

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**INCORPORACIÓN DE MODELOS DE CARGA EN ESTADO
ESTABLE PARA EL CÁLCULO DE FLUJOS DE POTENCIA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELÉCTRICO**

CRISTIAN JAVIER JIMÉNEZ SALAZAR

cristian.jimenez@epn.edu.ec

DIRECTOR: DR.-ING. ARCOS MARTÍNEZ HUGO NEPTALÍ

hugo.arcos@epn.edu.ec

Quito, Agosto 2018

AVAL

Certifico que el presente trabajo fue desarrollado por CRISTIAN JAVIER JIMÉNEZ SALAZAR bajo mi supervisión.

Dr.-Ing. Arcos Martínez Hugo Neptalí
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo CRISTIAN JAVIER JIMÉNEZ SALAZAR, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

CRISTIAN JAVIER JIMÉNEZ SALAZAR

DEDICATORIA

A Jaime, Yelanda y Oswaldo.

Cristian Javier

AGRADECIMIENTO

Quiero comenzar agradeciendo a Dios y a la Virgen Santísima por permitirme alcanzar una meta más en mi vida profesional.

A mis padres por su confianza, esfuerzo y apoyo incondicional para lograr culminar con éxitos esta etapa de mi vida.

A mi hermano Jaime por el ánimo y apoyo brindado, siendo mi inspiración para continuar cada día hacia adelante.

A mis maestros por compartirme lo mejor de sus conocimientos y brindarme su amistad.

Un especial agradecimiento a los doctores Dr. Hugo Arcos y Dr. Franklin Quilumba, por haberme dado la oportunidad de desarrollar este proyecto de titulación bajo su dirección. Gracias por la ayuda, motivación y el tiempo brindado durante la ejecución del proyecto, y sobretodo por su amistad.

Al Ing. Mauricio Soria por su ayuda y motivación durante el desarrollo de mi trabajo de titulación.

A mis amigos y compañeros de aula, por todos los momentos compartidos que quedan guardados en mi memoria.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VIII
ABSTRACT	IX
1. INTRODUCCIÓN.....	1
1.1 Objetivos	2
1.2 Alcance	2
1.3 Marco Teórico	3
1.3.1 Software Libre.....	3
1.3.1.1 Características del Software Libre.....	3
1.3.2 Python.....	4
1.3.2.1 Características de Python.....	4
1.3.3 Modelación de los Elementos de Red para el Cálculo de Flujos de Potencia.....	5
1.3.3.1 Barras.....	6
1.3.3.2 Generadores.....	6
1.3.3.3 Transformadores	7
1.3.3.4 Líneas de Transmisión	9
1.3.3.5 Compensación Reactiva.....	10
1.3.3.6 Motor de Inducción	11
1.3.3.6.1 Modelo de Jaula Individual (Single Cage Model)	11
1.3.3.6.2 Modelo de Jaula Individual con Desplazamiento de Corriente (Jaula de Ardilla).....	12
1.3.3.6.3 Modelo de Doble Jaula.....	13
1.3.3.6.4 Modelo de Doble Jaula con Desplazamiento de Corriente.....	14
1.3.3.7 Carga [5].....	14
1.3.4 Modelos de Carga.....	15
1.3.4.1 Modelo de Carga Dinámico	16
1.3.4.2 Modelo de Carga en Estado Estable	16

1.3.4.2.1	Modelo de Carga Exponencial	16
1.3.4.2.2	Modelo de Carga Polinomial	18
1.3.4.2.3	Modelo de Carga Lineal	19
1.3.4.2.4	Modelo de Carga Completo.....	20
1.3.4.2.5	Modelo de Carga del Motor de Inducción.....	21
1.3.4.2.6	Modelo de Carga con Interfaz de Electrónica de Potencia.....	22
2.	METODOLOGÍAS PARA LA INCORPORACIÓN DE MODELOS DE CARGA EN EL CÁLCULO DE FLUJOS DE POTENCIA	24
2.1	Introducción.....	24
2.2	Método de Newton Raphson (N-R) para el cálculo de flujos de potencia	24
2.2.1	Tipos de Barras.....	24
2.2.2	Ecuaciones de Red.....	25
2.2.3	Ecuaciones de Inyección de Potencia	26
2.2.4	Errores de Potencia	27
2.2.5	Formulación del Algoritmo de Newton Raphson.	28
2.2.6	Aplicación del algoritmo de Newton Raphson al cálculo de Flujos de Potencia.....	31
2.2.6.1	Cálculo de los elementos del Jacobiano.....	32
2.2.6.1.1	Cálculo de los elementos fuera de la diagonal $i \neq j$	32
2.2.6.1.2	Cálculo de los elementos de la diagonal ($i=j$)	33
2.2.7	Diagrama de flujo del algoritmo de Newton Raphson aplicado a la solución de Flujos de Potencia	34
2.3	Metodologías para la Incorporación de Modelos de Carga en Estado Estable al Cálculo de Flujos de Potencia	36
2.3.1	Metodología de Inyección Nodal.....	36
2.3.1.1	Diagrama de flujo.....	37
2.3.2	Metodología de Flujos Sucesivos	39
2.3.2.1	Diagrama de Flujo	40
2.3.3	Metodología de Reestructuración de Potencias Calculadas.....	42
2.3.3.1	Diagrama de Flujo	43
3.	APLICACIÓN, ANÁLISIS Y RESULTADOS	45
3.1	Introducción.....	45
3.2	Ejemplo Aplicado a un Sistema de Dos Barras.....	45
3.2.1	Ejemplo de Cálculo del Flujo de Potencia sin considerar la Incorporación de modelos de carga.....	45

3.2.2	Ejemplo de Cálculo del Flujo de Potencia considerando la Incorporación de Modelos de Carga.....	51
3.2.2.1	Metodología para agrupar cargas.....	51
3.2.2.2	Metodología para agrupar motores de inducción.....	55
3.2.2.3	Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación de Modelos de Carga mediante la Metodología de Inyección Noda.....	57
3.2.2.4	Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación de Modelos de Carga mediante la Metodología de Flujos Sucesivos.....	64
3.2.2.5	Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación de Modelos de Carga mediante la Metodología de Reestructuración de Potencias Calculadas.....	71
3.2.2.6	Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación del Modelo de Carga de Motor de Inducción.....	76
3.3	Presentación de los Sistemas de Prueba.....	83
3.4	Resultados obtenidos al Incorporar Modelos de Carga en estado estable al cálculo de Flujos Potencia.....	87
3.4.1	Resultados obtenidos en los sistemas de 2 barras.....	87
3.4.2	Resultados obtenidos en los sistemas de 4 barras.....	93
3.4.3	Resultados obtenidos en el sistema de 14 barras.....	96
3.4.4	Resultados obtenidos en el sistema de 30 barras.....	105
3.4.5	Análisis de Resultados.....	120
4.	CONCLUSIONES Y RECOMENDACIONES.....	123
4.1	Conclusiones.....	123
4.2	Recomendaciones.....	124
5.	REFERENCIAS BIBLIOGRÁFICAS.....	125
6.	ANEXOS.....	127
	ANEXO I. Práctica Propuesta para el Laboratorio de Introducción a Sistemas Eléctricos de Potencia (IEE602).....	127
	ANEXO II. Parámetros de los Modelos de Carga utilizados en las Simulaciones.....	131
	ANEXO III. Sistemas de Prueba Modelados en DigSILENT PowerFactory y PSAT.....	135
	ANEXO IV. Descripción y Ejecución del Código desarrollado en Python.....	140
	ANEXO V. Archivos Desarrollados en Excel para Agrupar Cargas.....	158
	ORDEN DE EMPASTADO.....	160

RESUMEN

El presente trabajo de titulación establece una herramienta computacional para la incorporación de modelos de carga en estado estable en la metodología de Newton Raphson utilizada para resolver flujos de potencia. Para incorporar los modelos de carga en estado estable como función del voltaje en la solución del flujo de potencia, se utilizan tres metodologías: 1) Inyección Nodal, 2) Flujos Sucesivos y 3) Reestructuración de Potencias Calculadas, cada metodología permite modelar la carga dentro del flujo de potencia mediante cuatro modelos en estado estable: 1) Polinomial, 2) Exponencial, 3) Lineal y 4) Motor de Inducción.

El modelamiento matemático de la carga se realiza en un programa computacional escrito en Python 3.5.2, el cual es un software libre de código abierto muy completo en la resolución de problemas basados en algoritmos.

La validación de los resultados obtenidos en el programa se realiza mediante la comparación de los mismos, con resultados obtenidos en softwares comerciales destinados al análisis y simulación de sistemas eléctricos de potencia como PSAT y DlgSILENT PowerFactory.

Las redes eléctricas de prueba utilizadas son las estandarizadas IEEE de 14 barras, IEEE de 30 barras y el sistema de 4 barras del libro de Grainger y Stevenson, ya que sus resultados del cálculo de flujos de potencia se encuentran publicados en fuentes confiables y sirven de referencia para realizar una comparación cuantitativa con los resultados arrojados por el programa.

Además del desarrollo del código, se propone una práctica para el laboratorio de Introducción a Sistemas Eléctricos de Potencia (IEE602) de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional.

Palabras Claves: Flujo de Potencia, Modelos de Carga, Newton – Raphson, Sistemas de Potencia.

ABSTRACT

The following research thesis work corroborates a computational tool for the incorporation of load models in stable states in the Newton Raphson methodology used to solve power flows. To incorporate load models in stable state as voltage functions to solve power flows, three methodologies are used: 1) Node Injection, 2) Successive Flows and 3) Restructuring of Calculated Powers, each methodology allows for the modeling of the load within the power flow by means of four models in a stable state: 1) Polynomial, 2) Exponential, 3) Linear and 4) Induction motor.

The mathematical modeling of the load is done using a computer program written in Python 3.5.2, which is a free software offering very complete open access to solve problems based on algorithms.

The validation of the results obtained using the above program is done by comparing them with the results obtained using commercial software that analyses and simulates power electric systems such as PSAT and DlgSILENT PowerFactory.

The electrical test networks that were used are the standardized IEEE 14-bus, IEEE 30-bus and IEEE 4-bus systems from the book published by Grainger and Stevenson since the results for calculating power flows are public and serve as a reliable reference to carry out quantitative comparisons with the results given by the Python 3.5.2 program.

Besides developing the code, a practicum is proposed for the Introduction to Electrical Power Systems Laboratory (IEE6O2) course given at the School of Electrical and Electronic Engineering of the Escuela Politécnica Nacional (National Polytechnic School).

Key Words: Power Flow, Load Models, Newton - Raphson, Power Systems.

1. INTRODUCCIÓN

En el estudio y análisis de sistemas eléctricos se ha otorgado mucha atención al análisis y modelación de la generación y transmisión, sin embargo se ha prestado menos atención al comportamiento de las cargas aun cuando éstas tienen un impacto significativo en el rendimiento del sistema eléctrico de potencia.

La carga es uno de los elementos más importantes del sistema eléctrico de potencia y su adecuada modelación permite obtener mejores resultados. En nuestro país el operador del sistema en sus estudios en estado estable representa a la carga mediante el modelo de potencia constante, aspecto que no está desapegado de lo que sucede a nivel mundial como se muestra en la Figura 1.1, donde de 160 operadores encuestados solo el 20% incorpora distintos modelos de carga en estado estable a sus simulaciones y estudios [1].

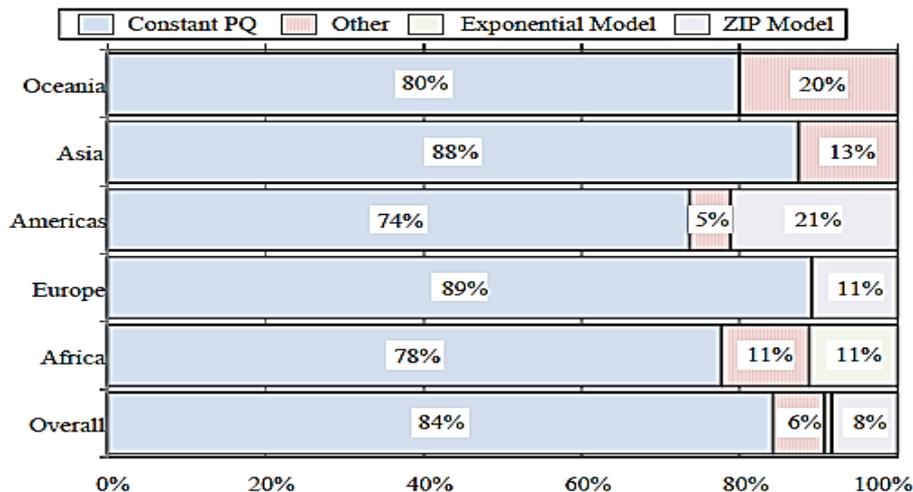


Figura 1.1. Modelos de carga en estado estable utilizados a nivel mundial [1].

De las herramientas computacionales dedicadas para el análisis y simulación de sistemas de potencia, el cálculo de flujos de potencia es probablemente la de mayor uso, por cuanto permite el análisis del sistema en diferentes etapas, como la planificación, diseño y operación. El objetivo de este proyecto de titulación es incorporar los modelos de carga en estado estable al cálculo de flujos de potencia mediante el método de Newton Raphson, para lo cual se desarrolla una herramienta computacional de ingeniería basada en un lenguaje de programación de código abierto que permite el análisis de la influencia en estado estable del comportamiento de la carga en las redes eléctricas.

En la cátedra de Introducción a Sistemas Eléctricos de Potencia, dictada en la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, los estudiantes

conocen cómo modelar los elementos del sistema eléctrico de potencia, sin embargo el p nsun y las fuentes bibliogr ficas de consulta modelan la carga considerando el modelo de potencia constante. En el presente trabajo de titulaci n se crea una herramienta computacional que ayudar  a los nuevos estudiantes y profesores de la asignatura de Introducci n a Sistemas El ctricos de Potencia en el estudio y an lisis de la modelaci n de carga en estado estable y su influencia en las redes el ctricas. De la misma manera la herramienta computacional podr  ser utilizada por el operador del sistema contribuyendo a mejorar los resultados de sus estudios en estado estable.

1.1 Objetivos

1.1.1 Objetivo General

Desarrollar un programa computacional para incorporar modelos de carga en estado estable al algoritmo de c lculo de flujos de potencia mediante el m todo de Newton Raphson.

1.1.2 Objetivos Espec ficos

- a) Estudiar los tipos de carga en estado estable mediante los modelos exponencial, polinomial, lineal y de motor de inducci n.
- b) Establecer diferentes metodolog as para incorporar modelos de carga en estado estable dentro del c lculo de flujos de potencia mediante el m todo de Newton Raphson.
- c) Identificar y analizar las variables a modificar dentro del algoritmo de c lculo de flujos de potencia mediante el m todo Newton Raphson para incorporar modelos de carga en estado estable en funci n del voltaje.
- d) Comparar los resultados y establecer diferencias al calcular flujos de potencia con y sin la incorporaci n de modelos de carga.
- e) Aplicar y validar los modelos desarrollados en redes el ctricas estandarizadas.

1.2 Alcance

El presente trabajo de titulaci n consiste en incorporar mediante tres metodolog as: 1) Inyecci n Nodal, 2) Flujos Sucesivos y 3) Reestructuraci n de Potencias Calculadas, cuatro modelos de carga en estado estable en funci n del voltaje 1) exponencial, 2) polinomial, 3) lineal y 4) motor de inducci n, dentro de la soluci n del flujo de potencia de Newton Raphson, comparar los resultados entre el flujo de potencia con modelo de

carga de potencia constante y el flujo de potencia con la incorporación de los nuevos modelos de carga, establecer diferencias existentes y emitir conclusiones en base a los resultados que se obtengan.

La plataforma de programación será PYTHON versión 3.5.2, el cual es un software libre muy completo en la resolución de problemas basados en algoritmos, con este paquete computacional se desarrollará un código para resolver la problemática propuesta.

Para validar los resultados de las metodologías utilizadas, se utilizarán diversas redes de prueba estandarizadas como: la IEEE de 14 barras y la IEEE de 30 barras, cuyos resultados del cálculo de flujos de potencia se encuentran publicados en fuentes confiables y sirven de referencia para realizar una comparación cuantitativa con los resultados arrojados por el programa.

De la misma manera, estas redes serán modeladas en paquetes comerciales como DlgSILENT PowerFactory y PSAT. De esta manera, se puede comparar los resultados del programa con aquellos generados en este tipo de software incluyendo modelos de carga.

Finalmente, se planteará la realización de una práctica para el laboratorio de Introducción a Sistemas Eléctricos de Potencia (IEE602).

1.3 Marco Teórico

En esta sección se presenta una introducción al software utilizado para programar la herramienta computacional destinada a resolver la problemática abordada en el presente trabajo de titulación, se enlistan algunas características importantes del software de programación y se describen las razones por las cuales fue elegido. Finalmente se aborda el modelamiento matemático en estado estable de los elementos constitutivos de una red eléctrica especialmente en lo concerniente al cálculo de flujos de potencia y se realiza una descripción completa en lo que respecta a modelación de carga.

1.3.1 Software Libre

El software libre o “free software” en inglés, es aquel que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Para poder estudiar y modificar un software libre este debe permitir el acceso a su código fuente [2].

1.3.1.1 Características del Software Libre

Las cuatro características o libertades que debe poseer un software para ser considerado como libre son las siguientes:

- El usuario debe tener la libertad de usar el programa, con cualquier propósito (privado, educativo, público, comercial, militar, etc.)
- El usuario debe tener la libertad de estudiar cómo funciona el programa, y adaptarlo para resolver las necesidades por las cuales lo eligió, es decir el usuario debe tener acceso al código fuente.
- El usuario debe tener la libertad de distribuir copias a cualquier persona.
- El usuario debe tener la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

1.3.2 Python

Python es un lenguaje de programación de alto nivel creado en Holanda por Guido Van Rossum a principios de los años 90, es un software libre interpretador de instrucciones que a diferencia de otros programas permite ir probando los códigos desarrollados sin la necesidad de que estos estén finalizados en su totalidad.

La razones para utilizar Python ante otros softwares son diversas, entre las más importantes se pueden mencionar que es un software libre, portable, fácil de entender y utilizar, que contiene varias librerías y permite construir funciones y aplicaciones propias.

1.3.2.1 Características de Python

Entre las características más importantes de Python se tienen que es un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos [3].

- *Lenguaje Interpretado o de script*

Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar todo el código, Python permite ejecutar y visualizar como se van ejecutando las distintas partes de un código.

- *Tipado dinámico*

La característica de tipado dinámico de Python permite obviar la declaración del tipo de variable, ya que el intérprete de Python reconoce el tipo durante la ejecución del programa en base a los valores que se le hayan asignado a la variable.

- *Fuertemente Tipado*

Python no permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente.

- *Multiplataforma*

El intérprete de Python está disponible para distintas plataformas como: Windows, Linux, Ubuntu entre otras, por lo que si no se utilizan librerías específicas de cada plataforma, un programa podrá correr en todos estos sistemas sin grandes cambios.

- *Orientado a Objetos*

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

1.3.3 Modelación de los Elementos de Red para el Cálculo de Flujos de Potencia

El cálculo de flujos de potencia en estado estable tiene como objetivo establecer el valor de la magnitud, ángulo de voltaje y valores de las potencias activa y reactiva en las barras del sistema.

El sistema de potencia está conformado por varios elementos interconectados entre sí, cuyo objetivo es abastecer a la carga desde la generación pasando por la transmisión. Entre los elementos más comunes que pueden ser encontrados en cualquier red se tienen:

- ✓ Barras.
- ✓ Generadores.
- ✓ Transformadores.
- ✓ Líneas de Transmisión.
- ✓ Compensación reactiva.
- ✓ Motores de Inducción.
- ✓ Carga.

En las secciones presentadas a continuación se explicará el modelamiento matemático de los elementos de red citados.

1.3.3.1 Barras

Las barras o nodos eléctricos son los elementos más simples de la red, sin embargo son de mucha importancia ya que en estas van conectados los demás elementos eléctricos del sistema. Las barras no tienen un modelo matemático específico, pero juegan un papel muy importante dentro del cálculo de flujos de potencia ya que dependiendo de la generación conectada a ella se puede definir tipos de barra.

1.3.3.2 Generadores

En estudios de estado estable los generadores son la fuente de potencia activa y/o reactiva de la red eléctrica, por lo general vienen representados por máquinas sincrónicas.

