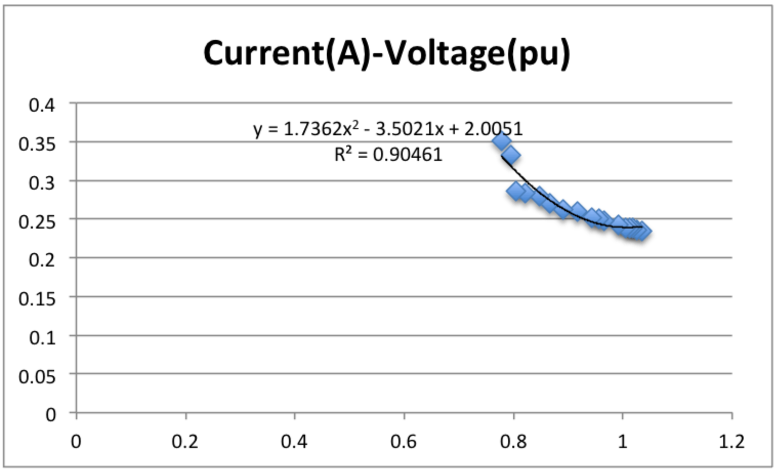


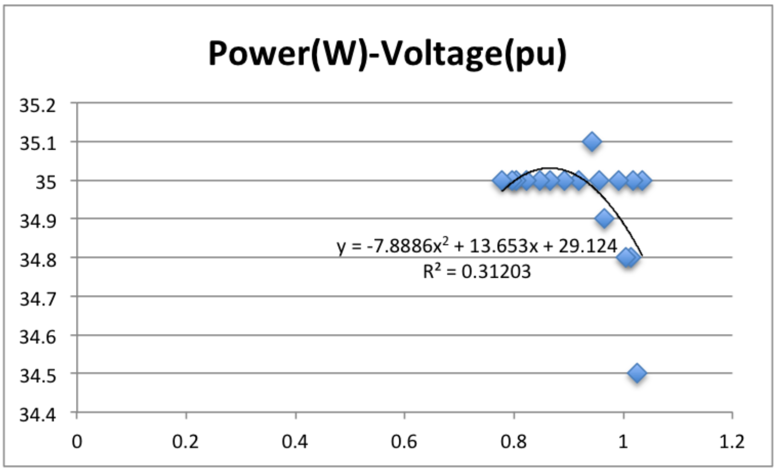
Figure 41: Test of a PC and Flat Screen

Voltage	**Voltage pu**	**Current**	**P**	**Q**	**cosfi**
238	1,034782609	0,235	35	45	0,613940614
236	1,026086957	0,236	34,5	43	0,625800466
234	1,017391304	0,238	35	43	0,631271398
233	1,013043478	0,238	34,8	44	0,62033761
231	1,004347826	0,239	34,8	43	0,629094216
228	0,991304348	0,242	35	43	0,631271398
222	0,965217391	0,248	34,9	43	0,630184665
220	0,956521739	0,25	35	43	0,631271398
217	0,943478261	0,252	35,1	42	0,641261973
211	0,917391304	0,26	35	42	0,6401844
205	0,891304348	0,263	35	42	0,6401844
199	0,865217391	0,27	35	40	0,658504608
195	0,847826087	0,28	35	40	0,658504608
189	0,82173913	0,284	35	40	0,658504608
185	0,804347826	0,286	35	39	0,667909742
183	0,795652174	0,332	35	40	0,658504608
179	0,77826087	0,351	35	40	0,658504608

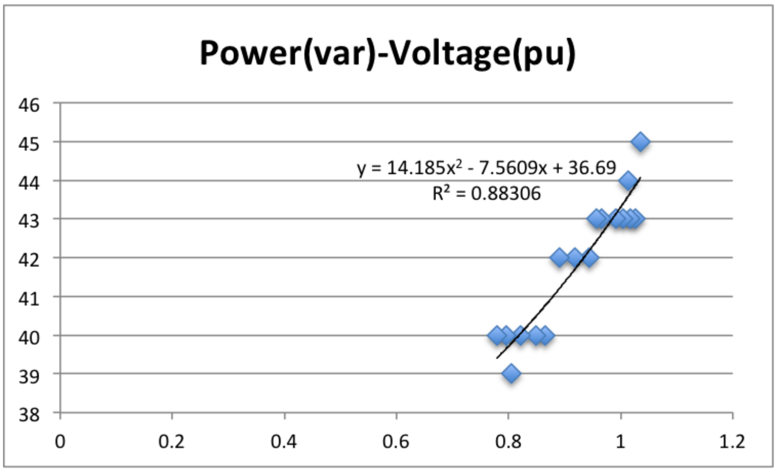
Table 20: PC and Flat Screen Measurements at a variable voltage



(a) Current Consumption



(b) Power (W) Consumption



(c) Power (var) Consumption

Figure 42: Test of a PC and Screen at variable voltage

Appendix: Simulator Tool

Final Project
Bacherlor in Electromechanical Engineering
ICAI School of Engineering
22. May 2016

Functions: Definition of the consumption of each load profile depending on their voltage level