La principal función de un generador es la de entregar una potencia activa de generación constante P_{Gi} manteniendo regulado el voltaje de la barra (V_i) a la que se encuentra conectado mediante inyección o consumo de potencia reactiva. A la barra asociada a este tipo de generador se la conoce como PV ya que se conocen la potencia de generación y la magnitud del voltaje. En la práctica casi todos los generadores actúan como PV ya que cuentan con un regulador automático de voltaje (AVR) y despachan un valor fijo de potencia activa programado de acuerdo a la demanda.

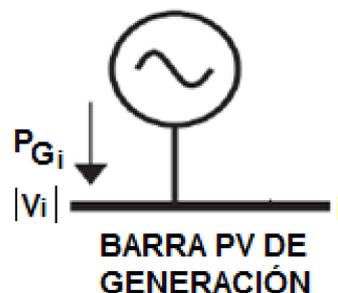


Figura 1.2. Barra PV de generación de un sistema de potencia.

Uno de los generadores, por lo general el más robusto es decir el de mayor capacidad y de más alta inercia, es definido como generador slack u oscilante, el voltaje de la barra (V_i) asociada a este generador es controlado por el sistema de regulación y el valor del mismo se fija para compensar las condiciones operativas del sistema. El valor del ángulo (θ_i) de la barra asociada a este generador sirve para referenciar el resto de ángulos razón por la cual normalmente es considerado 0° , además el conjunto barra - generador se conoce también como $V\theta$.

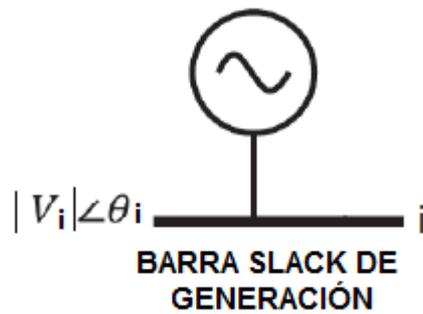


Figura 1.3. Barra slack u oscilante de generación de un sistema de potencia.

Adicionalmente, un generador puede ser modelado mediante una barra PQ de generación donde la potencia activa de generación P_{Gi} y la potencia reactiva de generación Q_{Gi} son conocidas o especificadas. En este modelo el voltaje de la barra depende de la máxima potencia reactiva que el generador pueda entregar.

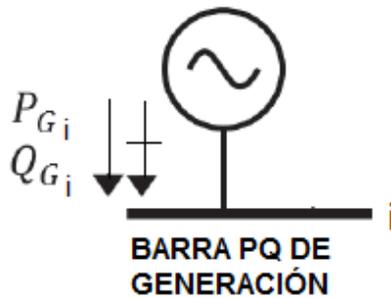


Figura 1.4. Barra PQ de generación de un sistema de potencia.

Cabe indicar que la barra de generación recibe el nombre de PQ, PV o Slack según la característica del generador conectada a ella.

1.3.3.3 Transformadores

La función principal del transformador dentro del sistema de potencia es elevar o disminuir el nivel de voltaje para satisfacer las condiciones de transmisión o de carga. El modelo utilizado para representar al transformador es el de dos devanados con un cambiador de tomas bajo carga, en este modelo se puede despreciar la rama de magnetización del circuito equivalente cuando se trabaja en estado estable ya que la corriente de excitación es mucho menor a la corriente de carga [3].

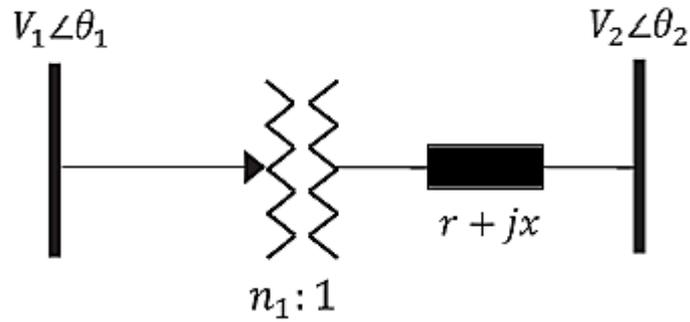


Figura 1.5. Transformador de dos devanados con cambiador de taps de un lado.

Donde:

n_1 : Valor en por unidad del cambiador de taps.

$r + jx$: Resistencia y reactancia de primario y secundario referidos al lado de bajo voltaje.

El cambiador de taps obliga a modelar al transformador mediante un modelo π de admitancias indicado en la Figura 1.6, en el cual la admitancia viene dada por la siguiente ecuación:

$$y = \frac{1}{n_1^2 * (r + jx)}$$

Ecuación 1.1. Admitancia en el modelo π de un transformador con cambiador de taps.

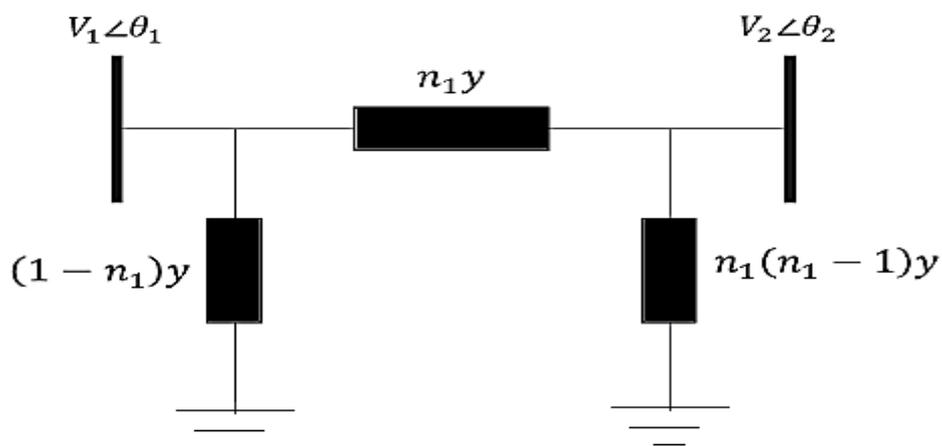


Figura 1.6. Modelo π de un transformador con cambiador de taps de un lado.

1.3.3.4 Líneas de Transmisión

Las líneas de transmisión son el medio físico mediante el cual se evacua la energía eléctrica desde las centrales de generación hasta la carga. Para el cálculo de flujos de potencia en estado estable se considera el modelo π mostrado en la Figura. 1.7.

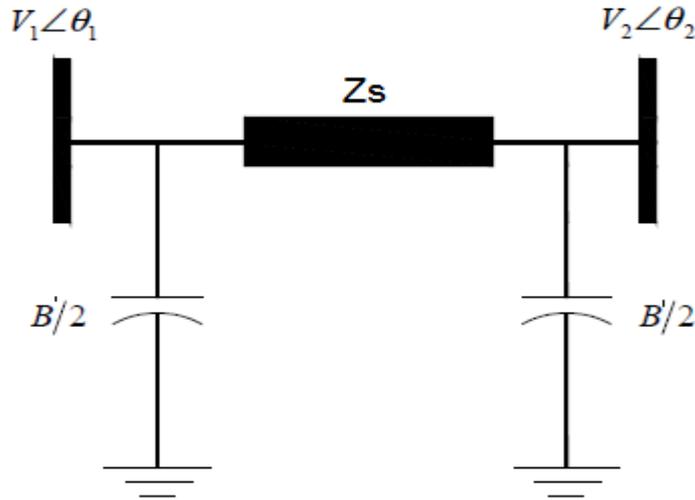


Figura 1.7. Modelo π de una línea de transmisión.

Donde Z_S es la impedancia serie de la línea y B' es la susceptancia total de la línea.

Para líneas de transmisión cortas (longitud hasta 80 km) y medias (longitud entre 80 km y 270 km) se cumple:

$$Z_S = l * (r_{LT} + jx_{LT}) = (R_{LT} + jX_{LT})$$

Ecuación 1.2. Impedancia serie de líneas de transmisión cortas y medias.

$$B' = b * l$$

Ecuación 1.3. Susceptancia total de líneas de transmisión cortas y medias.

Para líneas de transmisión largas (longitud mayor a 270 km) los parámetros Z_S y B' se calculan de la siguiente manera:

$$Z_S = (r_{LT} + jx_{LT}) * \frac{\sinh(\gamma l)}{\gamma l}$$

Ecuación 1.4. Impedancia serie de líneas de transmisión largas.

$$\frac{B'}{2} = \frac{b}{2} * \frac{\tanh(\frac{\gamma l}{2})}{\frac{\gamma l}{2}}$$

Ecuación 1.5. Suceptancia de líneas de transmisión largas.

$$\gamma l = l * \sqrt{b * (r_{LT} + jx_{LT})}$$

Ecuación 1.6. Constante de propagación.

Donde:

l : Longitud de la línea de transmisión.

r_{LT} : Resistencia del conductor por unidad de longitud.

x_{LT} : Reactancia del conductor por unidad de longitud.

R_{LT} : Resistencia total del conductor.

X_{LT} : Reactancia total del conductor.

b : Suceptancia de la línea por unidad de longitud.

1.3.3.5 Compensación Reactiva

La compensación reactiva está representada por los elementos conectados en paralelo a una barra cuya función es la de compensar la potencia reactiva mediante el uso de reactores (compensación inductiva) o capacitores (compensación capacitiva). El modelamiento de la compensación reactiva se realiza mediante una impedancia conectada a tierra cuya admitancia es:

$$y_{sh} = g_{sh} + jb_{sh}$$

Ecuación 1.6. Modelamiento de la compensación reactiva.

Donde:

y_{sh} : Admitancia equivalente en por unidad.

g_{sh} : Conductancia equivalente en por unidad.

b_{sh} : Suceptancia equivalente en por unidad.

Cabe señalar que esta admitancia debe ser considerada dentro del cálculo de la matriz de admitancias Y_{barra} tomando en cuenta la ubicación del elemento de compensación

reactiva. En caso de existir compensación reactiva tanto inductiva como capacitiva en una misma barra esta debe ser calculada como un equivalente de ambas.

1.3.3.6 Motor de Inducción

El modelo de motor de inducción puede ser representado mediante el modelo de máquina asincrónica indicado en el siguiente circuito equivalente:

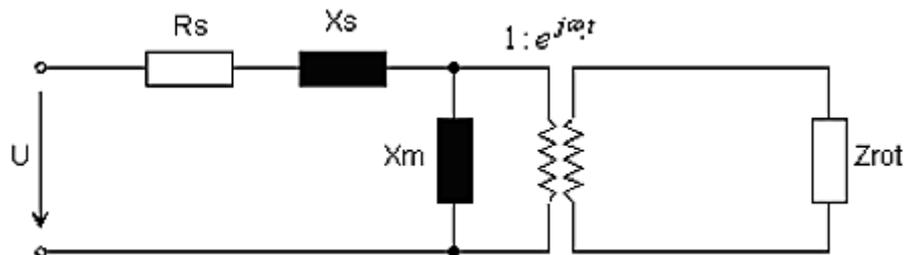


Figura 1.8. Circuito equivalente del motor de inducción [4].

Donde:

U : Voltaje de alimentación del motor de inducción.

R_s : Resistencia del estator.

X_s : Reactancia del estator.

X_m : Reactancia de magnetización.

Z_{rot} : Impedancia del rotor dependiendo del modelo.

Los modelos de impedancia del rotor varían en detalle y complejidad y deben seleccionarse cuidadosamente según los tipos de estudios que se realizarán y la disponibilidad de los datos de entrada.

Los modelos de impedancia del rotor se describen a continuación:

1.3.3.6.1 Modelo de Jaula Individual (Single Cage Model)

El modelo de jaula individual es el más simple de los modelos de impedancia del rotor, se caracteriza por una única rama R-L con resistencia del rotor dependiente del deslizamiento.

En este modelo, la impedancia del rotor se refiere al lado del estator para que la relación $1:e^{j\omega t}$ sea 1:1. La ventaja de este modelo es su simplicidad, particularmente cuando los datos del fabricante están incompletos o no están disponibles.

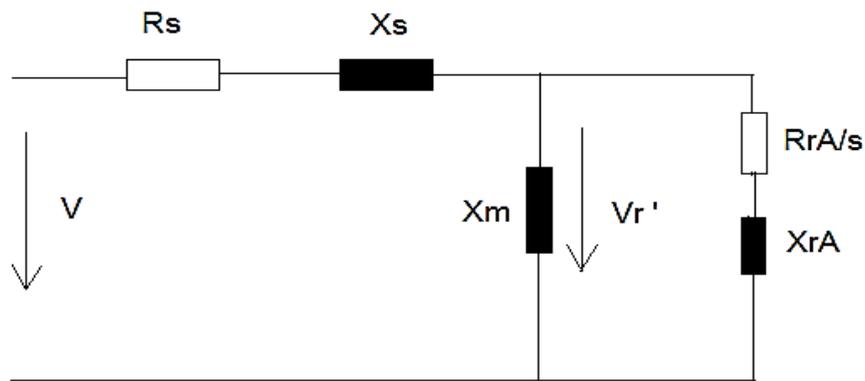


Figura 1.9. Modelo de jaula individual.

Donde:

R_{rA} : Resistencia del rotor en por unidad.

X_{rA} : Reactancia del rotor en por unidad.

s : Deslizamiento.

V : Voltaje del estator.

V_r' : Voltaje del rotor.

1.3.3.6.2 Modelo de Jaula Individual con Desplazamiento de Corriente (Jaula de Ardilla)

Este modelo se utiliza para representar motores que trabajan a velocidades bajas, es decir que operan con deslizamientos elevados. El efecto de desplazamiento de corriente se puede modelar agregando una rama R-L adicional en paralelo al rotor de jaula individual. La rama R-L adicional (R_{rA2}/s y X_{rA2}) se modela para representar el rotor de la jaula de ardilla en el arranque, donde predomina la reactancia de fuga del rotor. A medida que la velocidad aumenta, la influencia de la rama R-L adicional disminuye [4].

R_{rA0} y X_{rA0} mostrados en la Figura 1.10 representan una impedancia de fuga del rotor independiente del deslizamiento, la cual debe ser definida por el usuario para proporcionar una flexibilidad al modelo. Cabe indicar que por defecto, la impedancia de fuga del rotor se establece en cero.

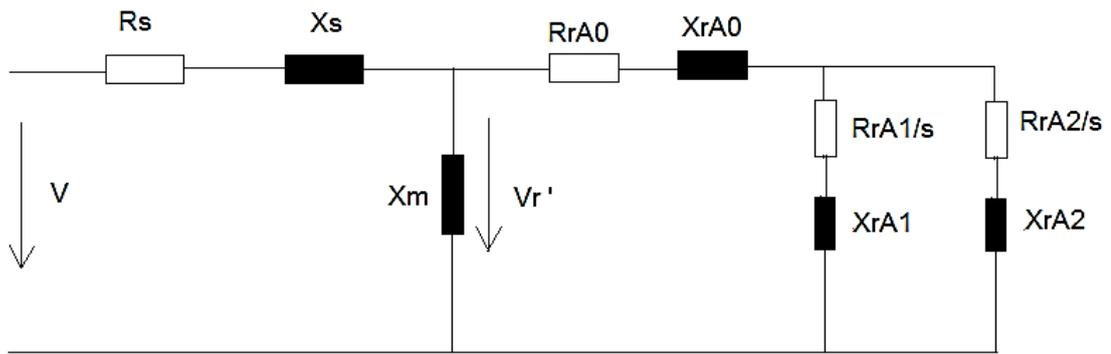


Figura 1.10. Modelo de jaula individual con desplazamiento de corriente.

Donde:

R_{rA0} y X_{rA0} : Resistencia y reactancia definidas por el usuario para agregar flexibilidad al modelo.

R_{rA1} y X_{rA1} : Resistencia y Reactancia del rotor.

R_{rA2} y X_{rA2} : Resistencia y Reactancia utilizadas para representar el desplazamiento de la corriente.

s: Deslizamiento.

1.3.3.6.3 Modelo de Doble Jaula

El rotor de doble jaula se modela de forma similar al rotor de jaula de ardilla con dos ramas R-L paralelas que representan las jaulas externa (A) e interna (B). La diferencia clave es que en el modelo de doble jaula se incluye una reactancia de fuga del rotor X_{rm} en lugar de la impedancia independiente del deslizamiento (R_{rA0} y X_{rA0}) [4].

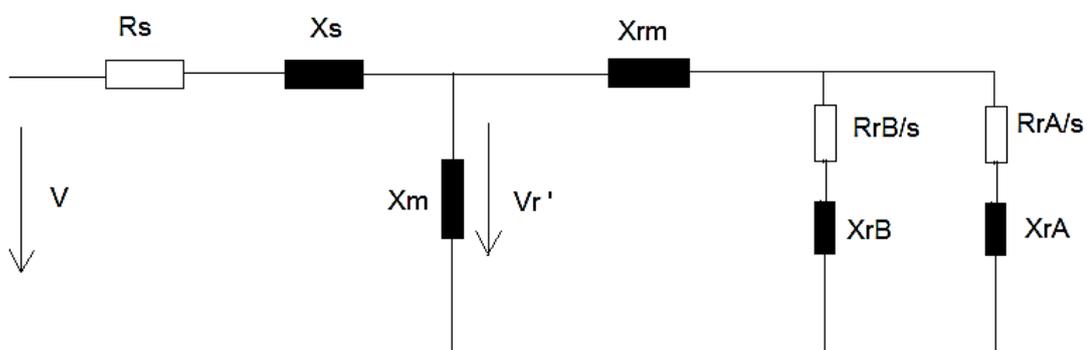


Figura 1.11. Modelo de doble jaula.

1.3.3.6.4 Modelo de Doble Jaula con Desplazamiento de Corriente

El modelo de rotor con doble jaula con desplazamiento de corriente se usa para representar un rotor de doble jaula de barra profunda, donde la jaula externa muestra el desplazamiento de corriente en el arranque [4]. En total, la dependencia de la frecuencia de la impedancia del rotor se puede aproximar por hasta tres ramas R-L paralelas como se muestra en la Figura 1.12.

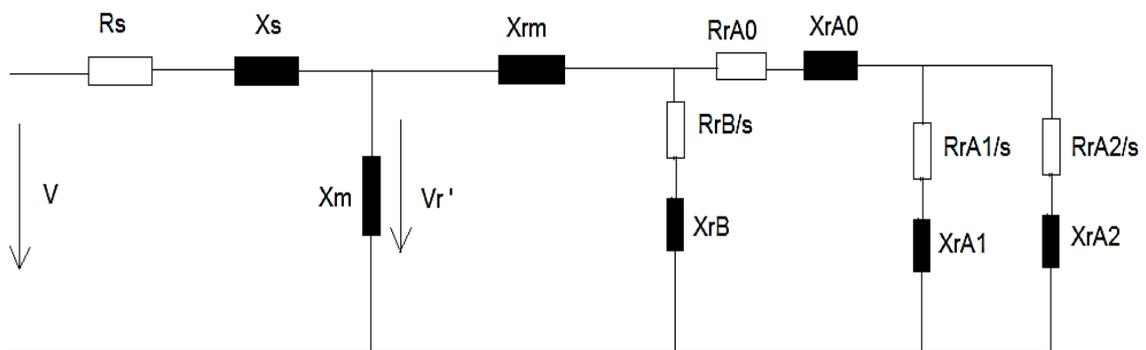


Figura 1.12. Modelo de doble jaula con desplazamiento de corriente.

Para la realización del presente proyecto se utilizará el modelo de motor de inducción de jaula individual, ya que su circuito equivalente es el que más se asemeja al circuito utilizado para definir el modelo de carga.

1.3.3.7 Carga [5]

La carga puede tener varios conceptos dentro del sistema eléctrico de potencia, entre los que más se ajustan a su definición se tienen:

- ✓ Un dispositivo conectado a un sistema de potencia que consume energía eléctrica.
- ✓ La potencia total (activa y/o reactiva) consumida por todos los dispositivos conectados a un sistema eléctrico de potencia.
- ✓ Una parte del sistema eléctrico de potencia que no está explícitamente representada en un modelo de sistema, sino que se trata como si fuera un único dispositivo de consumo de energía conectado a una barra en un modelo de sistema.
- ✓ La potencia de salida de un generador o planta generadora.

1.3.4 Modelos de Carga

Un modelo de carga es la representación analítica y matemática basada en un circuito equivalente o en un componente físico que puede ser utilizado para el análisis, la predicción o estimación de las características de carga pertinentes en los estudios de sistemas de potencia [1].

Los modelos de carga pueden representarse tanto en estado estable como en estado dinámico, dependiendo del tipo de estudio que se esté realizando.

Los modelos de carga frecuentemente usados para estudios en estado estable y dinámico son los siguientes: [1]

Modelos de carga en estado estable:

- ✓ Modelo de carga Exponencial.
- ✓ Modelo de carga Polinomial.
- ✓ Modelo de carga Lineal.
- ✓ Modelo de carga Completo.
- ✓ Modelo de carga del Motor de Inducción.
- ✓ Modelo de carga con Interfaz de Electrónica de Potencia.

Modelos de carga dinámicos:

- ✓ Modelo de carga dinámico con motor de inducción.
- ✓ Modelo de carga dinámico exponencial.
- ✓ Modelo de carga compuesto.
- ✓ Modelo de carga función de transferencia con motor de inducción.
- ✓ Modelo de carga de distribución.
- ✓ Modelo de carga con barra de carga de potencia.
- ✓ Modelo de carga de sistema de almacenamiento de energía distribuido (DESS).

El presente trabajo de titulación concentrará su atención en los modelos de carga en estado estable.

1.3.4.1 Modelo de Carga Dinámico

Los modelos de carga en estado dinámico por lo general se describen en forma de ecuación diferencial, relacionando dentro de dicha ecuación la potencia activa y la potencia reactiva con el voltaje y la frecuencia.

Los estudios dinámicos se refieren al análisis de estabilidad que involucra las respuestas transitorias del sistema después de someterlo a varias perturbaciones [6].

1.3.4.2 Modelo de Carga en Estado Estable

Los modelos de carga en estado estable son adecuados para representar cargas que exhiben cambios simples, casi instantáneos e invariables en el tiempo en la demanda de energía, después de una desviación en el voltaje de alimentación o en la frecuencia de la barra de conexión [1].

La forma general de un modelo de carga en estado estable viene representada por la dependencia de la potencia activa y reactiva con el voltaje y la frecuencia. Matemáticamente, el modelo de carga en estado estable puede representarse de la siguiente manera:

$$P = f_P(V, f)$$

Ecuación 1.7. Representación del modelo de carga mediante potencia activa.

$$Q = f_Q(V, f)$$

Ecuación 1.8. Representación del modelo de carga mediante potencia reactiva.

1.3.4.2.1 Modelo de Carga Exponencial

El modelo de carga exponencial es uno de los más utilizados en estudios en estado estable y se expresa de la siguiente manera:

$$P = P_n * \left(\frac{V}{V_n}\right)^{k_{pu}} * \left(\frac{f}{f_n}\right)^{k_{pf}}$$

Ecuación 1.9. Modelo de carga exponencial para potencia activa.

$$Q = Q_n * \left(\frac{V}{V_n}\right)^{k_{qu}} * \left(\frac{f}{f_n}\right)^{k_{qf}}$$

Ecuación 1.10. Modelo de carga exponencial para potencia reactiva.

Donde:

P: Potencia activa en función del voltaje y la frecuencia.

Q: Potencia reactiva en función del voltaje y la frecuencia.

V: Voltaje de la barra donde se encuentra la carga.

V_n : Voltaje nominal de la carga.

P_n : Potencia activa a voltaje nominal V_n y frecuencia nominal f_n .

Q_n : Potencia reactiva a voltaje nominal V_n y frecuencia nominal f_n .

k_{pu} y k_{qu} : Exponentes que describen el cambio de la demanda de carga en respuesta a las variaciones de voltaje.

k_{pf} y k_{qf} : Exponentes que describen el cambio de la demanda de carga en respuesta a las variaciones de frecuencia.

Sin embargo, como el estudio del presente proyecto es un estudio en estado estable se puede despreciar la dependencia de la carga con la frecuencia considerando que los cambios de voltaje son mucho más frecuentes y más pronunciados que los cambios en la frecuencia del sistema [1]. Con esta simplificación las ecuaciones 1.9 y 1.10 pueden escribirse de la siguiente manera:

$$P = P_n * \left(\frac{V}{V_n}\right)^{k_{pu}}$$

Ecuación 1.11. Modelo de carga exponencial simplificado para potencia activa.

$$Q = Q_n * \left(\frac{V}{V_n}\right)^{k_{qu}}$$

Ecuación 1.12. Modelo de carga exponencial simplificado para potencia reactiva.

Donde los exponentes k_{pu} y k_{qu} describen las derivadas parciales de la potencia activa y reactiva con respecto al voltaje e indican la magnitud de cambio de la potencia activa y reactiva en por unidad [1].

Si los exponentes k_{pu} y k_{qu} toman los valores de 0, 1 o 2, se dice que la carga se modela como potencia constante, corriente constante o impedancia constante, respectivamente.

1.3.4.2.2 Modelo de Carga Polinomial

El modelo de carga polinomial es otro de los modelos de carga en estado estable utilizado con frecuencia, este modelo se representa mediante un polinomio de segundo grado como se indica en las ecuaciones 1.13 y 1.14:

$$P = P_n * \left[p_1 * \left(\frac{V}{V_n} \right)^2 + p_2 * \left(\frac{V}{V_n} \right) + p_3 \right]$$

Ecuación 1.13. Modelo de carga polinomial simplificado para potencia activa.

$$Q = Q_n * \left[q_1 * \left(\frac{V}{V_n} \right)^2 + q_2 * \left(\frac{V}{V_n} \right) + q_3 \right]$$

Ecuación 1.14. Modelo de carga polinomial simplificado para potencia reactiva.

Donde:

P: Potencia activa en función del voltaje.

Q: Potencia reactiva en función del voltaje.

V: Magnitud del voltaje en la barra de carga.

V_n : Voltaje nominal de la carga.

P_n : Potencia activa a voltaje nominal V_n .

Q_n : Potencia reactiva a voltaje nominal V_n .

p_1 y q_1 : Parámetros que representan la participación relativa de la carga de impedancia constante.

p_2 y q_2 : Parámetros que representan la participación relativa de la carga de corriente constante.

p_3 y q_3 : Parámetros que representan la participación relativa de la carga de potencia constante.

Este modelo de carga es también conocido como “modelo ZIP” ya que está representado por coeficientes de impedancia constante (p_1 y q_1), coeficientes de corriente constante (p_2 y q_2), y coeficientes de potencia constante (p_3 y q_3).

Los parámetros del modelo polinomial pueden tomar valores positivos, negativos o ser igual a cero pero deben cumplir las siguientes restricciones:

$$p_1 + p_2 + p_3 = 1$$

Ecuación 1.15. Restricción para potencia activa del modelo de carga polinomial.

$$q_1 + q_2 + q_3 = 1$$

Ecuación 1.16. Restricción para potencia reactiva del modelo de carga polinomial.

1.3.4.2.3 Modelo de Carga Lineal

El modelo de carga lineal, por lo general, se usa en estudios donde el voltaje varía en un rango estrecho alrededor del valor nominal, como por ejemplo en el análisis de estabilidad de pequeña perturbación [1]. Cuando las variaciones de voltaje son muy grandes este modelo no es recomendable ya que puede producir cálculos inexactos.

Al modelo de carga lineal se lo representa de la siguiente manera:

$$P = P_n * \left[a_0 + a_2 * \left(\frac{V}{V_n} \right) \right]$$

Ecuación 1.17. Modelo de carga lineal para potencia activa.

$$Q = Q_n * \left[b_0 + b_2 * \left(\frac{V}{V_n} \right) \right]$$

Ecuación 1.18. Modelo de carga lineal para potencia reactiva.

Donde:

P: Potencia activa en función del voltaje.

Q: Potencia reactiva en función del voltaje.

V: Magnitud del voltaje en la barra de carga.

V_n : Voltaje nominal de la carga.

P_n : Potencia activa a voltaje nominal V_n .

Q_n : Potencia reactiva a voltaje nominal V_n .

a_0 y b_0 : Parámetros que representan la participación relativa de la carga de potencia constante.

a_2 y b_2 : Parámetros que representan la participación relativa de la carga de potencia constante.

Al igual que en el modelo polinomial los parámetros pueden tomar valores positivos, negativos o cero, pero deben cumplir las siguientes restricciones:

$$a_0 + a_2 = 1$$

Ecuación 1.19. Restricción para potencia activa del modelo de carga lineal.

$$b_0 + b_2 = 1$$

Ecuación 1.20. Restricción para potencia reactiva del modelo de carga lineal.

1.3.4.2.4 Modelo de Carga Completo

El modelo de carga completo viene dado o es representado por la suma de un modelo polinomial más dos exponenciales como se muestra en la ecuación 1.21.

$$P = P_n * [P_{ZIP} + P_{EXP1} + P_{EXP2}]$$

Ecuación 1.21. Formulación del modelo de carga completo.

Donde:

$$P_{ZIP} = \left[p_1 * \left(\frac{V}{V_n} \right)^2 + p_2 * \left(\frac{V}{V_n} \right) + p_3 \right]$$

Ecuación 1.22. Modelo polinomial del modelo de carga completo.

$$P_{EXP1} = \left[p_4 * \left(\frac{V}{V_n} \right)^{a1} * (1 + k_{pf1} * \Delta f) \right]$$

Ecuación 1.23. Modelo exponencial del modelo de carga completo.

$$P_{EXP2} = \left[p_5 * \left(\frac{V}{V_n} \right)^{a2} * (1 + k_{pf2} * \Delta f) \right]$$

Ecuación 1.24. Modelo exponencial del modelo de carga completo.

La potencia reactiva se modela utilizando expresiones similares a las ecuaciones 1.22, 1.23 y 1.24, cabe señalar que a pesar de ser un modelo en estado estable este busca capturar el comportamiento dinámico de las cargas, es decir este modelo captura características tanto estáticas como dinámicas de las cargas representadas [1].

Este modelo propone trabajar con voltajes extremadamente bajos causados por las cargas en estado estable en especial por las cargas de electrónica de potencia. Ejemplos

claros de este tipo de cargas son las lámparas fluorescentes compactas, los televisores de alta definición y los contactores magnéticos [1].

1.3.4.2.5 Modelo de Carga del Motor de Inducción

El modelo de carga de motor de inducción en estado estable es utilizado en países desarrollados ya que el aporte de participación en la carga de los motores de inducción representa un 70% del total de la demanda [1].

El modelo de carga se obtiene en base al circuito equivalente del motor de inducción mostrado en la Figura 1.13.

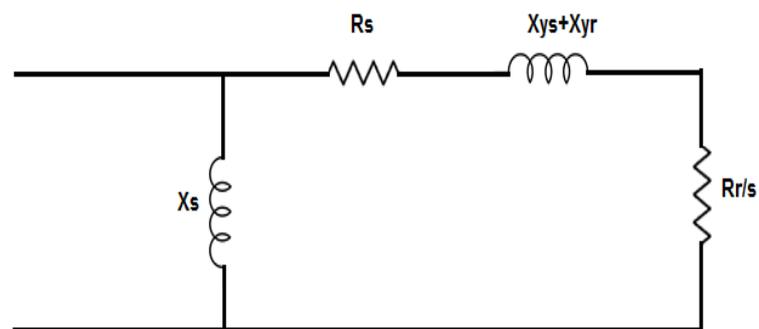


Figura 1.13. Circuito equivalente del motor de inducción utilizado para determinar el modelo de carga en estado estable.

Donde:

- R_s : Resistencia del estator.
- R_r : Resistencia del rotor.
- X_{ys} : Reactancia de dispersión del estator.
- X_{yr} : Reactancia de dispersión del rotor.
- X_m : Reactancia de magnetización.
- $X_s = X_m + X_{ys}$: Reactancia shunt.
- $s = \frac{\omega_s - \omega_r}{\omega_s}$: Deslizamiento del motor.
- ω_s : Velocidad angular sincrónica.
- ω_r : Velocidad angular del rotor.

Cabe indicar que en el circuito equivalente los parámetros del rotor se encuentran referidos al lado del estator.

Las ecuaciones del modelo de carga en estado estable para el motor de inducción son las siguientes:

$$P = \left(R_s + \frac{R_r}{s} \right) * \frac{V^2}{\left(R_s + \frac{R_r}{s} \right)^2 + (X_{ys} + X_{yr})^2}$$

Ecuación 1.25. Modelo de motor de inducción para potencia activa.

$$Q = (X_{ys} + X_{yr}) * \frac{V^2}{\left(R_s + \frac{R_r}{s} \right)^2 + (X_{ys} + X_{yr})^2} + \frac{V^2}{X_s}$$

Ecuación 1.26. Modelo de motor de inducción para potencia reactiva.

1.3.4.2.6 Modelo de Carga con Interfaz de Electrónica de Potencia

El modelo de carga con interfaz de electrónica de potencia ha tomado un gran impulso en los últimos años debido a que las cargas electrónicas han aumentado considerablemente en la demanda del sistema, además se prevé que estas cargas seguirán aumentando con el pasar de los años. Este tipo de modelo de carga se ha clasificado en cuatro grupos: 1) suministro de potencia de corriente continua, 2) fuentes de luz eficientes como por ejemplo las de tipo LED, 3) lámparas compactas fluorescentes (CFL) y 4) cargas de motores controlados y motores trifásicos de velocidad ajustable [1].

Los modelos de carga para estos cuatro grupos son diferentes, pueden ser representados mediante circuitos completos o circuitos equivalentes. En la práctica se recomienda utilizar los circuitos equivalentes ya que son más simples que los circuitos completos.

Los modelos desarrollados mediante circuitos equivalentes son representados por un puente rectificador de diodos, una impedancia de entrada, un capacitor y una resistencia equivalente como se muestra en la Figura 1.14.

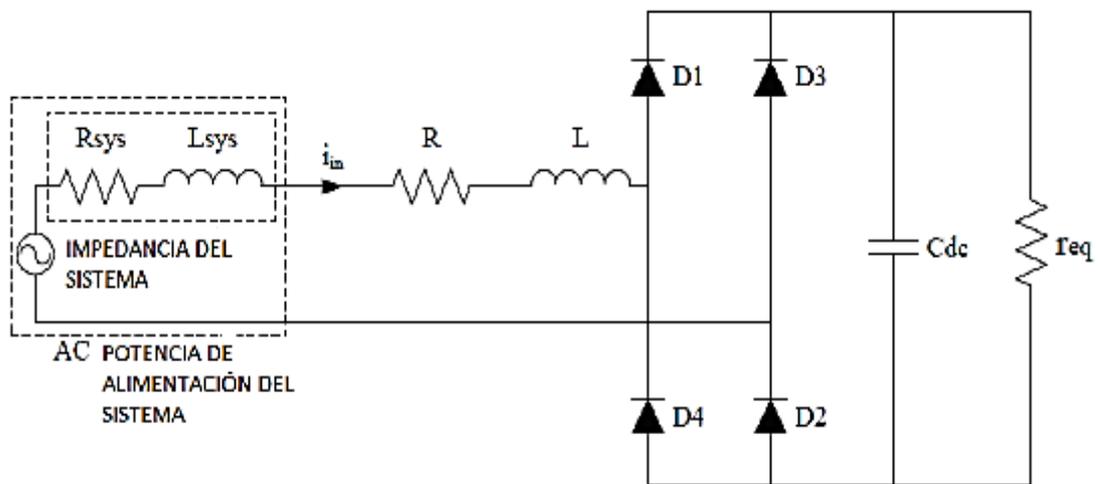


Figura 1.14. Circuito equivalente para el modelo de carga con interfaz de electrónica de potencia [1].

2. METODOLOGÍAS PARA LA INCORPORACIÓN DE MODELOS DE CARGA EN EL CÁLCULO DE FLUJOS DE POTENCIA

2.1 Introducción

El cálculo de flujos de potencia es de gran importancia dentro de la planificación, operación y expansión de los sistemas eléctricos de potencia, ya que la información obtenida después de su cálculo ayuda a determinar las mejores condiciones de operación del sistema [3].

En este capítulo se explica el método de Newton Raphson usado para la resolución de flujos de potencia en estado estable. Además se explican las metodologías de Inyección Nodal, Flujos Sucesivos y Reestructuración de Potencias Calculadas usadas para incorporar modelos de carga en estado estable dentro de la solución de flujos de potencia mediante el método de Newton Raphson.

2.2 Método de Newton Raphson (N-R) para el cálculo de flujos de potencia

El método de Newton Raphson permite resolver un sistema de ecuaciones no lineales de manera iterativa y en un tiempo pequeño a pesar del gran número de cálculos que se realizan para resolver un sistema $n \cdot n$ (n : número de barras). Este método es recomendado en la solución de ecuaciones algebraicas no lineales como por ejemplo en la solución de las ecuaciones nodales de potencia inyectada que describen los sistemas eléctricos en estado estacionario ya que presenta una convergencia robusta y rápida.

2.2.1 Tipos de Barras

En un sistema eléctrico de potencia se tienen cuatro variables reales asociadas a cada una de las barras:

- Magnitud del Voltaje.
- Ángulo de voltaje.
- Potencia activa.
- Potencia reactiva.

Por lo general, en cada una de las barras dos de las variables son conocidas inicialmente y las dos restantes son calculadas después de ejecutar el flujo de potencia. En base a las variables conocidas las barras pueden clasificarse según lo indicado en la tabla 2.1:

Tabla 2.1. Tipos de Barras

Tipo de Barra	Variables Conocidas	Variables a Determinarse
Slack	V y θ	P y V
De voltaje controlado o PV	P y V	Q y θ
De carga o PQ	P y Q	V y θ
De compensación	V	P, Q y θ

2.2.2 Ecuaciones de Red

Las relaciones entre el voltaje y la corriente en las barras de un sistema eléctrico de potencia se pueden representar por las ecuaciones de malla o mediante las ecuaciones de nodo. Se prefiere la representación mediante ecuaciones de nodo ya que el número de ecuaciones independientes de nodo es menor que el número de ecuaciones independientes de malla [3].

Las ecuaciones de red en términos de la matriz de admitancias de nodo pueden ser escritas como se muestra en la ecuación 2.1.

$$\begin{bmatrix} \bar{I}_1 \\ \bar{I}_2 \\ \bar{I}_3 \\ \dots \\ \bar{I}_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & Y_{23} & \dots & Y_{2n} \\ Y_{31} & Y_{32} & Y_{33} & \dots & Y_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ Y_{n1} & Y_{n2} & Y_{n3} & \dots & Y_{nn} \end{bmatrix} * \begin{bmatrix} \bar{V}_1 \\ \bar{V}_2 \\ \bar{V}_3 \\ \dots \\ \bar{V}_n \end{bmatrix}$$

Ecuación 2.1. Relación entre voltajes y corrientes.

Donde:

n: Número total de nodos.

Y_{ii} : Admitancia propia del nodo i. (Suma de las admitancias conectadas al nodo i)

Y_{ij} : Admitancia mutua entre los nodos i y j. (Negativo de la suma de las admitancias conectadas entre los nodos i y j.

\bar{V}_i : Fasor de voltaje (fase- neutro) del nodo i.

\bar{I}_i : Fasor de corriente que fluye al nodo i.

La ecuación 2.1 puede reescribirse de la siguiente manera:

$$I = Y_{barras} * V$$

Ecuación 2.2. Ecuación de red.

Donde Y_{barras} es la matriz de admitancias de barra del sistema, esta matriz tiene las siguientes características:

- Es una matriz cuadrada de orden n (número de barras) con términos complejos.
- Es una matriz simétrica respecto a la diagonal, es decir $Y_{ij} = Y_{ji}$.
- Los elementos de la diagonal de la matriz son negativos, mientras que los elementos fuera de la diagonal son positivos.
- Se calcula en base a las impedancias y admitancias de los elementos del sistema.

Cabe resaltar que debido a que los elementos del sistema eléctrico de potencia se modelan mediante un equivalente trifásico balanceado no existe circulación de corriente por neutro ni por tierra, razón por la cual si se descompone la red trifásica en sus redes de secuencia positiva, negativa y cero, solo se considera el equivalente monofásico de la red de secuencia positiva en el cálculo de flujos de potencia.

2.2.3 Ecuaciones de Inyección de Potencia

La ecuación 2.1 no se puede resolver directamente debido a que presenta como incógnitas al voltaje en las barras y a las inyecciones de corriente, estas variables por lo general son desconocidas en la mayoría de las barras del sistema. Dada esta problemática se utilizan los datos de potencia en generadores y cargas para establecer los voltajes en las barras, los cuales pueden determinarse mediante las siguientes expresiones:

$$P_i = V_i * \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sen \theta_{ij}) V_j$$

Ecuación 2.3. Inyección de potencia activa.

$$Q_i = V_i * \sum_{j=1}^n (G_{ij} \sen \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j$$

Ecuación 2.4. Inyección de potencia reactiva.