The name of the functions define which type of load is:
P-ZIP-"Consumption Level and Type"-"Summer/Winter"

Final Project
Bacherlor in Electromechanical Engineering
ICAI School of Engineering
22. May 2016

```
function PZIPBinv = PZIPBinv(V,Ui)
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;
```

```
%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;
%Modelado te tipos de cargas/casas/comercios/industrias

%Perfiles de carga, Viviendas tipo A, B, C, y iluminacion

%Carga tipo B: Electrificacion b?sica-media: 4000-7000 W verano/invierno

PZIPBinv=2*Slamp+5*Sdlamp+Sfridge+Smi1+SPC+Sheater1+SPC;


end
```

```
function PZIPBver = PZIPBver(V,Ui)

%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
```

```
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

%Perfiles de carga, Viviendas tipo A, B, C, y iluminacion

%Carga tipo B: Electrificacion b?sica-media: 4000-7000 W verano/invierno

PZIPBver=2*Slamp+5*Sdlamp+Sfridge+Smi1+SPC+2*SAC+SPC;

end
```

```matlab
function [ PZIPAinv ] = PZIPAinv( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu

PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47    ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;
```

```
%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

%Perfiles de carga, Viviendas tipo A, B, C, y iluminacion

%Carga tipo A: Electrificacion media-alta: 7000-10000 W verano/invierno

PZIPAinv=4*Slamp+10*Sdlamp+2*Sfridge+Smi1+Smi1+2*SPC+2*Sheater1;
PZIPAinv=PZIPAinv+2*SPC+Sheater2


end
```

```
function [ PZIPAver ] = PZIPAver( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;
```

```
%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;
%Modelado te tipos de cargas/casas/comercios/industrias

%Perfiles de carga, Viviendas tipo A, B, C, y iluminacion

%Carga tipo A: Electrificacion media-alta: 7000-10000 W verano/invierno

PZIPAver=4*Slamp+10*Sdlamp+2*Sfridge+2*Smi1+2*SPC+2*SFan;
PZIPAver=PZIPAver+2*SAC+2*SPC


end
```

```
function PZIPCinv = PZIPCinv(V,Ui)

%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
```

```
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

%Perfiles de carga, Viviendas tipo A, B, C, y iluminacion

%Carga tipo C: Electrificacion Alta : 10000-15000 W verano/invierno

PZIPCinv=6*Slamp+12*Sdlamp+2*Sfridge+Smi3+2*Sheater1;
PZIPCinv=PZIPCinv+2*Sheater2+2*SPC

end
```

```
function PZIPCver = PZIPCver(V,Ui)

%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu

PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
```

```
end
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

%Perfiles de carga, Viviendas tipo A, B, C, y iluminacion

%Carga tipo C: Electrificacion Alta : 10000-15000 W verano/invierno

PZIPCver=6*Slamp+12*Sdlamp+2*Sfridge+Smi3+2*SFan;
PZIPCver=PZIPCver+4*SAC+2*SPC

end
```

```
function PZIPilu1 = PZIPilu1(V,Ui)
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;
```

```
%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;
%Iluminaic?n Tipo1:1kW , Tipo2: 4kW, Tipo3:8kW. Con lamparas descarga

PZIPilu1=4*Sdlamp;

end
```

```
function PZIPilu2 = PZIPilu2(V,Ui)
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu

PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
```

```
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;
%Iluminaic?n Tipo1:1kW , Tipo2: 4kW, Tipo3:8kW. Con lamparas descarga

PZIPilu2=16*Sdlamp;

end
```

```
function PZIPilu3 = PZIPilu3(V,Ui)
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde

%Estas potencias estan ensayadas para el nivel de tensi?n en pu

PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
```

```
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Iluminaic?n Tipo1:1kW , Tipo2: 4kW, Tipo3:8kW. Con lamparas descarga

PZIPilu3=32*Sdlamp;

end
```

```
function [ PZIPInd1ver ] = PZIPInd1ver( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA
```

```
Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

PZIPInd1ver=Smi1*2+5*Sdlamp+Sheater1;


end
```

```
function [ PZIPInd1ver ] = PZIPInd1ver( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;
V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
```

```
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

PZIPInd1ver=Smi1*2+5*Sdlamp+SAC;


end
```

```
function [ PZIPInd2inv ] = PZIPInd2inv( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);