Donde:

P_i : Potencia activa total inyectada en el nodo i.

Q_i : Potencia reactiva total inyectada en el nodo i.

V_i : Magnitud de voltaje de la barra i.

V_j : Magnitud de voltaje de la barra j.

G_{ij} : Conductancia mutua entre los nodos i y j.

B_{ij} : Suceptancia mutua entre los nodos i y j.

$\theta_{ij} = \theta_i - \theta_j$: Diferencia angular entre los nodos i y j.

Las ecuaciones 2.3 y 2.4 se conocen como ecuaciones de inyección de potencia ya que representan la inyección total de potencia en una barra i del sistema. Los valores del flujo de potencia no pueden ser calculados directamente debido a que las magnitudes y los ángulos del voltaje no son conocidos inicialmente.

2.2.4 Errores de Potencia

Los errores o variaciones de potencia establecen que en cualquier barra del sistema la suma de potencias activas y reactivas inyectadas debe ser cero, es decir debe existir un balance entre la generación, carga y potencia inyectada.

El error de potencias en las barras de un sistema se representa mediante las siguientes ecuaciones:

$$\Delta P_i = P_i^{esp} - P_i^{cal}$$

Ecuación 2.5. Error de potencia activa.

$$\Delta Q_i = Q_i^{esp} - Q_i^{cal}$$

Ecuación 2.6. Error de potencia reactiva.

Donde:

$P_i^{esp} = P_{Gi} - P_{Li}$: Potencia activa especificada.

$Q_i^{esp} = Q_{Gi} - Q_{Li}$: Potencia reactiva especificada.

P_i^{cal} : Potencia activa calculada. Se calcula mediante la ecuación 2.2.

Q_i^{cal} : Potencia reactiva calculada. Se calcula mediante la ecuación 2.3.

P_{Gi} :	Potencia activa de generación. Se obtiene en base a los datos de los despachos programados en los generadores.
Q_{Gi} :	Potencia reactiva de generación. Se obtiene en base a los datos de los despachos programados en los generadores.
P_{Li} :	Potencia activa de carga. Se obtiene en base a los datos de las cargas del sistema.
Q_{Li} :	Potencia reactiva de generación. Se obtiene en base a los datos de las cargas del sistema.

2.2.5 Formulación del Algoritmo de Newton Raphson.

El método de Newton Raphson para la solución de flujos de potencia es una técnica iterativa que resuelve un conjunto de n ecuaciones no lineales con n incógnitas, se basa en la expansión de la serie de Taylor para una función de dos o más variables [3].

Para expandir una función mediante la serie de Taylor se debe aplicar la expresión mostrada en la ecuación 2.7, para ello la función $f(x)$ debe ser diferenciable en un intervalo.

$$f(x) = f(x_0) + \frac{df(x_0)}{x_0} \Delta x + \frac{d^2f(x_0)}{x_0^2} \Delta x^2 + \dots + \frac{d^n f(x_0)}{x_0^n} \Delta x^n$$

Ecuación 2.7. Expansión de serie de Taylor.

Donde:

$$\Delta x = x - x_0$$

Ecuación 2.8. Error de la variable de estado x .

Si se considera que la aproximación x_0 es lo suficientemente buena la expresión Δx toma un valor pequeño, razón por la cual las derivadas de orden superior pueden despreciarse debido a que sus valores son muy cercanos a cero. Tomando en cuenta esta aproximación la ecuación 2.7 puede reescribirse de la siguiente manera:

$$f(x) \cong f(x_0) + \frac{df(x_0)}{dx_0} (x - x_0) = 0$$

Ecuación 2.9. Expansión de serie de Taylor.

Si se despeja x de la ecuación 2.8 se obtiene:

$$x = x_0 + \frac{f(x_0)}{f'(x_0)}$$

Ecuación 2.10. Variable actual de la serie de Taylor.

El valor de x calculado a partir de la aproximación x_0 , se usa para determinar un nuevo valor de x de la siguiente manera:

$$x^{(i+1)} = x^{(i)} + \frac{f(x^{(i)})}{f'(x^{(i)})}$$

Ecuación 2.11. Actualización de la variable x en la serie de Taylor.

Donde i representa el número de la iteración calculada, este proceso se repite hasta que la diferencia entre el término calculado en la última iteración y el calculado en la iteración anterior es menor a una tolerancia establecida al inicio del proceso iterativo, por lo general esta diferencia es muy cercana a cero.

A continuación se aplica el método de Newton Raphson al siguiente sistema de ecuaciones no lineales:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= h_1 \\ f_2(x_1, x_2, \dots, x_n) &= h_2 \\ &\dots\dots\dots \\ f_n(x_1, x_2, \dots, x_n) &= h_n \end{aligned}$$

Ecuación 2.12. Sistema de Ecuaciones no Lineales.

La ecuación 2.12 se puede reescribir de la siguiente manera:

$$F(X) = H$$

Ecuación 2.13. Representación del sistema de ecuaciones no lineales.

Si el conjunto de estimaciones iniciales $\{x_1^0, x_2^0, \dots, x_n^0\}$ es cercano a la solución, la expansión de la serie de Taylor truncada a la primera derivada y tratada matricialmente es la siguiente:

$$\begin{bmatrix} h_1 - f_1(x_1^0, x_2^0, \dots, x_n^0) \\ h_2 - f_2(x_1^0, x_2^0, \dots, x_n^0) \\ \vdots \\ h_n - f_n(x_1^0, x_2^0, \dots, x_n^0) \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)_0 & \left(\frac{\partial f_1}{\partial x_2}\right)_0 & \dots & \left(\frac{\partial f_1}{\partial x_n}\right)_0 \\ \left(\frac{\partial f_2}{\partial x_1}\right)_0 & \left(\frac{\partial f_2}{\partial x_2}\right)_0 & \dots & \left(\frac{\partial f_2}{\partial x_n}\right)_0 \\ \vdots & \vdots & \ddots & \vdots \\ \left(\frac{\partial f_n}{\partial x_1}\right)_0 & \left(\frac{\partial f_n}{\partial x_2}\right)_0 & \dots & \left(\frac{\partial f_n}{\partial x_n}\right)_0 \end{bmatrix} \begin{bmatrix} \Delta x_1^0 \\ \Delta x_2^0 \\ \vdots \\ \Delta x_n^0 \end{bmatrix}$$

Ecuación 2.14. Expansión de la serie de Taylor truncada a la primera derivada.

La ecuación 2.14 se puede reescribir de la siguiente manera:

$$\Delta F = J * \Delta X$$

Ecuación 2.15. Expansión de la serie de Taylor en forma submatricial.

Donde:

ΔF : Matriz de errores resultantes de la diferencia de valores estimados y evaluados.

J : Jacobiano para las funciones f_i .

ΔX : Vector de incógnitas Δx_i .

Mediante la ecuación 2.14 se linealiza los errores ΔF y las correcciones ΔX a través del Jacobiano. Dentro del método de Newton Raphson la actualización de los valores en cada iteración se representa de la siguiente manera:

$$X^{(actual)} = X^{(anterior)} + \Delta X^{(i)}$$

Ecuación 2.16. Actualización de la variable de estado.

El proceso se repite hasta que los errores ΔF sean cero o menores a una tolerancia especificada, mientras más pequeña sea esta tolerancia mayor será la exactitud del resultado. Cabe indicar que en cada nueva iteración el Jacobiano se recalcula.

2.2.6 Aplicación del algoritmo de Newton Raphson al cálculo de Flujos de Potencia.

El algoritmo de Newton Raphson dentro de la solución de flujos de potencia es utilizado para resolver las ecuaciones de inyección de potencia debido a que estas presentan una característica no lineal.

Al utilizar esta metodología para resolver el flujo de potencia las variables de estado a determinarse son la magnitud del voltaje y los ángulos de voltaje en todas las barras del sistema. Entonces el vector de variables de estado a determinarse puede representarse de la siguiente manera:

$$X = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \\ V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix}$$

Ecuación 2.17. Variables de estado del método de Newton - Raphson.

Las ecuaciones de inyección de potencia reciben el nombre de potencias calculadas y se representan de la siguiente manera:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sen \theta_{ij}) V_j$$

Ecuación 2.18. Potencia activa calculada.

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \sen \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j$$

Ecuación 2.19. Potencia reactiva calculada.

De la misma manera, mediante el balance generación carga para todas las barras del sistema se calculan las potencias especificadas mediante las siguientes expresiones:

$$P_i^{esp} = P_i^{generación} - P_i^{carga}$$

Ecuación 2.20 Potencia activa especificada.

$$Q_i^{esp} = Q_i^{generación} - Q_i^{carga}$$

Ecuación 2.21 Potencia reactiva especificada.

Cabe mencionar que para cada barra del sistema se calcula una ecuación de potencia activa y potencia reactiva, razón por la cual para un sistema de n número de barras el número de ecuaciones calculadas es 2·n.

Entonces la solución de flujos de potencia mediante el algoritmo de Newton -Raphson se representa como se muestra a continuación:

$$\begin{bmatrix} P_1^{esp} - P_1^{cal} \\ \vdots \\ P_n^{esp} - P_n^{cal} \\ Q_1^{esp} - Q_1^{cal} \\ \vdots \\ Q_n^{esp} - Q_n^{cal} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_1}{\partial \theta_1} & \dots & \frac{\partial P_1}{\partial \theta_n} & V_1 \frac{\partial P_1}{\partial V_1} & \dots & V_n \frac{\partial P_1}{\partial V_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_n}{\partial \theta_1} & \dots & \frac{\partial P_n}{\partial \theta_n} & V_1 \frac{\partial P_n}{\partial V_1} & \dots & V_n \frac{\partial P_n}{\partial V_n} \\ \frac{\partial Q_1}{\partial \theta_1} & \dots & \frac{\partial Q_1}{\partial \theta_n} & V_1 \frac{\partial Q_1}{\partial V_1} & \dots & V_n \frac{\partial Q_1}{\partial V_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_n}{\partial \theta_1} & \dots & \frac{\partial Q_n}{\partial \theta_n} & V_1 \frac{\partial Q_n}{\partial V_1} & \dots & V_n \frac{\partial Q_n}{\partial V_n} \end{bmatrix} \begin{bmatrix} \Delta \theta_1 \\ \vdots \\ \Delta \theta_n \\ \frac{\Delta V_1}{V_1} \\ \vdots \\ \frac{\Delta V_n}{V_n} \end{bmatrix}$$

Ecuación 2.22. Representación de la solución de flujos de potencia mediante el algoritmo de Newton - Raphson.

La expresión 2.22 puede representarse en forma de sub-matrices de la siguiente manera:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \theta} & V \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \theta} & V \frac{\partial Q}{\partial V} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \frac{\Delta V}{V} \end{bmatrix}$$

Ecuación 2.23. Representación submatricial de la solución de flujos de potencia mediante algoritmo de Newton - Raphson.

2.2.6.1 Cálculo de los elementos del Jacobiano

En la ecuación 2.23 se observa que los elementos del Jacobiano pueden calcularse a partir de las derivadas parciales de las expresiones 2.18 y 2.19 de la siguiente manera:

2.2.6.1.1 Cálculo de los elementos fuera de la diagonal ($i \neq j$)

$$\frac{\partial P_i}{\partial \theta_j} = V_i V_j (G_{ij} \text{sen} \theta_{ij} - B_{ij} \text{cos} \theta_{ij})$$

Ecuación 2.24. Cálculo de elementos fuera de la diagonal del Jacobiano.

$$V_j \frac{\partial P_i}{\partial V_j} = V_i V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij})$$

Ecuación 2.25. Cálculo de elementos fuera de la diagonal del Jacobiano.

$$\frac{\partial Q_i}{\partial \theta_j} = -V_j \frac{\partial P_i}{\partial V_j} = -V_i V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij})$$

Ecuación 2.26. Cálculo de elementos fuera de la diagonal del Jacobiano.

$$V_j \frac{\partial Q_i}{\partial V_j} = \frac{\partial P_i}{\partial \theta_j} = V_i V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij})$$

Ecuación 2.27. Cálculo de elementos fuera de la diagonal del Jacobiano.

2.2.6.1.2 Cálculo de los elementos de la diagonal (i=j)

$$\frac{\partial P_i}{\partial \theta_i} = -Q_i^{cal} - V_i^2 B_{ii}$$

Ecuación 2.28. Cálculo de elementos de la diagonal del Jacobiano.

$$V_i \frac{\partial P_i}{\partial V_i} = P_i^{cal} + V_i^2 G_{ii}$$

Ecuación 2.29. Cálculo de elementos de la diagonal del Jacobiano.

$$\frac{\partial Q_i}{\partial \theta_i} = P_i^{cal} - V_i^2 G_{ii}$$

Ecuación 2.30. Cálculo de elementos de la diagonal del Jacobiano.

$$V_i \frac{\partial Q_i}{\partial V_i} = Q_i^{cal} - V_i^2 B_{ii}$$

Ecuación 2.31. Cálculo de elementos de la diagonal del Jacobiano.

El proceso iterativo de Newton Raphson se repite hasta que los errores de potencia activa ΔP y potencia reactiva ΔQ de todas las barras sean cero o menores a una tolerancia especificada al inicio del algoritmo. En cada nueva iteración se recalcula el Jacobiano y se actualizan los valores de las variables de estado, sumando a estas los valores de $\Delta \theta$ y ΔV obtenidos en esa iteración, es decir para una nueva iteración se tiene:

$$\theta_i^{(k+1)} = \theta_i^{(k)} + \Delta\theta_i^{(k)}$$

Ecuación 2.32. Actualización del valor del ángulo de voltaje.

$$|V_i|^{(k+1)} = |V_i|^{(k)} + \Delta|V_i|^{(k)} = |V_i|^{(k)} \left(1 + \frac{\Delta|V_i|^{(k)}}{|V_i|^{(k)}} \right)$$

Ecuación 2.33. Actualización del valor de la magnitud de voltaje.

Como ya se mencionó, la finalidad de esta metodología es la de obtener la magnitud del voltaje y el ángulo de voltaje en todas las barras del sistema, por tal razón las variables de estado del proceso van a ser la magnitud y ángulo de voltaje, las mismas que dentro del proceso iterativo se inicializan según el tipo de barra. Para barras de carga o PQ la magnitud de voltaje en las barras se inicializa en 1 p.u. y el ángulo de voltaje en 0°, mientras que para barras PV y slack las magnitudes de voltaje se inicializan en sus valores conocidos permaneciendo estos constantes durante todo el proceso, en lo referente a los ángulos estos se inicializan en cero para ambos tipos de barras pero únicamente el ángulo de la barra slack permanece constante durante todo el proceso. También es importante señalar que la barra slack o de referencia no aporta ecuaciones al proceso debido a que la magnitud y ángulo de voltaje son conocidos, todo lo contrario que las barras PQ, las cuales aportan las dos ecuaciones al sistema, las barras PV aportan solo la ecuación de potencia activa ya que debido a que no se conoce la potencia reactiva de generación no se puede determinar ΔQ .

2.2.7 Diagrama de flujo del algoritmo de Newton Raphson aplicado a la solución de Flujos de Potencia

La Figura 2.1 muestra el diagrama utilizado para resolver un flujo de potencia mediante el algoritmo de Newton Raphson, cabe mencionar que en este diagrama se considera que los datos de entrada vienen dados en por unidad, razón por la cual no es necesario realizar una transformación previa. En el diagrama se observa que el proceso inicia con la lectura de datos e inicialización de las variables de estado, seguidamente se calculan las potencias especificadas P^{esp} , Q^{esp} y las potencias calculadas P^{cal} y Q^{cal} . Luego se inicia el proceso iterativo con el cálculo de los errores de potencia, se chequea si estos cumplen con la tolerancia especificada para el proceso, en caso de no cumplir la convergencia se calcula el Jacobiano y se actualizan los valores de las variables de estado. Este proceso se realiza hasta que los errores de potencia cumplan con la tolerancia establecida, finalmente cuando el proceso haya convergido satisfactoriamente se calculan las potencias en las barras y se imprimen los resultados.

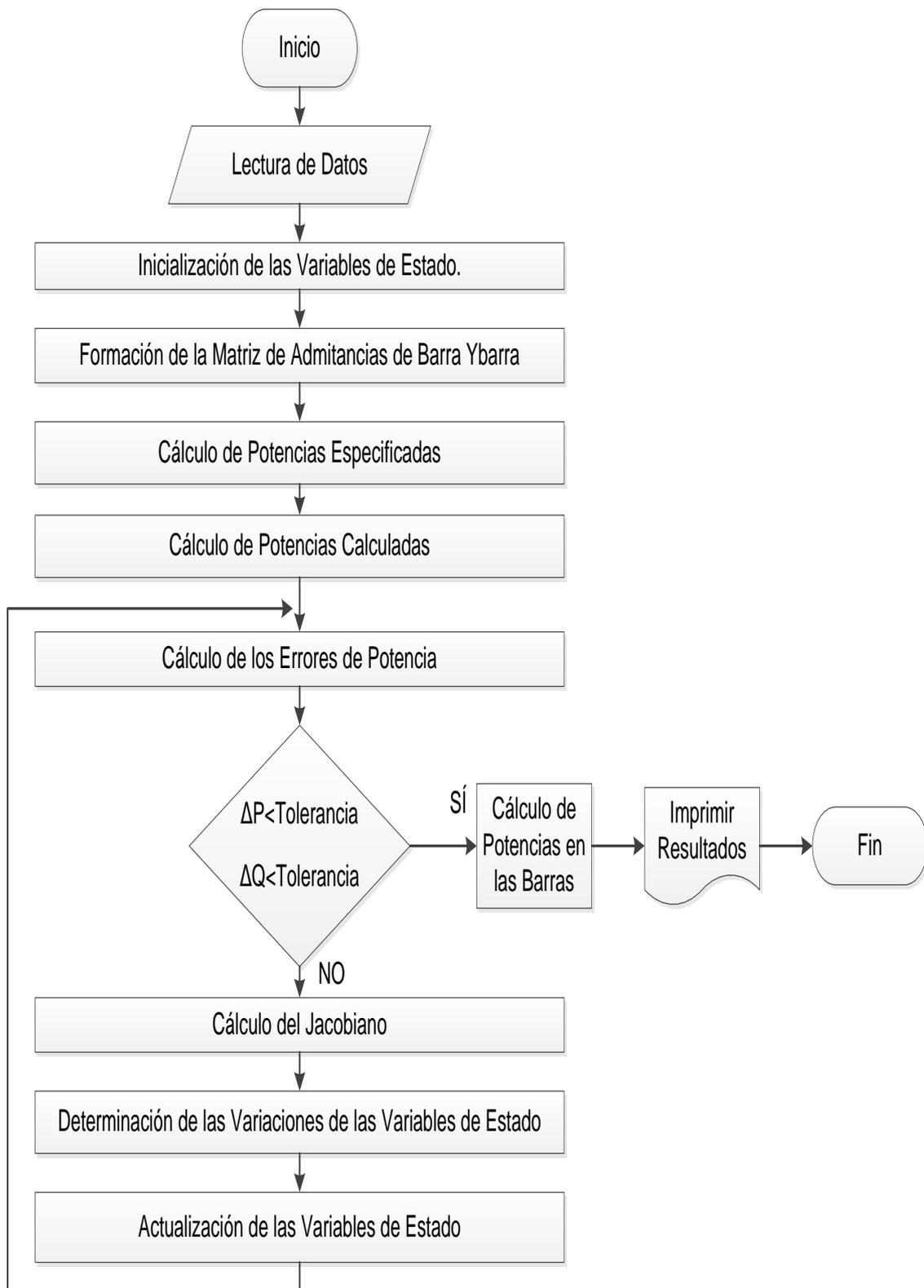


Figura 2.1. Diagrama de Flujo de la Metodología de Newton Raphson para el cálculo de flujos de potencia

2.3 Metodologías para la Incorporación de Modelos de Carga en Estado Estable al Cálculo de Flujos de Potencia

En esta sección se detallan tres metodologías que permiten la incorporación de modelos de carga en estado estable dentro del cálculo de flujos de potencia mediante el método de Newton Raphson.

2.3.1 Metodología de Inyección Nodal

En la formulación del cálculo de flujos de potencia mediante la aplicación del Algoritmo de Newton – Raphson mostrado en las secciones 2.2.5 y 2.2.6 se muestra que el sistema de ecuaciones a resolverse es el siguiente:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta \theta \\ \Delta V/V \end{bmatrix}$$

Ecuación 2.34. Sistema de ecuaciones a resolverse en el Cálculo de Flujos de Potencia mediante la metodología de Newton - Raphson.

Donde:

$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$: Matriz de Errores de Potencia.

$[J]$: Matriz Jacobiana.

$\begin{bmatrix} \Delta \theta \\ \Delta V/V \end{bmatrix}$: Matriz de Variables de Estado.

La matriz de errores de potencia se forma mediante la diferencia entre potencias especificadas y calculadas como se muestra en la Ecuación 2.35.

$$\begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} = \begin{bmatrix} P_i^{esp} - P_i^{cal} \\ Q_i^{esp} - Q_i^{cal} \end{bmatrix}$$

Ecuación 2.35. Formación de la matriz de errores de potencia.

En la solución del flujo de potencia normal mediante la metodología de Newton – Raphson los valores de las potencias especificadas son constantes, pero en la metodología de Inyección Nodal el efecto del modelo de carga en función del voltaje viene incluido dentro de este término, así por ejemplo para el modelo de carga exponencial se tiene:

$$P_i^{esp} = P_n * \left(\frac{V}{V_n}\right)^{k_{pu}}$$

Ecuación 2.36. Potencia activa especificada para el modelo de carga exponencial.

$$Q_i^{esp} = Q_n * \left(\frac{V}{V_n}\right)^{k_{qu}}$$

Ecuación 2.37. Potencia reactiva especificada para el modelo de carga exponencial.

La metodología de Inyección Nodal consiste únicamente en modificar la forma en la que se calculan las potencias especificadas, estas potencias como ya se mencionó se calculan mediante el modelo de carga que se desee incorporar a la solución de flujos de potencia. Cabe mencionar que debido a esta modificación, el valor de las potencias especificadas ya no es constante sino que depende directamente del voltaje calculado en la barra.

Esta metodología permite incorporar todos los modelos de carga en estado estable mostrados en la sección 1.3.4.2.

2.3.1.1 Diagrama de flujo

El algoritmo de esta metodología presenta una ligera modificación con respecto al diagrama de flujo mostrado en la sección 2.2.7. Para el cálculo del flujo de potencia mediante esta metodología en cada nueva iteración se deben recalculan las potencias especificadas, debido a que estas dependen del voltaje actual de la barra. El diagrama de flujo de esta metodología se muestra a continuación en la Figura 2.2, en este se observa el proceso para calcular un flujo de potencia incorporando modelos de carga mediante la metodología de Inyección Nodal, el proceso inicia con la lectura de datos e inicialización de las variables de estado, luego se calcula la matriz de admitancias de barra Y_{barra} . A continuación se inicia el proceso iterativo con el cálculo de las potencias activas y reactivas especificadas P^{esp} , Q^{esp} , seguidamente se calculan las potencias inyectadas en las barras P^{cal} y Q^{cal} , luego se calculan los errores de potencia y se chequea el cumplimiento de la tolerancia especificada para el proceso. En caso de no cumplir la convergencia se calcula el Jacobiano y se actualizan los valores de las variables de estado, este proceso se realiza hasta que los errores de potencia cumplan con la tolerancia establecida, finalmente cuando el proceso haya convergido satisfactoriamente se calculan las potencias en las barras y se imprimen los resultados.

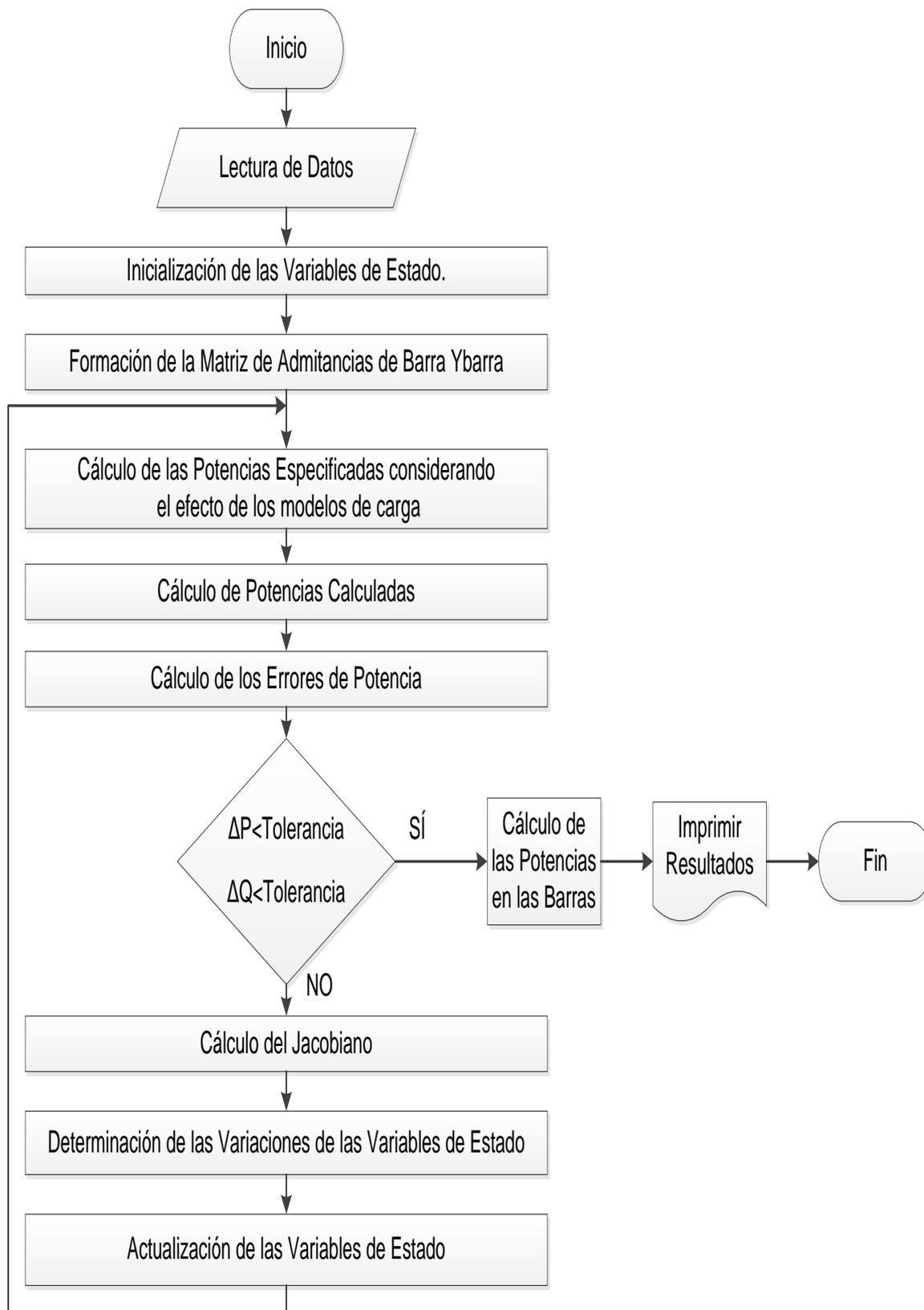


Figura 2.2. Diagrama de Flujo de la Metodología de Inyección Nodal.

2.3.2 Metodología de Flujos Sucesivos

Esta metodología recibe el nombre de Flujos Sucesivos debido a que para incorporar los modelos de carga en estado estable se realiza la solución de varios flujos de potencia.

En esta metodología la inicialización de los valores de las variables de estado se realiza a través de la solución del flujo de potencia normal del sistema, es decir se resuelve el flujo de potencia sin considerar modelos de carga y los valores obtenidos son considerados como los valores iniciales de las variables de estado.

Al igual que en la metodología anterior el efecto de los modelos de carga se ve reflejado en el cálculo de las potencias especificadas, las mismas que dependen del modelo de carga que se desee incorporar, así por ejemplo para un modelo de carga polinomial se tiene:

$$P_i^{esp} = P_n * \left[p_1 * \left(\frac{V}{V_n} \right)^2 + p_2 * \left(\frac{V}{V_n} \right) + p_3 \right]$$

Ecuación 2.38. Potencia activa especificada para el modelo de carga polinomial.

$$Q_i^{esp} = Q_n * \left[q_1 * \left(\frac{V}{V_n} \right)^2 + q_2 * \left(\frac{V}{V_n} \right) + q_3 \right]$$

Ecuación 2.39. Potencia reactiva especificada para el modelo de carga polinomial.

En esta metodología se deben fijar dos tolerancias una utilizada para determinar el error aceptable de los errores de potencia en el cálculo del flujo y otra para determinar que el algoritmo utilizado para incorporar modelos de carga ha concluido. Estas tolerancias pueden tener los mismos valores aunque se recomienda que la tolerancia del algoritmo sea menor a la tolerancia del flujo.

El criterio utilizado para determinar la convergencia del algoritmo consiste en comparar las magnitudes de voltaje obtenidas en la solución actual con las magnitudes obtenidas en la solución anterior, es decir el valor absoluto de la diferencia de todas las magnitudes de voltaje debe ser menor que una tolerancia fijada.

$$Error = |V_{actual} - V_{anterior}|$$

Ecuación 2.40. Criterio de convergencia de la metodología de Flujos Sucesivos.

Mientras el *Error* mostrado en la Ecuación 2.40 no cumpla con la tolerancia establecida para la convergencia del algoritmo, se procede a calcular un nuevo flujo de potencia y así sucesivamente hasta lograr que se cumpla con la tolerancia.

Cabe resaltar que esta metodología se caracteriza por realizar un número considerable de iteraciones para obtener los resultados. El número total de iteraciones realizadas es igual a la suma de las iteraciones de los distintos flujos ejecutados hasta obtener la respuesta.

$$it_{total} = it_{flujo\ 1} + \dots + it_{flujo\ i}$$

Ecuación 2.41. Número de iteraciones totales de la Metodología de Flujos Sucesivos.

Esta metodología permite incorporar todos los modelos de carga en estado estable mostrados en la sección 1.3.4.2.

2.3.2.1 Diagrama de Flujo

La principal característica de esta metodología es que las potencias especificadas considerando el modelo de carga permanecen constantes durante el cálculo del flujo de potencia, pero se actualizan si luego de comparar el criterio de convergencia del algoritmo se necesita correr un nuevo flujo de potencia. El diagrama de flujo de esta metodología se muestra a continuación en la Figura 2.3, en este se observa el proceso para calcular un flujo de potencia incorporando modelos de carga mediante la metodología de Flujos Sucesivos, el proceso inicia con la lectura de datos e inicialización de las variables de estado mediante los valores obtenidos en el cálculo del flujo de potencia del sistema sin considerar modelos de carga. A continuación se inicia el proceso iterativo con el cálculo de las potencias activas y reactivas especificadas P^{esp} , Q^{esp} , seguidamente se calculan las potencias inyectadas en las barras P^{cal} y Q^{cal} , luego se calculan los errores de potencia y se chequea el cumplimiento de la tolerancia especificada para el proceso, en caso de no cumplir la convergencia se calcula el Jacobiano y se actualizan los valores de las variables de estado. Este proceso se realiza hasta que los errores de potencia cumplan con la tolerancia establecida. Cuando se hayan determinado los valores de las variables de estado, se comprueba la condición de convergencia del algoritmo mediante la diferencia entre las magnitudes de voltaje actuales y las magnitudes de voltaje anteriores, en caso de no cumplir con esta condición se procede a calcular un nuevo flujo de potencia. Finalmente cuando el proceso haya convergido satisfactoriamente se calculan las potencias en las barras y se imprimen los resultados.

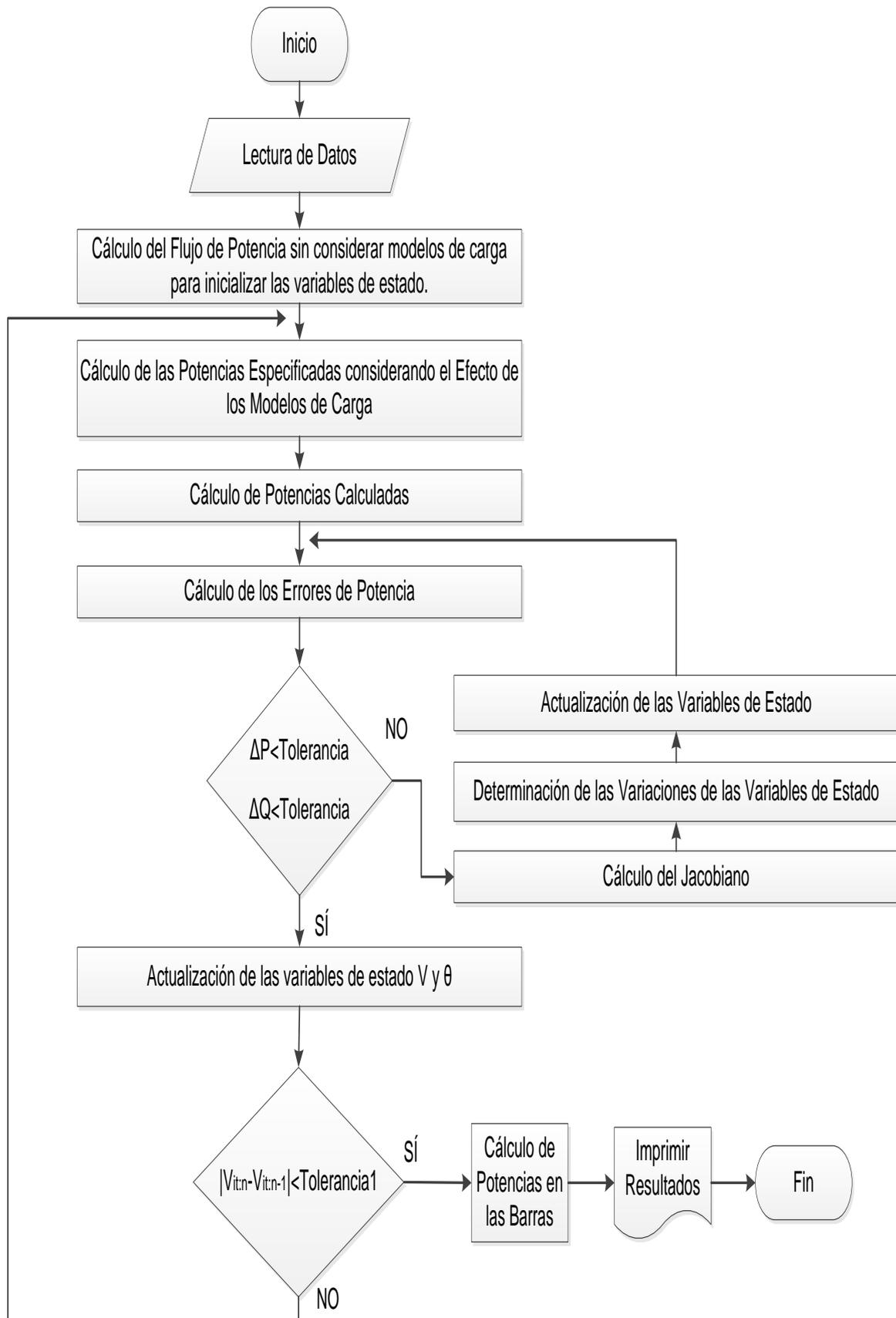


Figura 2.3. Diagrama de Flujo de la Metodología de Flujos Sucesivos.

2.3.3 Metodología de Reestructuración de Potencias Calculadas

Esta metodología consiste en incorporar el efecto de los modelos de carga dentro del cálculo de las potencias calculadas, así por ejemplo para el modelo lineal se tiene:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) V_j - P_n * \left[a_0 + a_2 * \left(\frac{V}{V_n} \right) \right]$$

Ecuación 2.42. Potencia activa calculada para el modelo de carga lineal.

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j - Q_n * \left[b_0 + b_2 * \left(\frac{V}{V_n} \right) \right]$$

Ecuación 2.43. Potencia reactiva calculada para el modelo de carga lineal.

Entonces las expresiones generales para determinar las potencias calculadas incorporando el efecto de los modelos de carga dentro de la metodología de Reestructuración de Potencias Calculadas son:

$$P_i^{cal} = P_i^{inyección} - P_i^{modelo de carga}$$

Ecuación 2.44. Expresión para el cálculo de potencia activa calculada.

$$Q_i^{cal} = Q_i^{inyección} - Q_i^{modelo de carga}$$

Ecuación 2.45. Expresión para el cálculo de potencia reactiva calculada.

Debido a estos cambios el cálculo de la matriz de errores de potencia también sufre modificaciones, ya que dentro de esta metodología se calcula mediante la siguiente expresión:

$$\begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} = \begin{bmatrix} -P_i^{cal} \\ -Q_i^{cal} \end{bmatrix}$$

Ecuación 2.46. Cálculo de la matriz de errores de potencia en la metodología de Reestructuración de Potencias Calculadas.

Las expresiones para el cálculo del Jacobiano son las mostradas en la sección 2.2.6.1, estas expresiones no sufren modificaciones pero los elementos de la diagonal van a presentar ligeros cambios en sus valores debido a que estos están directamente relacionados con las potencias calculadas. Las nuevas expresiones de cálculo de los elementos de la diagonal del Jacobiano son las siguientes:

$$\frac{\partial P_i}{\partial \theta_i} = -Q_i^{cal} + Q_i^{modelo\ de\ carga} - V_i^2 B_{ii}$$

Ecuación 2.47. Cálculo de elementos de la diagonal del Jacobiano.

$$V_i \frac{\partial P_i}{\partial V_i} = P_i^{cal} - P_i^{modelo\ de\ carga} + V_i^2 G_{ii}$$

Ecuación 2.48. Cálculo de elementos de la diagonal del Jacobiano.

$$\frac{\partial Q_i}{\partial \theta_i} = P_i^{cal} - P_i^{modelo\ de\ carga} - V_i^2 G_{ii}$$

Ecuación 2.49. Cálculo de elementos de la diagonal del Jacobiano.

$$V_i \frac{\partial Q_i}{\partial V_i} = Q_i^{cal} - Q_i^{modelo\ de\ carga} - V_i^2 B_{ii}$$

Ecuación 2.50. Cálculo de elementos de la diagonal del Jacobiano.

En la solución del flujo de potencia normal mediante la metodología de Newton – Raphson los valores de las potencias calculadas son constantes, pero al utilizar la metodología de Reestructuración de Potencias Calculadas el efecto del modelo de carga en función del voltaje hace que estas deban actualizarse en cada nueva iteración.

Esta metodología permite incorporar todos los modelos de carga en estado estable mostrados en la sección 1.3.4.2.

2.3.3.1 Diagrama de Flujo

El algoritmo de esta metodología presenta una ligera modificación con respecto al diagrama de flujo mostrado en la sección 2.2.7. Entre los aspectos más relevantes que se pueden observar están que ya no se realiza el cálculo de las potencias especificadas y que en cada nueva iteración se deben recalculan las potencias calculadas, debido a que estas por el efecto de la incorporación de modelos de carga son directamente proporcionales al voltaje actual de la barra.

El diagrama de flujo de esta metodología se muestra a continuación en la Figura 2.4, en este se observa el proceso para calcular un flujo de potencia incorporando modelos de carga mediante la metodología de Reestructuración de Potencias Calculadas, el proceso inicia con la lectura de datos e inicialización de las variables de estado, luego se calcula la matriz de admitancias de barra Y_{barra} . A continuación se inicia el proceso iterativo con el cálculo de las potencias calculadas en las barras P^{cal} y Q^{cal} considerando dentro de estas el efecto del modelo de carga a incorporarse, luego se calculan los errores de

potencia y se chequea el cumplimiento de la tolerancia especificada para el proceso, en caso de no cumplir la convergencia se calcula el Jacobiano y se actualizan los valores de las variables de estado. Este proceso se realiza hasta que los errores de potencia cumplan con la tolerancia establecida, finalmente cuando el proceso haya convergido satisfactoriamente se calculan las potencias en las barras y se imprimen los resultados.