for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
```

```
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

PZIPInd2inv=Smi2*3+7*Sdlamp+Sheater2;
end
```

```
function [ PZIPInd2ver ] = PZIPInd2ver( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA
```

```
Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

PZIPInd2ver=Smi2*3+7*Sdlamp+2*SAC;
end
```

```
function [ PZIPInd3inv ] = PZIPInd3inv( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;

V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA
```

```
Slamp=Plamp+i*Qlamp;
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

PZIPInd3inv=2*Smi2+Smi3*1+7*Sdlamp+2*Sheater2;
end
```

```
function [ PZIPInd3ver ] = PZIPInd3ver( V, Ui )
%Las cargas se modelan para cierto valor de voltage. El resultado son
%Ecuaciones de tipo: ax4+bx3+cx2+dx+e, en las que X=Ui-V, V=voltaje
%Los datos de las cargas se almacenaran en vectores de longitud 5:abdcde
PZIPLamp500=         [334.96; 150.18; 45]
PZIPDischLamp=       [313.9 ; -6.7874; -32.94]
PZIPMindtipo1=       [3260.2; -5052.1; 2885.4]
PZIPMindtipo2=       [3511.5; -5642  ; 4002.5]
PZIPMindtipo3=       [5978.5; -9639  ; 6652 ]
PZIPHeater1=         [1387.4; -494.93; 233.43]
PZIPHeater2=         [3401.4; -1196  ; 579.19 ]
PZIPResistor=        [641.83; -91.238; 30.63 ]
PZIPFlatScreenPC=    [-7.88 ; 13.653 ; 29.124 ]
PZIPAC=              [1.17       ; -1.83    ; 1.66   ]*496.33
PZIPFan=             [-0.47       ; 1.71    ; 0.24   ]*164
PZIPFridge=          [1.17       ; -1.83    ; 1.66   ]*120

QZIPLamp500=         [656.999 ; -1086.6 ; 492.82]
QZIPDischLamp=       [440.05 ; -624.47 ; 301.01]
QZIPMindtipo1=       [20969   ; -31960; 13049]
QZIPMindtipo2=       [21066  ; -33039  ; 13961]
QZIPMindtipo3=       [15603    ; -24002; 10494 ]
QZIPHeater1=         [1142.4   ; -1877 ; 811.02]
QZIPHeater2=         [727.54  ; -1082.7 ; 453.75 ]
QZIPResistor=        [0      ; 0  ; 0 ]
QZIPFlatScreenPC=    [14.185 ; -7.5609 ; 36.69 ]
QZIPAC=              [15.68       ; -27.15    ; 12.47   ]*125.94
QZIPFan=             [2.34       ; -3.12    ; 1.78   ]*83.28
QZIPFridge=          [7.07       ; -10.9    ; 4.87   ]*52.47

%Calculo de la potencia consumida dependiendo del nivel de voltaje


Pfridge=0;
PFan=0;
Plamp=0;
Pdlamp=0;
Pmi1=0;
Pmi2=0;
Pmi3=0;
Pheater1=0;
Pheater2=0;
Presistor=0;
PPC=0;
PAC=0;

Qfridge=0;
QFan=0;
Qlamp=0;
Qdlamp=0;
Qmi1=0;
Qmi2=0;
Qmi3=0;
Qheater1=0;
Qheater2=0;
Qresistor=0;
QPC=0;
QAC=0;
V=norm(V);

for b=1:1:3

    %Consumo de Potencia Activa
    Plamp=Plamp+PZIPLamp500(b,1)*((V/Ui)^(3-b));
    Pdlamp=Pdlamp+PZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Pmi1=Pmi1+PZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Pmi2=Pmi2+PZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Pmi3=Pmi3+PZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Pheater1=Pheater1+PZIPHeater1(b,1)*((V/Ui)^(3-b));
    Pheater2=Pheater2+PZIPHeater2(b,1)*((V/Ui)^(3-b));
    Presistor=Presistor+PZIPResistor(b,1)*((V/Ui)^(3-b));
    PPC=PPC+PZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    PAC=PAC+PZIPAC(b,1)*((V/Ui)^(3-b));
    PFan=PFan+PZIPFan(b,1)*((V/Ui)^(3-b));
    Pfridge=Pfridge+PZIPFridge(b,1)*((V/Ui)^(3-b));

    %Consumo de Potencia Reactiva
    Qlamp=Qlamp+QZIPLamp500(b,1)*((V/Ui)^(3-b));
    Qdlamp=Qdlamp+QZIPDischLamp(b,1)*((V/Ui)^(3-b));
    Qmi1=Qmi1+QZIPMindtipo1(b,1)*((V/Ui)^(3-b));
    Qmi2=Qmi2+QZIPMindtipo2(b,1)*((V/Ui)^(3-b));
    Qmi3=Qmi3+QZIPMindtipo3(b,1)*((V/Ui)^(3-b));
    Qheater1=Qheater1+QZIPHeater1(b,1)*((V/Ui)^(3-b));
    Qheater2=Qheater2+QZIPHeater2(b,1)*((V/Ui)^(3-b));
    Qresistor=Qresistor+QZIPResistor(b,1)*((V/Ui)^(3-b));
    QPC=QPC+QZIPFlatScreenPC(b,1)*((V/Ui)^(3-b));
    QAC=QAC+QZIPAC(b,1)*((V/Ui)^(3-b));
    QFan=QFan+QZIPFan(b,1)*((V/Ui)^(3-b));
    Qfridge=Qfridge+QZIPFridge(b,1)*((V/Ui)^(3-b));
end
b=1;

%       POTENCIA ACTIVA Y REACTIVA CONSUMIDA

Slamp=Plamp+i*Qlamp;
```

```
Sdlamp=Pdlamp+i*Qdlamp;
Smi1=Pmi1+i*Qmi1;
Smi2=Pmi2+i*Qmi2;
Smi3=Pmi3+i*Qmi3;
Sheater1=Pheater1+i*Qheater1;
Sheater2=Pheater2+i*Qheater2;
Sresistor=Presistor+i*Qresistor;
SPC=PPC+i*QPC;
SAC=PAC+i*QAC;
SFan=PFan+i*QFan;
Sfridge=Pfridge+i*Qfridge;

%Modelado te tipos de cargas/casas/comercios/industrias

PZIPInd3ver=2*Smi2+Smi3*1+7*Sdlamp+4*SAC;
end
```