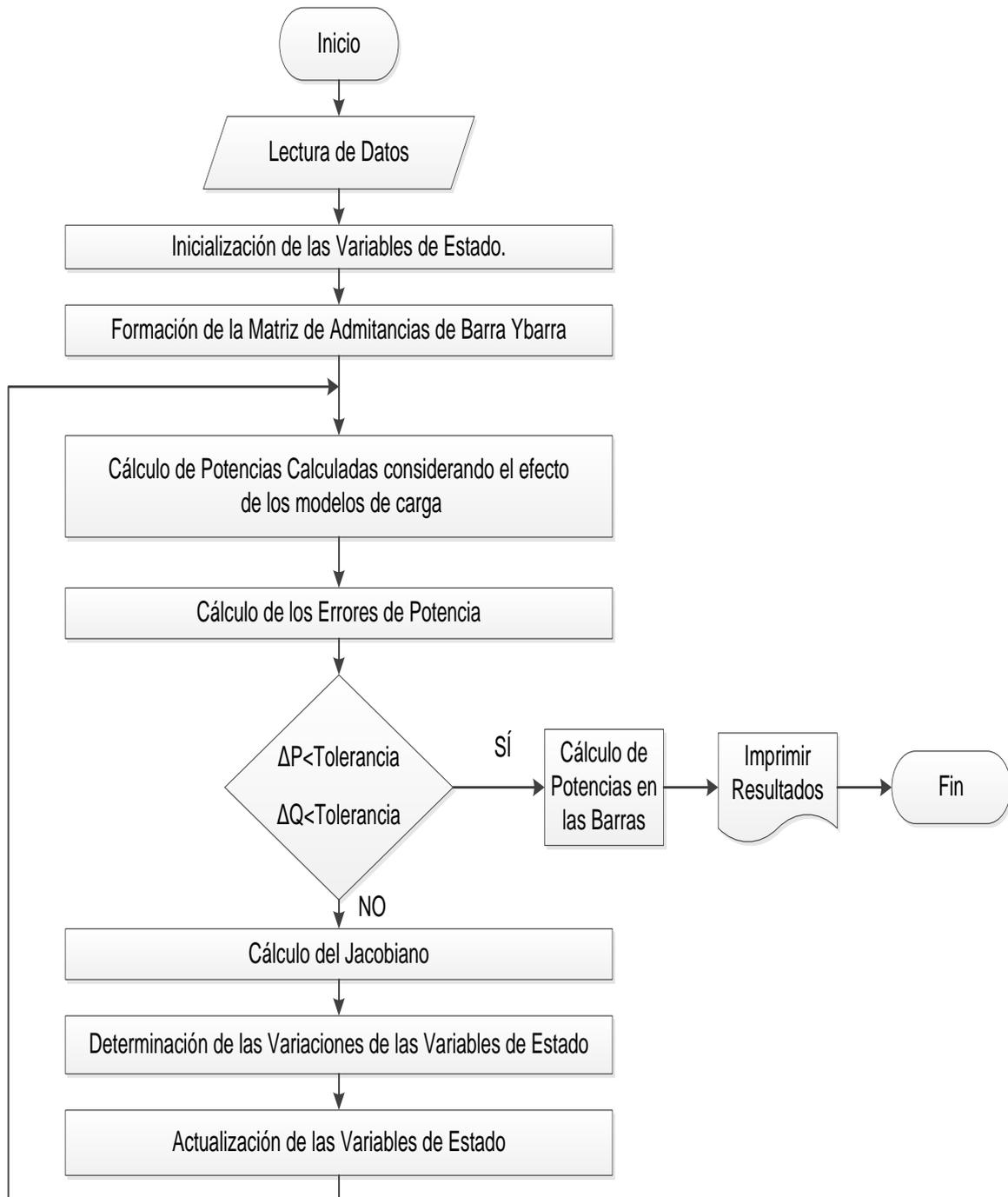


Figura 2.3. Diagrama de Flujo de la Metodología de Reestructuración de Potencias Calculadas.

3. APLICACIÓN, ANÁLISIS Y RESULTADOS

3.1 Introducción

En este capítulo se presenta un ejemplo de flujos de potencia con y sin la incorporación de modelos de carga para un sistema de dos barras. Previo a presentar el ejemplo del flujo de potencia con incorporación de modelos de carga se asumen la composición y los porcentajes de carga en la barra, para presentar un ejemplo de la metodología de agrupación de cargas. En este capítulo también se presentan las características de los sistemas de prueba y los resultados de los flujos de potencia con y sin la incorporación de modelos de carga obtenidos para cada uno de estos.

Finalmente para validar la herramienta computacional desarrollada mediante un análisis de errores se comparan los resultados obtenidos en Python con los resultados obtenidos en los softwares comerciales PSAT y DlgSILENT PowerFactory.

3.2 Ejemplo Aplicado a un Sistema de Dos Barras

En esta sección se presenta un ejemplo aplicado al sistema de dos barras mostrado en la Figura 3.1, primero se muestra el cálculo del flujo de potencia sin incorporar modelos de carga es decir considerando a la carga como un modelo de potencia constante, seguidamente se muestra la incorporación del modelo polinomial dentro del algoritmo de Newton-Raphson utilizando las tres metodologías descritas en la sección 2.3.

3.2.1 Ejemplo de Cálculo del Flujo de Potencia sin considerar la Incorporación de modelos de carga

A continuación se presenta el cálculo del flujo de potencia para el sistema de dos barras considerando a la carga como un modelo de potencia constante, es decir sin la incorporación de modelos de carga.

El proceso a seguir es el mostrado en la sección 2.2.7.

- Se leen e interpretan los datos de entrada:

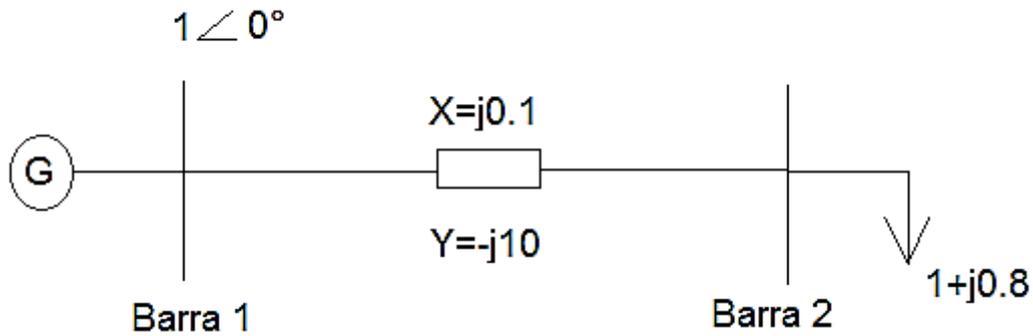


Figura 3.1. Sistema de dos Barras.

Los datos de la Figura 3.1 vienen dados en por unidad razón por lo que se procede directamente a calcular el flujo de potencia aplicando la metodología de Newton - Raphson.

- Se inicializan las variables de estado y se fija la tolerancia deseada:

$$V_2^{(0)} = 1 \text{ p.u.}$$

$$\theta_2^{(0)} = 0^\circ = 0[\text{rad}]$$

$$\text{tolerancia} = 1 * 10^{-3}$$

- Se calcula la matriz de admitancias de barra Y_{barra} :

$$Y_{barra} = \begin{bmatrix} -j10 & j10 \\ j10 & -j10 \end{bmatrix}$$

- Se calculan las potencias especificadas en función al balance generación carga:

$$P_2^{esp} = P_2^{Generación} - P_2^{Carga}$$

$$P_2^{esp} = 0 - 1 \quad \rightarrow \quad P_2^{esp} = -1$$

$$Q_2^{esp} = Q_2^{Generación} - Q_2^{Carga}$$

$$Q_2^{esp} = 0 - 0.8 \quad \rightarrow \quad Q_2^{esp} = -0.8$$

- Se calculan las potencias calculadas en base a las ecuaciones de inyección de potencia:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sen \theta_{ij}) V_j$$

$$P_2^{cal} = 10V_2 \text{sen}\theta_2$$

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \text{sen}\theta_{ij} - B_{ij} \text{cos}\theta_{ij}) V_j$$

$$Q_2^{cal} = 10V_2^2 - 10V_2 \text{cos}\theta_2$$

- Se calculan los errores de potencia:

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} P_2^{esp} - P_2^{cal} \\ P_2^{esp} - Q_2^{cal} \end{bmatrix}$$

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} -1 - 10V_2 \text{sen}\theta_2 \\ -0.8 - 10V_2^2 + 10V_2 \text{cos}\theta_2 \end{bmatrix}$$

- Se evalúan los errores de potencia en los valores iniciales:

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \Delta Q_2^{(0)} \end{bmatrix} = \begin{bmatrix} -1 \\ -0.8 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-1| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.8| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en las condiciones iniciales:

$$[J] = \begin{bmatrix} 10V_2 \text{cos}\theta_2 & 10 \text{sen}\theta_2 \\ 10V_2 \text{sen}\theta_2 & 20V_2 - 10 \text{cos}\theta_2 \end{bmatrix}$$

$$[J]^{(0)} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ -0.8 \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(0)} \\ \Delta V_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.1 \\ -0.08 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta\theta_2^{(0)}$$

$$\theta_2^{(1)} = 0 - 0.1 = -0.1[\text{rad}]$$

$$V_2^{(1)} = V_2^{(0)} + \Delta V_2^{(0)}$$

$$V_2^{(1)} = 1 - 0.08 = 0.92[\text{p. u.}]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(1)} \\ \Delta Q_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.082 \\ -0.11 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-0.082| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.11| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \text{sen}\theta_2 \\ 10V_2 \text{sen}\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(1)} = \begin{bmatrix} 9.15 & -0.998 \\ -0.918 & 8.45 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -0.082 \\ -0.11 \end{bmatrix} = \begin{bmatrix} 9.15 & -0.998 \\ -0.918 & 8.45 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(1)} \\ \Delta V_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.011 \\ -0.014 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(2)} = \theta_2^{(1)} + \Delta\theta_2^{(1)}$$

$$\theta_2^{(2)} = -0.111[\text{rad}]$$

$$V_2^{(2)} = V_2^{(1)} + \Delta V_2^{(1)}$$

$$V_2^{(2)} = 0.906[p. u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(2)} \\ \Delta Q_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0.0036 \\ -0.0041 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.0036| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.0041| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \operatorname{sen}\theta_2 \\ 10V_2 \operatorname{sen}\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(2)} = \begin{bmatrix} 9.004 & -1.108 \\ -1.004 & 8.182 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} 0.0036 \\ -0.0041 \end{bmatrix} = \begin{bmatrix} 9.004 & -1.108 \\ -1.004 & 8.182 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(2)} \\ \Delta V_2^{(2)} \end{bmatrix} = \begin{bmatrix} -0.0003 \\ -0.0005 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(3)} = \theta_2^{(2)} + \Delta\theta_2^{(2)}$$

$$\theta_2^{(3)} = -0.1107[rad]$$

$$V_2^{(3)} = V_2^{(2)} + \Delta V_2^{(2)}$$

$$V_2^{(3)} = 0.9055[p. u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(3)} \\ \Delta Q_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0.000342 \\ -0.000272 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.000342| < 1 * 10^{-3} \rightarrow (V)$$

$$\Delta Q_2 = |-0.000272| < 1 * 10^{-3} \rightarrow (V)$$

- Como todos los valores de los errores de potencia cumplen con la tolerancia, se tiene que la solución son los valores de las variables de estado con los cuales se cumplió dicha condición:

$$\theta_2^{(3)} = -0.1107[\text{rad}] \rightarrow -6.34^\circ$$

$$V_2^{(3)} = 0.9055[\text{p.u.}]$$

- Una vez determinados los valores de magnitud y ángulo de voltaje de todas las barras del sistema se calculan las potencias de envío y recepción de las mismas, así como también las potencias de pérdidas. Para el sistema de dos barras se tiene:

$$\bar{V}_{\text{envío}} = V_1 = 1 \angle 0^\circ$$

$$\bar{V}_{\text{recepción}} = V_2 = 0.9055 \angle -6.34^\circ$$

$$Z_{12} = R_{12} + jX_{12} = j0.1$$

$$\bar{I}_{\text{envío}} = \frac{(\bar{V}_{\text{envío}} - \bar{V}_{\text{recepción}})}{Z_{12}}$$

$$\bar{I}_{12} = \frac{(1 \angle 0^\circ - 0.9055 \angle -6.34^\circ)}{j0.1}$$

$$\bar{I}_{12} = 1 - j1 [\text{p.u.}]$$

$$\bar{I}_{\text{recepción}} = \frac{(\bar{V}_{\text{recepción}} - \bar{V}_{\text{envío}})}{Z_{12}}$$

$$\bar{I}_{21} = \frac{(0.9055 \angle -6.34^\circ - 1 \angle 0^\circ)}{j0.1}$$

$$\bar{I}_{21} = -1 + j1 [\text{p.u.}]$$

$$S_{\text{envío}} = \bar{V}_{\text{envío}} * (\bar{I}_{\text{envío}})^*$$

$$S_{12} = \bar{V}_1 * (\bar{I}_{12})^*$$

$$S_{12} = 1\angle 0^\circ * (1 + j1)$$

$$S_{12} = 1 + j1 [p.u.]$$

$$S_{recepción} = \bar{V}_{recepción} * (\bar{I}_{recepción})^*$$

$$S_{21} = \bar{V}_2 * (\bar{I}_{21})^*$$

$$S_{21} = (0.9055\angle -6.34^\circ) * (-1 - j1)$$

$$S_{21} = -1 - j0.8 [p.u.]$$

$$S_{pérdidas} = S_{12} + S_{21}$$

$$S_{pérdidas} = (1 + j1 - 1 - j0.8)$$

$$S_{pérdidas} = j0.2 [p.u.]$$

3.2.2 Ejemplo de Cálculo del Flujo de Potencia considerando la Incorporación de Modelos de Carga

En esta sección se muestra la incorporación de modelos de carga dentro del cálculo del flujo de potencia mediante la aplicación de las metodologías de Inyección Nodal, Flujos Sucesivos y Reestructuración de Potencias Calculadas, previo a la incorporación de los modelos de carga se describen dos metodología que permiten agrupar cargas y representarlas mediante un modelo equivalente de carga.

Cabe señalar que los parámetros determinados con las metodologías de agrupación de cargas son utilizados en las tres metodologías de incorporación de modelos de carga.

3.2.2.1 Metodología para agrupar cargas

En esta sección se presenta una metodología que permite agrupar distintas cargas mediante un modelo de carga agregada, es decir un modelo equivalente resultante de la incorporación de distintas cargas.

Esta metodología necesita como datos de entrada la composición de carga en cada zona y los porcentajes componentes de cada categoría de carga [7].

El procedimiento a seguir es el siguiente:

- Recopilar los parámetros de las cargas que se quieren agrupar.

- Suponer un porcentaje de carga comercial, residencial e industrial para cada barra, tal que la suma de las tres categorías de carga sumen el 100%.
- Asumir una composición de cada categoría de carga para la barra, es decir se definen los porcentajes en los que actúan las cargas dentro de cada categoría. La suma de la composición de carga total debe ser 100%.
- Calcular Y_j porcentaje del tipo de carga individual en cada barra mediante la siguiente expresión:

$$Y_j = \frac{(X_{aj} * A + X_{bj} * B + X_{cj} * C)}{100}$$

Ecuación 3.1. Porcentaje del tipo de carga individual.

Donde:

Y_j : Porcentaje del tipo de carga individual en cada barra.

X_{aj} : Porcentaje del tipo de carga j en la categoría comercial.

A: Porcentaje de carga comercial en la barra.

X_{bj} : Porcentaje del tipo de carga j en la categoría industrial.

B: Porcentaje de carga industrial en la barra.

X_{cj} : Porcentaje del tipo de carga j en la categoría residencial.

C: Porcentaje de carga residencial en la barra.

- Determinar cada uno de los parámetros del modelo de carga agregado mediante la siguiente expresión:

$$a_z = \sum_{j=1}^N \frac{(Y_j * a_{zj})}{100}$$

Ecuación 3.2. Parámetro equivalente del modelo de carga agregado.

Donde:

a_z : Parámetro resultante del modelo agregado de carga en la barra.

Y_j : Porcentaje del tipo de carga individual en cada barra.

a_{zj} : Parámetro de la carga j.

Cabe señalar que esta metodología es aplicada únicamente a los modelos de carga polinomial, exponencial y lineal, ya que el modelo de motor de inducción necesita otro tipo de análisis para representar la agrupación de varios motores mediante un modelo equivalente.

A continuación haciendo una suposición de los datos de entrada: composición de carga en cada zona y porcentajes de los componentes en cada categoría de carga, se muestra un ejemplo en el cual se agrupan las cargas y se representan mediante un modelo de carga polinomial.

Las cargas que se requieren agrupar son las siguientes:

- Un foco ahorrador.
- Un celular.
- Un DVD.
- Un equipo de sonido.
- Una laptop.
- Un televisor led.
- Un microondas.

Como primer paso se recopilan los parámetros de las cargas que se quieren agrupar. Los parámetros se muestran a continuación en la Tabla 3.1:

Tabla 3.1. Parámetros de las cargas a agruparse [8].

CARGAS DE LA BARRA j	MODELO POLINOMIAL					
	POTENCIA ACTVA			POTENCIA REACTIVA		
	p_1	p_2	p_3	q_1	q_2	q_3
Foco Ahorrador	0,144	0,723	0,133	0,050	0,941	0,009
Celular	-2,121	3,329	-0,207	-1,819	3,045	-0,226
DVD	0,294	-0,405	1,111	0,157	0,055	0,788
Equipo de sonido	-2,747	6,137	-2,390	-2,361	5,579	-2,217
Laptop	0,824	-1,465	1,641	0,829	-1,249	1,419
TV LED	1,257	-2,254	1,997	0,843	-1,603	1,761
Microondas	-1,903	5,088	-2,185	-1,129	3,855	-1,727

Se supone un porcentaje de carga comercial del 47%, un porcentaje de carga residencial del 35% y un porcentaje de carga industrial del 18%.

Se asume la composición de cada categoría de carga, es decir la participación en porcentaje de las distintas cargas en cada categoría. La composición de carga asumida se muestra en la Tabla 3.2:

Tabla 3.2. Composición de carga asumida en cada categoría.

COMPOSICIÓN DE CARGA EN %			
CARGA	COMERCIAL	INDUSTRIAL	RESIDENCIAL
Foco Ahorrador	50	50	10
Celular	5	20	40
DVD	5	0	10
Equipo de sonido	15	25	10
Laptop	10	0	10
TV LED	10	0	10
Microondas	5	5	10

Se calculan los porcentajes del tipo de carga individual en cada barra, así por ejemplo para el foco ahorrador se tiene:

$$Y_{j \text{ foco}} = \frac{(50 * 47 + 50 * 18 + 10 * 35)}{100} = 36$$

Los resultados de las cargas se muestran a continuación en la Tabla 3.3.

Tabla 3.3. Resultados de los porcentajes del tipo de carga individual.

CARGA	Y_j
Foco Ahorrador	36,00
Celular	19,95
DVD	5,85
Equipo de sonido	15,05
Laptop	8,20
TV LED	8,20
Microondas	6,75

Finalmente mediante la aplicación de la Ecuación 3.2 se obtienen los parámetros agregados del modelo de carga polinomial que se muestran a continuación en la Tabla 3.4.

Tabla 3.4. Parámetros agregados para el modelo de carga polinomial.

p_1	p_2	p_3	q_1	q_2	q_3
-0,725	1,863	-0,137	-0,630	1,815	-0,185

3.2.2.2 Metodología para agrupar motores de inducción

En esta sección se presenta una metodología que permite agrupar varios motores de inducción en un modelo de carga equivalente.

Esta metodología necesita como datos de entrada las potencias aparentes y los parámetros eléctricos de los motores de inducción a agruparse [9]. Los parámetros eléctricos de los motores de inducción a agruparse son los mostrados en el circuito equivalente de la sección 1.3.4.2.5.

El procedimiento a seguir es el siguiente:

- Recopilar los parámetros de los motores de inducción que se quieren agrupar.
- Determinar la potencia aparente total del conjunto de motores de inducción a agruparse mediante la Ecuación 3.3:

$$S_{TOTAL} = \sum_{j=1}^n S_j$$

Ecuación 3.3. Potencia Aparente Total.

- Calcular el coeficiente de ponderación para cada motor de inducción, mediante la expresión mostrada en la Ecuación 3.4:

$$\sigma_j = \frac{S_j}{S_{TOTAL}}$$

Ecuación 3.4. Coeficiente de ponderación.

Donde:

σ_j : Coeficiente de ponderación del motor de inducción j.

S_j : Potencia aparente del motor de inducción j.

S_{TOTAL} : Potencia aparente total del grupo de motores de inducción a agruparse.

Cabe señalar que la suma de todos los coeficientes de ponderación debe ser igual a 1.

- Determinar los parámetros del modelo de carga de motor de inducción equivalente mediante la siguiente ecuación:

$$P_{agg} = \sum_{j=1}^n \sigma_j * P_j$$

Ecuación 3.5. Parámetro Agregado del Motor de Inducción.

Donde:

P_{agg} : Parámetro resultante de la agregación de varios motores de inducción.

σ_j : Coeficiente de ponderación.

P_j : Parámetro del motor de inducción j.

A continuación asumiendo un grupo de motores de inducción, se muestra un ejemplo en el cual se agrupan estos motores y se representan mediante un modelo de carga equivalente.

Como primer paso se recopilan los parámetros de los motores de inducción que se quieren agrupar. Los parámetros se muestran a continuación en la Tabla 3.5:

Tabla 3.5. Parámetros de los motores de inducción a agruparse [9].

MOTOR DE INDUCCIÓN	R_s	X_{ys}	X_m	R_{yr}	X_r	s	S
Tipo	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[kVA]
Refrigerador con congelador	0,056	0,087	2,400	0,053	0,082	0,020	206
Lavadora de Platos	0,110	0,140	2,800	0,110	0,065	0,030	43
Secadora	0,120	0,150	1,900	0,130	0,140	0,020	85
Lavadora de Ropa	0,110	0,120	2,000	0,110	0,130	0,040	85
Motor Industrial Tipo 1	0,031	0,100	3,200	0,018	0,180	0,060	313
Motor Industrial Tipo 2	0,013	0,067	3,800	0,009	0,170	0,080	268

Se determina la potencia aparente total del conjunto de motores de inducción a agruparse:

$$S_{TOTAL} = 206 + 43 + 85 + 85 + 313 + 268$$

$$S_{TOTAL} = 1000 [kVA]$$

Se calculan los coeficientes de ponderación para cada motor de inducción, mediante la Ecuación 3.4:

$$\sigma_j = \frac{S_j}{S_{TOTAL}}$$

$$\sigma_{Refrigerador} = \frac{206}{1000} = 0,206$$

Los resultados de los coeficientes de ponderación de los motores de inducción a agruparse se muestran a continuación en la Tabla 3.6:

Tabla 3.6. Coeficientes de ponderación de los motores de inducción a agruparse.

TIPO DE MOTOR DE INDUCCIÓN	σ
Refrigerador con congelador	0,206
Lavadora de Platos	0,043
Secadora	0,085
Lavadora de Ropa	0,085
Motor Industrial 1	0,313
Motor Industrial 2	0,268

Finalmente mediante la aplicación de la Ecuación 3.5 se obtienen los parámetros agregados del modelo de carga de motor de inducción que se muestran a continuación en la Tabla 3.7.

Tabla 3.7. Parámetros agregados para el modelo de carga de motor de inducción.

R_s	X_{ys}	X_m	R_r	X_{yr}	s
0,049	0,096	2,966	0,044	0,145	0,051

3.2.2.3 Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación de Modelos de Carga mediante la Metodología de Inyección Nodal

Se presenta un ejemplo de la solución de un flujo de potencia aplicado a un sistema de dos barras incorporando el modelo de carga polinomial mediante la metodología de

Inyección Nodal explicada en la sección 2.3.1. Los parámetros del modelo de carga polinomial a incorporarse son los mostrados en la Tabla 3.4.

- Se leen e interpretan los datos de entrada:

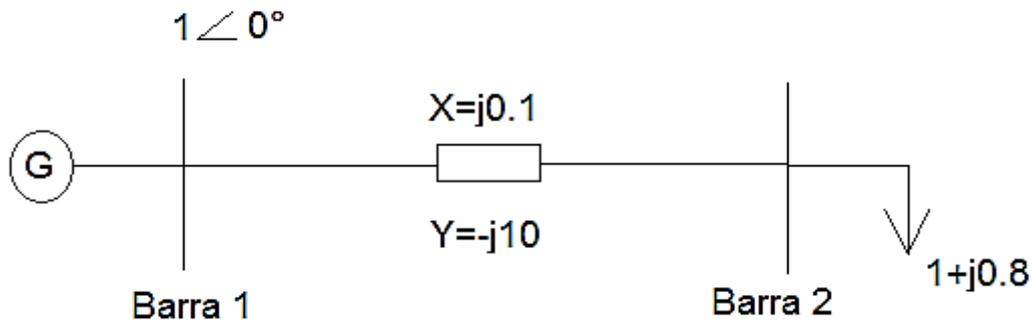


Figura 3.2. Datos del sistema de dos barras.

Los datos de la Figura 3.2 vienen dados en por unidad razón por lo que se procede a calcular el flujo de potencia incorporando el modelo de carga polinomial mediante la metodología de Inyección Nodal.

- Se inicializan las variables de estado y se fija la tolerancia deseada:

$$V_2^{(0)} = 1 \text{ p.u.}$$

$$\theta_2^{(0)} = 0^\circ = 0[\text{rad}]$$

$$\text{tolerancia} = 1 * 10^{-3}$$

- Se calcula la matriz de admitancias de barra Y_{barra} :

$$Y_{barra} = \begin{bmatrix} -j10 & j10 \\ j10 & -j10 \end{bmatrix}$$

- Se calculan las potencias especificadas considerando el modelo de carga polinomial:

$$P_2^{esp} = P_{n2} * \left[p_1 \left(\frac{V}{V_n} \right)^2 + p_2 \left(\frac{V}{V_n} \right) + p_3 \right]$$

$$P_{n2} = P_2^{Generación} - P_2^{Carga}$$

$$P_{n2} = 0 - 1 \rightarrow P_{n2} = -1 \text{ [p.u]}$$

$$P_2^{esp} = -1 * \left[-0.725 \left(\frac{V_2}{1} \right)^2 + 1.863 \left(\frac{V_2}{1} \right) - 0.137 \right]$$

$$P_2^{esp} = 0.725V_2^2 - 1.863V_2 + 0.137$$

$$Q_2^{esp} = Q_{n2} * \left[q_1 \left(\frac{V}{V_n} \right)^2 + q_2 \left(\frac{V}{V_n} \right) + q_3 \right]$$

$$Q_{n2} = Q_2^{Generación} - Q_2^{Carga}$$

$$Q_{n2} = 0 - 0.8 \rightarrow Q_{n2} = -0.8 [p.u]$$

$$Q_2^{esp} = -0.8 * \left[-0.630 \left(\frac{V_2}{1} \right)^2 + 1.815 \left(\frac{V_2}{1} \right) - 0.185 \right]$$

$$Q_2^{esp} = 0.504V_2^2 - 1.452V_2 + 0.148$$

- Se calculan las potencias calculadas en base a las ecuaciones de inyección de potencia:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sen \theta_{ij}) V_j$$

$$P_2^{cal} = 10V_2 \sen \theta_2$$

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \sen \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j$$

$$Q_2^{cal} = 10V_2^2 - 10V_2 \cos \theta_2$$

- Se calculan los errores de potencia:

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} P_2^{esp} - P_2^{cal} \\ P_2^{esp} - Q_2^{cal} \end{bmatrix}$$

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} 0.725V_2^2 - 1.863V_2 + 0.137 - 10V_2 \sen \theta_2 \\ 0.504V_2^2 - 1.452V_2 + 0.148 - 10V_2^2 + 10V_2 \cos \theta_2 \end{bmatrix}$$

- Se evalúan los errores de potencia en los valores iniciales:

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \Delta Q_2^{(0)} \end{bmatrix} = \begin{bmatrix} -1 \\ -0.8 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-1| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.8| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en las condiciones iniciales:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \operatorname{sen}\theta_2 \\ 10V_2 \operatorname{sen}\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(0)} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ -0.8 \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(0)} \\ \Delta V_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.1 \\ -0.08 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta\theta_2^{(0)}$$

$$\theta_2^{(1)} = 0 - 0.1 = -0.1[\operatorname{rad}]$$

$$V_2^{(1)} = V_2^{(0)} + \Delta V_2^{(0)}$$

$$V_2^{(1)} = 1 - 0.08 = 0.92[p. u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(1)} \\ \Delta Q_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.0449 \\ -0.0712 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-0.0449| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.0712| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \operatorname{sen}\theta_2 \\ 10V_2 \operatorname{sen}\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(1)} = \begin{bmatrix} 9.154 & -0.998 \\ -0.918 & 8.45 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -0.0449 \\ -0.0712 \end{bmatrix} = \begin{bmatrix} 9.154 & -1 \\ -0.918 & 8.45 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(1)} \\ \Delta V_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.0059 \\ -0.0091 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(2)} = \theta_2^{(1)} + \Delta\theta_2^{(1)}$$

$$\theta_2^{(2)} = -0.1059[\operatorname{rad}]$$

$$V_2^{(2)} = V_2^{(1)} + \Delta V_2^{(1)}$$

$$V_2^{(2)} = 0.9109[p.u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(2)} \\ \Delta Q_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0.0044 \\ 0.0041 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.0044| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |0.0041| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10\sin\theta_2 \\ 10V_2 \sin\theta_2 & 20V_2 - 10\cos\theta_2 \end{bmatrix}$$

$$[J]^{(2)} = \begin{bmatrix} 9.058 & -1.057 \\ -0.963 & 8.274 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} 0.0044 \\ 0.0041 \end{bmatrix} = \begin{bmatrix} 9.058 & -1.057 \\ -0.963 & 8.274 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(2)} \\ \Delta V_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0.0005 \\ 0.0005 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(3)} = \theta_2^{(2)} + \Delta\theta_2^{(2)}$$

$$\theta_2^{(3)} = -0.1054[\text{rad}]$$

$$V_2^{(3)} = V_2^{(2)} + \Delta V_2^{(2)}$$

$$V_2^{(3)} = 0.9114[\text{p. u.}]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(3)} \\ \Delta Q_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0.0001 \\ 0.0002 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.0001| < 1 * 10^{-3} \rightarrow (V)$$

$$\Delta Q_2 = |0.0002| < 1 * 10^{-3} \rightarrow (V)$$

- Como todos los valores de los errores de potencia cumplen con la tolerancia, se tiene que la solución son los valores de las variables de estado con los cuales se cumplió dicha condición:

$$\theta_2 = -0.1054[\text{rad}] \rightarrow -6.038^\circ$$

$$V_2 = 0.9114[\text{p. u.}]$$

- Una vez determinados los valores de magnitud y ángulo de voltaje de todas las barras del sistema se calculan las potencias de envío y recepción de las mismas, así como también las potencias de pérdidas:

$$\bar{V}_{envio} = V_1 = 1\angle 0^\circ$$

$$\bar{V}_{recepción} = V_2 = 0.9114\angle -6.038^\circ$$

$$Z_{12} = R_{12} + jX_{12} = j0.1$$

$$\bar{I}_{envio} = \frac{(\bar{V}_{envio} - \bar{V}_{recepción})}{Z_{12}}$$

$$\bar{I}_{12} = \frac{(1\angle 0^\circ - 0.9114\angle -6.038^\circ)}{j0.1}$$

$$\bar{I}_{12} = 0.958 - j0.936 [p. u.]$$

$$\bar{I}_{recepción} = \frac{(\bar{V}_{recepción} - \bar{V}_{envio})}{Z_{12}}$$

$$\bar{I}_{21} = \frac{(0.9114\angle -6.038^\circ - 1\angle 0^\circ)}{j0.1}$$

$$\bar{I}_{21} = -0.958 + j0.936 [p. u.]$$

$$S_{envio} = \bar{V}_{envio} * (\bar{I}_{envio})^*$$

$$S_{12} = \bar{V}_1 * (\bar{I}_{12})^*$$

$$S_{12} = 1\angle 0^\circ * (0.958 + j0.936)$$

$$S_{12} = 0.958 + j0.936 [p. u.]$$

$$S_{recepción} = \bar{V}_{recepción} * (\bar{I}_{recepción})^*$$

$$S_{21} = \bar{V}_2 * (\bar{I}_{21})^*$$

$$S_{21} = (0.9114\angle -6.038^\circ) * (-0.958 - j0.936)$$

$$S_{21} = -0.958 - j0.756 [p. u.]$$

$$S_{pérdidas} = S_{12} + S_{21}$$

$$S_{pérdidas} = (0.958 + j0.936 - 0.958 - j0.756)$$

$$S_{pérdidas} = j0.18 [p. u.]$$

3.2.2.4 Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación de Modelos de Carga mediante la Metodología de Flujos Sucesivos

Se presenta un ejemplo de la solución de un flujo de potencia aplicado a un sistema de dos barras incorporando el modelo de carga polinomial mediante la metodología de Flujos Sucesivos explicada en la sección 2.3.2. Los parámetros del modelo de carga polinomial a incorporarse son los mostrados en la Tabla 3.4.

- Se leen e interpretan los datos de entrada:

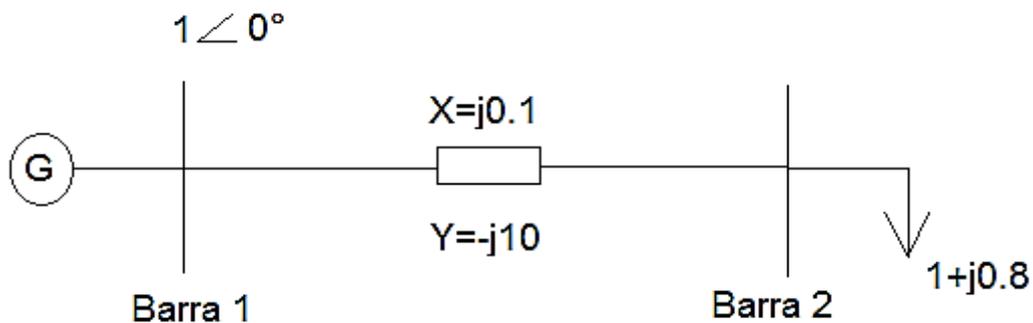


Figura 3.3. Datos del sistema de dos barras.

Los datos de la Figura 3.3 vienen dados en por unidad razón por lo que se procede a calcular el flujo de potencia incorporando el modelo de carga polinomial mediante la metodología de Flujos Sucesivos.

- Se corre un flujo de potencia del sistema sin considerar modelos de carga y con los resultados obtenidos se inicializan las variables de estado y se fijan las tolerancias deseadas. El cálculo del flujo de potencia sin considerar la incorporación de modelos de carga se muestra en la sección 3.2.1, razón por la cual se toman los valores obtenidos del ejemplo mostrado en esta sección para inicializar las variables de estado.

$$V_2^{(0)} = 0.9055 \text{ p.u.}$$

$$\theta_2^{(0)} = -0.1107[\text{rad}]$$

$$\text{tolerancia}_{\text{flujo}} = 1 * 10^{-3}$$

$$\text{tolerancia}_{\text{metodología}} = 1 * 10^{-3}$$

- Se calculan las potencias especificadas considerando el efecto de la incorporación del modelo de carga polinomial.

$$P_2^{esp} = P_{n2} * \left[p_1 \left(\frac{V}{V_n} \right)^2 + p_2 \left(\frac{V}{V_n} \right) + p_3 \right]$$

$$P_{n2} = P_2^{Generación} - P_2^{Carga}$$

$$P_{n2} = 0 - 1 \rightarrow P_{n2} = -1 [p.u]$$

$$P_2^{esp} = -1 * \left[-0.725 \left(\frac{0.9055}{1} \right)^2 + 1.863 \left(\frac{0.9055}{1} \right) - 0.137 \right]$$

$$P_2^{esp} = -0.9555$$

$$Q_2^{esp} = Q_{n2} * \left[q_1 \left(\frac{V}{V_n} \right)^2 + q_2 \left(\frac{V}{V_n} \right) + q_3 \right]$$

$$Q_{n2} = Q_2^{Generación} - Q_2^{Carga}$$

$$Q_{n2} = 0 - 0.8 \rightarrow Q_{n2} = -0.8 [p.u]$$

$$Q_2^{esp} = -0.8 * \left[-0.630 \left(\frac{0.9055}{1} \right)^2 + 1.815 \left(\frac{0.9055}{1} \right) - 0.185 \right]$$

$$Q_2^{esp} = -0.7535$$

- Se calculan las potencias calculadas en base a las ecuaciones de inyección de potencia:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sen \theta_{ij}) V_j$$

$$P_2^{cal} = 10V_2 \sen \theta_2$$

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \sen \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j$$

$$Q_2^{cal} = 10V_2^2 - 10V_2 \cos \theta_2$$

- Se calculan los errores de potencia:

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} P_2^{esp} - P_2^{cal} \\ P_2^{esp} - Q_2^{cal} \end{bmatrix}$$

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} -0.9555 - 10V_2 \text{sen}\theta_2 \\ -0.7535 - 10V_2^2 + 10V_2 \text{cos}\theta_2 \end{bmatrix}$$

- Se evalúan los errores de potencia en los valores iniciales:

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \Delta Q_2^{(0)} \end{bmatrix} = \begin{bmatrix} 0.0448 \\ 0.0468 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.0448| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |0.0468| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los valores iniciales:

$$[J] = \begin{bmatrix} 10V_2 \text{cos}\theta_2 & 10 \text{sen}\theta_2 \\ 10V_2 \text{sen}\theta_2 & 20V_2 - 10 \text{cos}\theta_2 \end{bmatrix}$$

$$[J]^{(0)} = \begin{bmatrix} 9 & -1.105 \\ -1 & 8.171 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} 0.0448 \\ 0.0468 \end{bmatrix} = \begin{bmatrix} 9 & -1.105 \\ -1 & 8.171 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(0)} \\ \Delta V_2^{(0)} \end{bmatrix} = \begin{bmatrix} 0.0058 \\ 0.0064 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta\theta_2^{(0)}$$

$$\theta_2^{(1)} = -0.1049[\text{rad}]$$

$$V_2^{(1)} = V_2^{(0)} + \Delta V_2^{(0)}$$

$$V_2^{(1)} = 0.9119[\text{p. u.}]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(1)} \\ \Delta Q_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.00067 \\ -0.00024 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia establecida para el flujo:

$$\Delta P_2 = |-0.00067| < 1 * 10^{-3} \rightarrow (V)$$

$$\Delta Q_2 = |-0.00024| < 1 * 10^{-3} \rightarrow (V)$$

- Se actualizan los valores iniciales de las variables de estado:

$$V_2^{(0)} = 0.9119 \text{ p.u.}$$

$$\theta_2^{(0)} = -0.1049[\text{rad}]$$

- Se verifica el cumplimiento de la tolerancia establecida para la metodología:

$$V_{it}^{actual} - V_{it}^{anterior} = 0.9119 - 0.9055 = 0.0064 < \text{tolerancia}_{metodología} \rightarrow (F)$$

- Como no se cumple con la tolerancia establecida para la metodología se procede a ejecutar un nuevo flujo de potencia considerando como valores iniciales los valores obtenidos en el cálculo del flujo de potencia anterior, para ello se calculan las potencias especificadas considerando el efecto de la incorporación del modelo de carga polinomial.

$$P_2^{esp} = P_{n2} * \left[p_1 \left(\frac{V}{V_n} \right)^2 + p_2 \left(\frac{V}{V_n} \right) + p_3 \right]$$

$$P_2^{esp} = -1 * \left[-0.725 \left(\frac{0.9119}{1} \right)^2 + 1.863 \left(\frac{0.9119}{1} \right) - 0.137 \right]$$

$$P_2^{esp} = -0.9589$$

$$Q_2^{esp} = Q_{n2} * \left[q_1 \left(\frac{V}{V_n} \right)^2 + q_2 \left(\frac{V}{V_n} \right) + q_3 \right]$$

$$Q_2^{esp} = -0.8 * \left[-0.630 \left(\frac{0.9119}{1} \right)^2 + 1.815 \left(\frac{0.9119}{1} \right) - 0.185 \right]$$

$$Q_2^{esp} = -0.7569$$

- Se calculan las potencias calculadas en base a las ecuaciones de inyección de potencia:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sen \theta_{ij}) V_j$$

$$P_2^{cal} = 10V_2 \sen \theta_2$$

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \sen \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j$$

$$Q_2^{cal} = 10V_2^2 - 10V_2 \cos \theta_2$$

- Se calculan los errores de potencia:

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} P_2^{esp} - P_2^{cal} \\ P_2^{esp} - Q_2^{cal} \end{bmatrix}$$

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} -0.9589 - 10V_2 \sen \theta_2 \\ -0.7569 - 10V_2^2 + 10V_2 \cos \theta_2 \end{bmatrix}$$