Simulator of a Rural Distribution Network

This simulator is able to measure, within other things, the losses of a Rural Distribution Network considering the load profile with different models such as Constant Power Consumption, Constant Impedance or ZIP Model

Final Project
Bacherlor in Electromechanical Engineering
ICAI School of Engineering
22. May 2016

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=14; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(0.97+i*0.2431) %Factor de Potencia


%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(  1   ,   2   )=  0.02016 +i* 0.46116 ;
Z(  2   ,   3   )=  0.0036  +i* 0.08235 ;
Z(  3   ,   4   )=  0.00344 +i* 0.07869 ;
Z(  4   ,   5   )=  0.005128    +i* 0.0006  ;
Z(  5   ,   6   )=  0.005128    +i* 0.0006  ;
Z(  6   ,   7   )=  0.031409    +i* 0.003675;
Z(  7   ,   8   )=  0.077561    +i* 0.009075;
Z(  8   ,   9   )=  0.019871    +i* 0.002325;
Z(  9   ,   10  )=  0.029486    +i* 0.00345 ;
Z(  10  ,   11  )=  0.043588    +i* 0.0051  ;
Z(  11  ,   12  )=  0.031409    +i* 0.003675;
Z(  12  ,   13  )=  0.042947    +i* 0.005025;
Z(  13  ,   14  )=  0.049998    +i* 0.00585 ;

Z=Z*0.1;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPBinv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAinv(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBinv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCinv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ui,Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBinv(Ui,Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAinv(Ui,Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAinv(Ui,Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1(Ui,Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAinv(Ui,Ui)/Ucarga(14,1))




%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
```

```
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPBinv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAinv(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBinv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCinv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ui,Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBinv(Ui,Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAinv(Ui,Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAinv(Ui,Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1(Ui,Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAinv(Ui,Ui)/Ucarga(14,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*(Z(v-1,v))

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       --------------------------------

N=14; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       -------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       -----------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(  1   ,   2   )=  0.02016 +i* 0.46116 ;
Z(  2   ,   3   )=  0.0036  +i* 0.08235 ;
Z(  3   ,   4   )=  0.00344 +i* 0.07869 ;
Z(  4   ,   5   )=  0.005128    +i* 0.0006  ;
Z(  5   ,   6   )=  0.005128    +i* 0.0006  ;
Z(  6   ,   7   )=  0.031409    +i* 0.003675;
Z(  7   ,   8   )=  0.077561    +i* 0.009075;
Z(  8   ,   9   )=  0.019871    +i* 0.002325;
Z(  9   ,   10  )=  0.029486    +i* 0.00345 ;
Z(  10  ,   11  )=  0.043588    +i* 0.0051  ;
Z(  11  ,   12  )=  0.031409    +i* 0.003675;
Z(  12  ,   13  )=  0.042947    +i* 0.005025;
Z(  13  ,   14  )=  0.049998    +i* 0.00585 ;

Z=0.1*Z;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       -----------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPBinv(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPBinv(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPBinv(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPCinv(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPCinv(Ui,Ui)/Ui);
ImpedanciasZcte(9,1)=Ui/conj(PZIPBinv(Ui,Ui)/Ui);
ImpedanciasZcte(10,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(11,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(12,1)=Ui/conj(PZIPilu1(Ui,Ui)/Ui);
ImpedanciasZcte(13,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);
Icarga(1,9)=Ucarga(10,1)/ImpedanciasZcte(9,1);
Icarga(1,10)=Ucarga(11,1)/ImpedanciasZcte(10,1);
Icarga(1,11)=Ucarga(12,1)/ImpedanciasZcte(11,1);
Icarga(1,12)=Ucarga(13,1)/ImpedanciasZcte(12,1);
Icarga(1,13)=Ucarga(14,1)/ImpedanciasZcte(13,1);
```

```
%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);
Icarga(1,9)=Ucarga(10,1)/ImpedanciasZcte(9,1);
Icarga(1,10)=Ucarga(11,1)/ImpedanciasZcte(10,1);
Icarga(1,11)=Ucarga(12,1)/ImpedanciasZcte(11,1);
Icarga(1,12)=Ucarga(13,1)/ImpedanciasZcte(12,1);
Icarga(1,13)=Ucarga(14,1)/ImpedanciasZcte(13,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)
```

```
end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=14; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(  1   ,   2   )=  0.02016 +i* 0.46116 ;
Z(  2   ,   3   )=  0.0036  +i* 0.08235 ;
Z(  3   ,   4   )=  0.00344 +i* 0.07869 ;
Z(  4   ,   5   )=  0.005128    +i* 0.0006  ;
Z(  5   ,   6   )=  0.005128    +i* 0.0006  ;
Z(  6   ,   7   )=  0.031409    +i* 0.003675;
Z(  7   ,   8   )=  0.077561    +i* 0.009075;
Z(  8   ,   9   )=  0.019871    +i* 0.002325;
Z(  9   ,   10  )=  0.029486    +i* 0.00345 ;
Z(  10  ,   11  )=  0.043588    +i* 0.0051  ;
Z(  11  ,   12  )=  0.031409    +i* 0.003675;
Z(  12  ,   13  )=  0.042947    +i* 0.005025;
Z(  13  ,   14  )=  0.049998    +i* 0.00585 ;

Z=0.1*Z;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPBinv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAinv(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBinv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCinv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ucarga(9,1),Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBinv(Ucarga(10,1),Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAinv(Ucarga(11,1),Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAinv(Ucarga(12,1),Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1(Ucarga(13,1),Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAinv(Ucarga(14,1),Ui)/Ucarga(14,1))




%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);
```

```
%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPBinv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAinv(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBinv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCinv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ucarga(9,1),Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBinv(Ucarga(10,1),Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAinv(Ucarga(11,1),Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAinv(Ucarga(12,1),Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1(Ucarga(13,1),Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAinv(Ucarga(14,1),Ui)/Ucarga(14,1))


%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=14; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=0.95*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*() %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(  1   ,   2   )=  0.02016 +i* 0.46116 ;
Z(  2   ,   3   )=  0.0036  +i* 0.08235 ;
Z(  3   ,   4   )=  0.00344 +i* 0.07869 ;
Z(  4   ,   5   )=  0.005128    +i* 0.0006  ;
Z(  5   ,   6   )=  0.005128    +i* 0.0006  ;
Z(  6   ,   7   )=  0.031409    +i* 0.003675;
Z(  7   ,   8   )=  0.077561    +i* 0.009075;
Z(  8   ,   9   )=  0.019871    +i* 0.002325;
Z(  9   ,   10  )=  0.029486    +i* 0.00345 ;
Z(  10  ,   11  )=  0.043588    +i* 0.0051  ;
Z(  11  ,   12  )=  0.031409    +i* 0.003675;
Z(  12  ,   13  )=  0.042947    +i* 0.005025;
Z(  13  ,   14  )=  0.049998    +i* 0.00585 ;

Z=0.1*Z;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPBver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAver(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAver(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCver(Ui,Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBver(Ui,Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAver(Ui,Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAver(Ui,Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1(Ui,Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAver(Ui,Ui)/Ucarga(14,1))




%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);
```

```
%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPBver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAver(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAver(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCver(Ui,Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBver(Ui,Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAver(Ui,Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAver(Ui,Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1(Ui,Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAver(Ui,Ui)/Ucarga(14,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=14; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(  1   ,   2   )=  0.02016 +i* 0.46116 ;
Z(  2   ,   3   )=  0.0036  +i* 0.08235 ;
Z(  3   ,   4   )=  0.00344 +i* 0.07869 ;
Z(  4   ,   5   )=  0.005128    +i* 0.0006  ;
Z(  5   ,   6   )=  0.005128    +i* 0.0006  ;
Z(  6   ,   7   )=  0.031409    +i* 0.003675;
Z(  7   ,   8   )=  0.077561    +i* 0.009075;
Z(  8   ,   9   )=  0.019871    +i* 0.002325;
Z(  9   ,   10  )=  0.029486    +i* 0.00345 ;
Z(  10  ,   11  )=  0.043588    +i* 0.0051  ;
Z(  11  ,   12  )=  0.031409    +i* 0.003675;
Z(  12  ,   13  )=  0.042947    +i* 0.005025;
Z(  13  ,   14  )=  0.049998    +i* 0.00585 ;

Z=0.1*Z;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPBver(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPBver(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPBver(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPCver(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPCver(Ui,Ui)/Ui);
ImpedanciasZcte(9,1)=Ui/conj(PZIPBver(Ui,Ui)/Ui);
ImpedanciasZcte(10,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(11,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(12,1)=Ui/conj(PZIPilu1(Ui,Ui)/Ui);
ImpedanciasZcte(13,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);
Icarga(1,9)=Ucarga(10,1)/ImpedanciasZcte(9,1);
Icarga(1,10)=Ucarga(11,1)/ImpedanciasZcte(10,1);
Icarga(1,11)=Ucarga(12,1)/ImpedanciasZcte(11,1);
Icarga(1,12)=Ucarga(13,1)/ImpedanciasZcte(12,1);
Icarga(1,13)=Ucarga(14,1)/ImpedanciasZcte(13,1);
```

```
%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);
Icarga(1,9)=Ucarga(10,1)/ImpedanciasZcte(9,1);
Icarga(1,10)=Ucarga(11,1)/ImpedanciasZcte(10,1);
Icarga(1,11)=Ucarga(12,1)/ImpedanciasZcte(11,1);
Icarga(1,12)=Ucarga(13,1)/ImpedanciasZcte(12,1);
Icarga(1,13)=Ucarga(14,1)/ImpedanciasZcte(13,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end
```

```
PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=14; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=0.9*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(  1   ,   2   )=  0.02016 +i* 0.46116 ;
Z(  2   ,   3   )=  0.0036  +i* 0.08235 ;
Z(  3   ,   4   )=  0.00344 +i* 0.07869 ;
Z(  4   ,   5   )=  0.005128    +i* 0.0006  ;
Z(  5   ,   6   )=  0.005128    +i* 0.0006  ;
Z(  6   ,   7   )=  0.031409    +i* 0.003675;
Z(  7   ,   8   )=  0.077561    +i* 0.009075;
Z(  8   ,   9   )=  0.019871    +i* 0.002325;
Z(  9   ,   10  )=  0.029486    +i* 0.00345 ;
Z(  10  ,   11  )=  0.043588    +i* 0.0051  ;
Z(  11  ,   12  )=  0.031409    +i* 0.003675;
Z(  12  ,   13  )=  0.042947    +i* 0.005025;
Z(  13  ,   14  )=  0.049998    +i* 0.00585 ;

Z=0.1*Z;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPBverx(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAverx(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAverx(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBverx(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAverx(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBverx(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCverx(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCverx(Ucarga(9,1),Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBverx(Ucarga(10,1),Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAverx(Ucarga(11,1),Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAverx(Ucarga(12,1),Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1x(Ucarga(13,1),Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAverx(Ucarga(14,1),Ui)/Ucarga(14,1))




%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);
```

```
%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPBverx(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPAverx(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAverx(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPBverx(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAverx(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBverx(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPCverx(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCverx(Ucarga(9,1),Ui)/Ucarga(9,1))
Icarga(1,9)=conj(PZIPBverx(Ucarga(10,1),Ui)/Ucarga(10,1))
Icarga(1,10)=conj(PZIPAverx(Ucarga(11,1),Ui)/Ucarga(11,1))
Icarga(1,11)=conj(PZIPAverx(Ucarga(12,1),Ui)/Ucarga(12,1))
Icarga(1,12)=conj(PZIPilu1x(Ucarga(13,1),Ui)/Ucarga(13,1))
Icarga(1,13)=conj(PZIPAverx(Ucarga(14,1),Ui)/Ucarga(14,1))


%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

Simulator of an Urban Distribution Network

This simulator is able to measure, within other things, the losses of an Urban Distribution Network considering the load profile with different models such as Constant Power Consumption, Constant Impedance or ZIP Model

Final Project
Bacherlor in Electromechanical Engineering
ICAI School of Engineering
22. May 2016

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAinv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBinv(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCinv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCinv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAinv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ui,Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```matlab
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAinv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBinv(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCinv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCinv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAinv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ui,Ui)/Ucarga(9,1))
%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPBinv(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPCinv(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPCinv(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPilu2(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPBinv(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPCinv(Ui,Ui)/Ui);



%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);





%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
```

```
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```matlab
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAinv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBinv(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCinv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCinv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAinv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ucarga(9,1),Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAinv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBinv(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCinv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCinv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBinv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAinv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCinv(Ucarga(9,1),Ui)/Ucarga(9,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBver(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCver(Ui,Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBver(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCver(Ui,Ui)/Ucarga(9,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```matlab
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPBver(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPCver(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPCver(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPilu2(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPBver(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPCver(Ui,Ui)/Ui);



%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);





%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
```

```
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAver(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBver(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCver(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCver(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBver(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAver(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCver(Ucarga(9,1),Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAver(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPBver(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPCver(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPCver(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPBver(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPAver(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPCver(Ucarga(9,1),Ui)/Ucarga(9,1))


%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

Simulator of a Semi-Urban Distribution Network

This simulator is able to measure, within other things, the losses of a Semi-Urban Distribution Network considering the load profile with different models such as Constant Power Consumption, Constant Impedance or ZIP Model

Final Project
Bacherlor in Electromechanical Engineering
ICAI School of Engineering
22. May 2016

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAinv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ui,Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAinv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ui,Ui)/Ucarga(9,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPilu1(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPInd1inv(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPAinv(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPInd1inv(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPInd1inv(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPilu1(Ui,Ui)/Ui);



%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);





%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
```

```
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       --------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       -------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       -----------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAinv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1inv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1inv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1inv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ucarga(9,1),Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAinv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAinv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1inv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAinv(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1inv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1inv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ucarga(9,1),Ui)/Ucarga(9,1))


%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=0.9*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAver(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ui,Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAver(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ui,Ui)/Ucarga(9,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPilu1(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPInd1ver(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPAver(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPInd1ver(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPInd1ver(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPilu1(Ui,Ui)/Ui);



%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);





%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
```

```
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.1*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPAver(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAver(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1ver(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAver(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1ver(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1ver(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ucarga(9,1),Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPAver(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPilu1(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPAver(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd1ver(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPAver(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd1ver(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd1ver(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPilu1(Ucarga(9,1),Ui)/Ucarga(9,1))


%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

Simulator of an Industrial Distribution Network

This simulator is able to measure, within other things, the losses of an Industrial Distribution Network considering the load profile with different models such as Constant Power Consumption, Constant Impedance or ZIP Model

Final Project
Bacherlor in Electromechanical Engineering
ICAI School of Engineering
22. May 2016

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2inv(Ui,Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1inv(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3inv(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2inv(Ui,Ui)/Ucarga(9,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPInd3inv(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPInd1inv(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPInd3inv(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPInd3inv(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPilu2(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPInd3inv(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPInd3inv(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPInd2inv(Ui,Ui)/Ui);


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);





%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
```

```
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPInd3inv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1inv(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3inv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3inv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3inv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3inv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2inv(Ucarga(9,1),Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPInd3inv(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1inv(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3inv(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3inv(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3inv(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3inv(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2inv(Ucarga(9,1),Ui)/Ucarga(9,1))


%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2ver(Ui,Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1ver(Ui,Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ui,Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3ver(Ui,Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2ver(Ui,Ui)/Ucarga(9,1))

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=1.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(1) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz

Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;


% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------

ImpedanciasZcte(1,1)=Ui/conj(PZIPInd3ver(Ui,Ui)/Ui);
ImpedanciasZcte(2,1)=Ui/conj(PZIPInd1ver(Ui,Ui)/Ui);
ImpedanciasZcte(3,1)=Ui/conj(PZIPInd3ver(Ui,Ui)/Ui);
ImpedanciasZcte(4,1)=Ui/conj(PZIPInd3ver(Ui,Ui)/Ui);
ImpedanciasZcte(5,1)=Ui/conj(PZIPilu2(Ui,Ui)/Ui);
ImpedanciasZcte(6,1)=Ui/conj(PZIPInd3ver(Ui,Ui)/Ui);
ImpedanciasZcte(7,1)=Ui/conj(PZIPInd3ver(Ui,Ui)/Ui);
ImpedanciasZcte(8,1)=Ui/conj(PZIPInd2ver(Ui,Ui)/Ui);



%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);





%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
```

```
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=Ucarga(2,1)/ImpedanciasZcte(1,1);
Icarga(1,2)=Ucarga(3,1)/ImpedanciasZcte(2,1);
Icarga(1,3)=Ucarga(4,1)/ImpedanciasZcte(3,1);
Icarga(1,4)=Ucarga(5,1)/ImpedanciasZcte(4,1);
Icarga(1,5)=Ucarga(6,1)/ImpedanciasZcte(5,1);
Icarga(1,6)=Ucarga(7,1)/ImpedanciasZcte(6,1);
Icarga(1,7)=Ucarga(8,1)/ImpedanciasZcte(7,1);
Icarga(1,8)=Ucarga(9,1)/ImpedanciasZcte(8,1);

%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

```
clc
clear all
format short
disp('  ')
disp('  ')
echo on

%       Definicion Parametros de la linea
%       ---------------------------------

N=9; % N?mero de nudos que tiene nuestra l?nea
H=N-1;% Parametro utilizado para expresar que N nudos implican N-1 cargas
Ui=230;% Tension Nominal de la l?nea a la que se han referido las cargas
Tensionnivel=01.15*Ui; %Tension a la que se encuentra el primer punto
Tensioncontrol=Tensionnivel*(01+i*0) %Factor de Potencia

%       Definicion parametros iterativos
%       --------------------------------

Mistmacth=zeros(N,1); %Vector comparativo Mistmatch
errmaxMISMATCHES = max(abs(Mistmacth)); %Error m?ximo


%       Matriz Tensiones en cada punto
%       ------------------------------

Ucarga=zeros(N,1); %Inicializaci?n de el Vector que recoge tensiones nudo

%Inicializacion valores iniciales tensiones, todos a la Tension de Control

k=1;
for k=1:N
    Ucarga(k,1)=Tensioncontrol;
    k=k+1;
end
k=0;

%       Definicion matriz impedancias de la linea

Z=zeros(N,N); %Inicializacion de la matriz
Z(1,2)=0.0295+1i*0.0092;
Z(2,3)=0.0258+1i*0.0081;
Z(3,4)=0.0013+1i*0.0004;
Z(4,5)=0.0045+1i*0.0003;
Z(5,6)=0.0009+1i*0.0003;
Z(6,7)=0.0305+1i*0.0095;
Z(7,8)=0.0229+1i*0.0072;
Z(8,9)=0.0126+1i*0.0039;



% (La primera iteracion que incializa los valores se hace 'a mano' con el
% script que vamos a ver desde ahora hasta la presentaci?n del bucle)

%       C?lculo y Definici?n de las Intensidades de Carga y de L?nea
%       ------------------------------------------------------------


%   Iniciaclizacion Intensidades por las cargas
Icarga=zeros(1,H);

Icarga(1,1)=conj(PZIPInd3ver(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1ver(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3ver(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3ver(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3ver(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3ver(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2ver(Ucarga(9,1),Ui)/Ucarga(9,1))



%   Inicializacion intensidades por la linea
Ilinea=zeros(1,H)
h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);

%Resolucion iterativa  de flujo de carga
errmaxMISMATCHES=100; %valor para que entre en el bucle
toleranciaMISMATCHES = 0.01*Ui; %precision de la iteracion 1%
numITERACIONES = 1;
Uiteracion=zeros(N,1)
Unueva=zeros(N,1)

while (errmaxMISMATCHES >= toleranciaMISMATCHES)

% Calculamos las nuevas tensiones sabiendo las intensidades de linea
    s=2;
    Ucarga(1,1)=Tensioncontrol
    for s=2:1:N
```

```
       Ucarga(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end


Icarga(1,1)=conj(PZIPInd3ver(Ucarga(2,1),Ui)/Ucarga(2,1))
Icarga(1,2)=conj(PZIPInd1ver(Ucarga(3,1),Ui)/Ucarga(3,1))
Icarga(1,3)=conj(PZIPInd3ver(Ucarga(4,1),Ui)/Ucarga(4,1))
Icarga(1,4)=conj(PZIPInd3ver(Ucarga(5,1),Ui)/Ucarga(5,1))
Icarga(1,5)=conj(PZIPilu2(Ucarga(6,1),Ui)/Ucarga(6,1))
Icarga(1,6)=conj(PZIPInd3ver(Ucarga(7,1),Ui)/Ucarga(7,1))
Icarga(1,7)=conj(PZIPInd3ver(Ucarga(8,1),Ui)/Ucarga(8,1))
Icarga(1,8)=conj(PZIPInd2ver(Ucarga(9,1),Ui)/Ucarga(9,1))


%Inicializacion intensidades por la l?nea

h=1;
Ilinea(1,H)=Icarga(1,H)
Y=H-1;
for h=Y:-1:1
    Ilinea(1,h)=Ilinea(1,h+1)+Icarga(1,h);
end
Ilinea(1,H)=Icarga(1,H);


%Calculamos las nuevas tensiones con las nuevas Icargas

    k=0;

    Uiteracion(1,1)=Tensioncontrol;
    for s=2:1:N
       Uiteracion(s,1)=Ucarga(s-1,1)-Ilinea(1,s-1)*Z(s-1,s)
    end

    for k=1:1:N
        Mistmacth(k,1)=norm(Ucarga(k,1))-norm(Uiteracion(k,1));
    end

    errmaxMISMATCHES = max(abs(Mistmacth));

    % Si estamos aqui, es necesario hacer una iteracion mas
    numITERACIONES =numITERACIONES + 1 ;

    %Actualizamos el valor de la Ucarga
     for s=1:1:N
       Ucarga(s,1)=Uiteracion(s,1)
     end

    if numITERACIONES >= 100
        errmaxMISMATCHES =0;
    else
        errmaxMISMATCHES = max(abs(Mistmacth))
    end
end



%Ahora viene la parte chula, nos calculamos las perdidas originadas
Perdidas=zeros(H,1);
v=1;
for v=2:1:N
Perdidas(v-1,1)=conj((Ilinea(1,v-1)))*(Ilinea(1,v-1))*Z(v-1,v)

end

PerdidasTOT=sum(Perdidas);
PerdidasTOTn=norm(PerdidasTOT);

p=1;
for p=1:1:N
    Univel(p,1)=abs(Ucarga(p,1))
end
p=1;

disp(PerdidasTOT);
disp(PerdidasTOTn);
```

References

[Bravo et al., 2010] Bravo, R., Yinger, R., Chassin, D., Huang, H., Lu, N., Hiskens, I., and Venkataramanan, G. (2010). Final project report load modeling transmission research. *Lawrence Berkeley National Laboratory (LBNL).*

[Energi, 2008] Energi, D. (2008). Danish electricity supply 08 statistical survey. *danishenergyassociation*, page 60.

[García, 2006] García, A. M. (2006). Evaluación del impacto de la generación distribuida en la operación y planificación de las redes de distribución eléctrica. *Universidad Pontificia Comillas, Escuela Técnica Superior de Ingeniería.*

[He, 2011] He, S. (2011). *Modeling Power System Load Using Intelligent Methods.* PhD thesis, Kansas State University.

[Kim and Kim, 2012] Kim, B.-H. and Kim, H.-R. (2012). Measurement-based estimation of the composite load model parameters. *Journal of Electrical Engineering and Technology*, 7(6):845–851.

[Liu et al., 2002] Liu, J., Salama, M., and Mansour, R. (2002). An efficient power flow algorithm for distribution systems with polynomial load. *International Journal of Electrical Engineering Education*, 39(4):371–386.

[Méndez Quezada, 2005] Méndez Quezada, V. H. (2005). *Generación Distribuida: Aspectos técnicos y su tratamiento regulatorio.* PhD thesis, Tesis doctoral, Universidad Pontificia Comillas.

[Papachristos, 2014] Papachristos, G. (2014). Residential electricity consumption in the netherlands: A model-based policy analysis.

[Schneider et al., 2010] Schneider, K. P., Fuller, J., Tuffner, F., and Singh, R. (2010). Evaluation of conservation voltage reduction (cvr) on a national level. *Pacific Northwest National Laboratory report*.