- Se evalúan los errores de potencia en los valores iniciales:

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \Delta Q_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.0041 \\ -0.0036 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-0.0041| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.0036| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en las condiciones iniciales:

$$[J] = \begin{bmatrix} 10V_2 \cos \theta_2 & 10 \sen \theta_2 \\ 10V_2 \sen \theta_2 & 20V_2 - 10 \cos \theta_2 \end{bmatrix}$$

$$[J]^{(0)} = \begin{bmatrix} 9.069 & -1.047 \\ -0.995 & 8.293 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -0.0041 \\ -0.0036 \end{bmatrix} = \begin{bmatrix} 9.069 & -1.047 \\ -0.995 & 8.293 \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(0)} \\ \Delta V_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.0005 \\ -0.0005 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta\theta_2^{(0)}$$

$$\theta_2^{(1)} = -0.1054[rad]$$

$$V_2^{(1)} = V_2^{(0)} + \Delta V_2^{(0)}$$

$$V_2^{(1)} = 0.9114[p.u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(1)} \\ \Delta Q_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.00006 \\ -0.00002 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia establecida para el flujo:

$$\Delta P_2 = |-0.00006| < 1 * 10^{-3} \rightarrow (V)$$

$$\Delta Q_2 = |-0.00002| < 1 * 10^{-3} \rightarrow (V)$$

- Se actualizan los valores iniciales de las variables de estado:

$$V_2^{(0)} = 0.9119 p.u.$$

$$\theta_2^{(0)} = -0.1049[rad]$$

- Se verifica el cumplimiento de la tolerancia establecida para la metodología:

$$V_{it}^{actual} - V_{it}^{anterior} = 0.9114 - 0.9119 = -0.0005 < tolerancia_{metodologia} \rightarrow (V)$$

- Como se cumplió con la tolerancia establecida para la metodología, se tiene que la solución son los valores de las variables de estado del último flujo de potencia calculado

$$\theta_2 = -0.1054[rad] \rightarrow -6.038^\circ$$

$$V_2 = 0.9114[p.u.]$$

- Una vez determinados los valores de magnitud y ángulo de voltaje de todas las barras del sistema se calculan las potencias de envío y recepción de las mismas, así como también las potencias de pérdidas:

$$\bar{V}_{envio} = V_1 = 1\angle 0^\circ$$

$$\bar{V}_{recepción} = V_2 = 0,9114\angle - 6.038^\circ$$

$$Z_{12} = R_{12} + jX_{12} = j0.1$$

$$\bar{I}_{envio} = \frac{(\bar{V}_{envio} - \bar{V}_{recepción})}{Z_{12}}$$

$$\bar{I}_{12} = \frac{(1\angle 0^\circ - 0.9114\angle - 6.038^\circ)}{j0.1}$$

$$\bar{I}_{12} = 0.958 - j0.936 [p. u.]$$

$$\bar{I}_{recepción} = \frac{(\bar{V}_{recepción} - \bar{V}_{envio})}{Z_{12}}$$

$$\bar{I}_{21} = \frac{(0.9114\angle - 6.038^\circ - 1\angle 0^\circ)}{j0.1}$$

$$\bar{I}_{21} = -0.958 + j0.936 [p. u.]$$

$$S_{envio} = \bar{V}_{envio} * (\bar{I}_{envio})^*$$

$$S_{12} = \bar{V}_1 * (\bar{I}_{12})^*$$

$$S_{12} = 1\angle 0^\circ * (0.958 + j0.936)$$

$$S_{12} = 0.958 + j0.936 [p. u.]$$

$$S_{recepción} = \bar{V}_{recepción} * (\bar{I}_{recepción})^*$$

$$S_{21} = \bar{V}_2 * (\bar{I}_{21})^*$$

$$S_{21} = (0.9114\angle - 6.038^\circ) * (-0.958 - j0.936)$$

$$S_{21} = -0.958 - j0.756 [p. u.]$$

$$S_{pérdidas} = S_{12} + S_{21}$$

$$S_{pérdidas} = (0.958 + j0.936 - 0.958 - j0.756)$$

$$S_{pérdidas} = j0.18 [p. u.]$$

3.2.2.5 Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación de Modelos de Carga mediante la Metodología de Reestructuración de Potencias Calculadas

Se presenta un ejemplo de la solución de un flujo de potencia aplicado a un sistema de dos barras incorporando el modelo de carga polinomial mediante la metodología de Reestructuración de Potencias Calculadas explicada en la sección 2.3.3. Los parámetros del modelo de carga polinomial a incorporarse son los mostrados en la Tabla 3.4.

- Se leen e interpretan los datos de entrada:

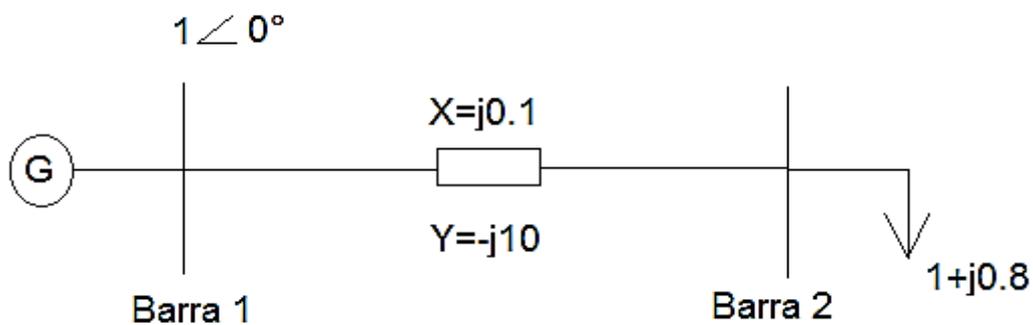


Figura 3.4. Datos del sistema de dos barras.

Los datos de la Figura 3.4 vienen dados en por unidad razón por lo que se procede a calcular el flujo de potencia incorporando el modelo de carga polinomial mediante la metodología de Reestructuración de Potencias Calculadas.

- Se inicializan las variables de estado y se fija la tolerancia deseada:

$$V_2^{(0)} = 1 \text{ p.u.}$$

$$\theta_2^{(0)} = 0^\circ = 0[\text{rad}]$$

$$\text{tolerancia} = 1 * 10^{-3}$$

- Se calcula la matriz de admitancias de barra Y_{barra} :

$$Y_{barra} = \begin{bmatrix} -j10 & j10 \\ j10 & -j10 \end{bmatrix}$$

- A diferencia de las dos metodologías anteriores en las cuales la incorporación del modelo de carga se considera dentro del cálculo de las potencias especificadas, en esta metodología se considera el efecto de la incorporación del modelo de carga dentro de la determinación de las potencias calculadas. Así por ejemplo para incorporar el modelo de carga polinomial:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \operatorname{sen} \theta_{ij}) V_j - P_{n2} * \left[p_1 \left(\frac{V}{V_n} \right)^2 + p_2 \left(\frac{V}{V_n} \right) + p_3 \right]$$

$$P_2^{cal} = 10V_2 \operatorname{sen} \theta_2 - 0.725V_2^2 + 1.863V_2 - 0.137$$

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \operatorname{sen} \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j - Q_{n2} * \left[q_1 \left(\frac{V}{V_n} \right)^2 + q_2 \left(\frac{V}{V_n} \right) + q_3 \right]$$

$$Q_2^{cal} = 10V_2^2 - 10V_2 \cos \theta_2 - 0.504V_2^2 + 1.452V_2 - 0.148$$

- Se calculan los errores de potencia:

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} -P_2^{cal} \\ -Q_2^{cal} \end{bmatrix}$$

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} -10V_2 \operatorname{sen} \theta_2 + 0.725V_2^2 - 1.863V_2 + 0.137 \\ -9.496V_2^2 + 10V_2 \cos \theta_2 - 1.452V_2 + 0.148 \end{bmatrix}$$

- Se evalúan los errores de potencia en los valores iniciales:

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \Delta Q_2^{(0)} \end{bmatrix} = \begin{bmatrix} -1 \\ -0.8 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-1| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.8| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en las condiciones iniciales:

$$[J] = \begin{bmatrix} 10V_2 \cos \theta_2 & 10 \operatorname{sen} \theta_2 - 1.45V_2 + 1.863 \\ 10V_2 \operatorname{sen} \theta_2 & 18.992V_2 - 10 \cos \theta_2 + 1.452 \end{bmatrix}$$

$$[J]^{(0)} = \begin{bmatrix} 10 & 0.413 \\ 0 & 10.444 \end{bmatrix}$$

Cabe señalar que a diferencia de las dos metodologías anteriores el Jacobiano en esta metodología presenta una variación en los términos de las barras de carga, ya que al incorporar el modelo de carga estas barras se modelan como funciones dependientes del voltaje.

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ -0.8 \end{bmatrix} = \begin{bmatrix} 10 & 0.413 \\ 0 & 10.444 \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta \theta_2^{(0)} \\ \Delta V_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.0968 \\ -0.0765 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta \theta_2^{(0)}$$

$$\theta_2^{(1)} = -0.0968[\text{rad}]$$

$$V_2^{(1)} = V_2^{(0)} + \Delta V_2^{(0)}$$

$$V_2^{(1)} = 0.9235[\text{p.u.}]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(1)} \\ \Delta Q_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.0726 \\ -0.0989 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-0.0726| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.0989| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos \theta_2 & 10 \sin \theta_2 - 1.45V_2 + 1.863 \\ 10V_2 \sin \theta_2 & 18.992V_2 - 10 \cos \theta_2 + 1.452 \end{bmatrix}$$

$$[J]^{(1)} = \begin{bmatrix} 9.192 & -0.443 \\ -0.893 & 9.038 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -0.0726 \\ -0.0998 \end{bmatrix} = \begin{bmatrix} 9.192 & -0.443 \\ -0.893 & 9.038 \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta \theta_2^{(1)} \\ \Delta V_2^{(1)} \end{bmatrix} = \begin{bmatrix} -0.0086 \\ -0.0120 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(2)} = \theta_2^{(1)} + \Delta \theta_2^{(1)}$$

$$\theta_2^{(2)} = -0.1054[rad]$$

$$V_2^{(2)} = V_2^{(1)} + \Delta V_2^{(1)}$$

$$V_2^{(2)} = 0.9114[p. u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(2)} \\ \Delta Q_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0.0001 \\ 0.0002 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.0044| < 1 * 10^{-3} \rightarrow (V)$$

$$\Delta Q_2 = |0.0041| < 1 * 10^{-3} \rightarrow (V)$$

- Como todos los valores de los errores de potencia cumplen con la tolerancia, se tiene que la solución son los valores de las variables de estado con los cuales se cumplió dicha condición:

$$\theta_2 = -0.1054[rad] \rightarrow -6.038^\circ$$

$$V_2 = 0.9114[p. u.]$$

- Una vez determinados los valores de magnitud y ángulo de voltaje de todas las barras del sistema se calculan las potencias de envío y recepción de las mismas, así como también las potencias de pérdidas:

$$\bar{V}_{envio} = V_1 = 1 \angle 0^\circ$$

$$\bar{V}_{recepción} = V_2 = 0.9114\angle - 6.038^\circ$$

$$Z_{12} = R_{12} + jX_{12} = j0.1$$

$$\bar{I}_{envío} = \frac{(\bar{V}_{envío} - \bar{V}_{recepción})}{Z_{12}}$$

$$\bar{I}_{12} = \frac{(1\angle 0^\circ - 0.9114\angle - 6.038^\circ)}{j0.1}$$

$$\bar{I}_{12} = 0.958 - j0.936 [p. u.]$$

$$\bar{I}_{recepción} = \frac{(\bar{V}_{recepción} - \bar{V}_{envío})}{Z_{12}}$$

$$\bar{I}_{21} = \frac{(0.9114\angle - 6.038^\circ - 1\angle 0^\circ)}{j0.1}$$

$$\bar{I}_{21} = -0.958 + j0.936 [p. u.]$$

$$S_{envío} = \bar{V}_{envío} * (\bar{I}_{envío})^*$$

$$S_{12} = \bar{V}_1 * (\bar{I}_{12})^*$$

$$S_{12} = 1\angle 0^\circ * (0.958 + j0.936)$$

$$S_{12} = 0.958 + j0.936 [p. u.]$$

$$S_{recepción} = \bar{V}_{recepción} * (\bar{I}_{recepción})^*$$

$$S_{21} = \bar{V}_2 * (\bar{I}_{21})^*$$

$$S_{21} = (0.9114\angle - 6.038^\circ) * (-0.958 - j0.936)$$

$$S_{21} = -0.958 - j0.756 [p. u.]$$

$$S_{pérdidas} = S_{12} + S_{21}$$

$$S_{pérdidas} = (0.958 + j0.936 - 0.958 - j0.756)$$

$$S_{pérdidas} = j0.18 [p. u.]$$

3.2.2.6 Ejemplo de Cálculo de Flujo de Potencia considerando la Incorporación del Modelo de Carga de Motor de Inducción

En esta sección se presenta un ejemplo de la solución de un flujo de potencia aplicado a un sistema de dos barras incorporando el modelo de carga de motor de inducción mediante la metodología de Inyección Nodal explicada en la sección 2.3.1. Los parámetros del modelo de carga de motor de inducción a incorporarse son los mostrados en la Tabla 3.7.

- Se leen e interpretan los datos de entrada:

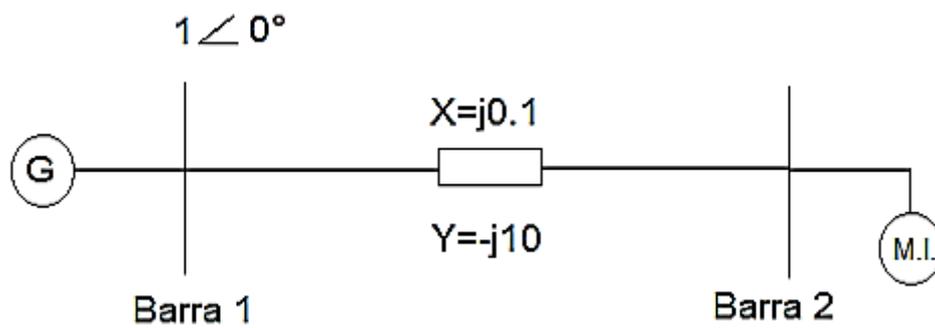


Figura 3.5. Datos del sistema de dos barras.

Los datos de la Figura 3.5 vienen dados en por unidad razón por lo que se procede a calcular el flujo de potencia incorporando el modelo de carga de motor de inducción mediante la metodología de Inyección Nodal.

- Se inicializan las variables de estado y se fija la tolerancia deseada:

$$V_2^{(0)} = 1 \text{ p.u.}$$

$$\theta_2^{(0)} = 0^\circ = 0[\text{rad}]$$

$$\text{tolerancia} = 1 * 10^{-3}$$

- Se calcula la matriz de admitancias de barra Y_{barra} :

$$Y_{barra} = \begin{bmatrix} -j10 & j10 \\ j10 & -j10 \end{bmatrix}$$

- Se calculan las potencias especificadas considerando el modelo de carga de motor de inducción mostrado en la sección 1.3.4.2.5:

$$P_2^{esp} = -\left(R_s + \frac{R_r}{s}\right) * \frac{V_2^2}{\left(R_s + \frac{R_r}{s}\right)^2 + (X_{ys} + X_{yr})^2}$$

$$P_2^{esp} = -\left(0.049 + \frac{0.044}{0.051}\right) * \frac{V_2^2}{\left(0.049 + \frac{0.044}{0.051}\right)^2 + (0.096 + 0.145)^2}$$

$$P_2^{esp} = -1.0252V_2^2$$

$$Q_2^{esp} = -(X_{ys} + X_{yr}) * \frac{V^2}{\left(R_s + \frac{R_r}{s}\right)^2 + (X_{ys} + X_{yr})^2} + \frac{V^2}{X_m + X_{ys}}$$

$$Q_2^{esp} = -\left[(0.096 + 0.145) * \frac{V^2}{\left(0.049 + \frac{0.044}{0.051}\right)^2 + (0.096 + 0.145)^2} + \frac{V^2}{2.966 + 0.096} \right]$$

$$Q_2^{esp} = -0.5976V_2^2$$

Cabe recalcar que este modelo de carga únicamente es aplicable en barras PQ. En el ejemplo mostrado luego de realizar el balance generación carga aparece el signo menos debido a que la barra 2 no posee generación.

- Se calculan las potencias calculadas en base a las ecuaciones de inyección de potencia:

$$P_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \cos \theta_{ij} + B_{ij} \sen \theta_{ij}) V_j$$

$$P_2^{cal} = 10V_2 \sen \theta_2$$

$$Q_i^{cal} = V_i \sum_{j=1}^n (G_{ij} \sen \theta_{ij} - B_{ij} \cos \theta_{ij}) V_j$$

$$Q_2^{cal} = 10V_2^2 - 10V_2 \cos \theta_2$$

- Se calculan los errores de potencia:

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} P_2^{esp} - P_2^{cal} \\ P_2^{esp} - Q_2^{cal} \end{bmatrix}$$

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} -1.0252V_2^2 - 10V_2 \sen \theta_2 \\ -0.5976V_2^2 - 10V_2^2 + 10V_2 \cos \theta_2 \end{bmatrix}$$

- Se evalúan los errores de potencia en los valores iniciales:

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \Delta Q_2^{(0)} \end{bmatrix} = \begin{bmatrix} -1.0252 \\ -0.5976 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-1.0252| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.5976| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en las condiciones iniciales:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \operatorname{sen}\theta_2 \\ 10V_2 \operatorname{sen}\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(0)} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -1.0252 \\ -0.5976 \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(0)} \\ \Delta V_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.1025 \\ -0.0598 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta\theta_2^{(0)}$$

$$\theta_2^{(1)} = 0 - 0.1025 = -0.1025[\operatorname{rad}]$$

$$V_2^{(1)} = V_2^{(0)} + \Delta V_2^{(0)}$$

$$V_2^{(1)} = 1 - 0.0598 = 0.9402[p. u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(1)} \\ \Delta Q_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0.0558 \\ -0.0154 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.0558| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.0154| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \operatorname{sen}\theta_2 \\ 10V_2 \operatorname{sen}\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(1)} = \begin{bmatrix} 9.35 & -1.02 \\ -0.96 & 8.86 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} 0.0558 \\ -0.0154 \end{bmatrix} = \begin{bmatrix} 9.35 & -1.02 \\ -0.96 & 8.86 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(1)} \\ \Delta V_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0.0226 \\ 0.0007 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(2)} = \theta_2^{(1)} + \Delta\theta_2^{(1)}$$

$$\theta_2^{(2)} = -0.0799[\operatorname{rad}]$$

$$V_2^{(2)} = V_2^{(1)} + \Delta V_2^{(1)}$$

$$V_2^{(2)} = 0.9409[p.u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(2)} \\ \Delta Q_2^{(2)} \end{bmatrix} = \begin{bmatrix} -0.1566 \\ -0.0029 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-0.1566| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |-0.0029| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \operatorname{sen}\theta_2 \\ 10V_2 \operatorname{sen}\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(2)} = \begin{bmatrix} 9.38 & -0.79 \\ -0.75 & 8.85 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} -0.1566 \\ -0.0029 \end{bmatrix} = \begin{bmatrix} 9.38 & -0.79 \\ -0.75 & 8.85 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(2)} \\ \Delta V_2^{(2)} \end{bmatrix} = \begin{bmatrix} -0.0168 \\ -0.0018 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(3)} = \theta_2^{(2)} + \Delta\theta_2^{(2)}$$

$$\theta_2^{(3)} = -0.0967[\operatorname{rad}]$$

$$V_2^{(3)} = V_2^{(2)} + \Delta V_2^{(2)}$$

$$V_2^{(3)} = 0.9391[p.u.]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(3)} \\ \Delta Q_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0.0026 \\ 0.0010 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |0.0026| < 1 * 10^{-3} \rightarrow (F)$$

$$\Delta Q_2 = |0.0010| < 1 * 10^{-3} \rightarrow (F)$$

- Como los errores de potencia no cumplen con la tolerancia especificada se calcula el Jacobiano y se lo evalúa en los nuevos valores de las variables de estado:

$$[J] = \begin{bmatrix} 10V_2 \cos\theta_2 & 10 \sin\theta_2 \\ 10V_2 \sin\theta_2 & 20V_2 - 10 \cos\theta_2 \end{bmatrix}$$

$$[J]^{(3)} = \begin{bmatrix} 9.35 & -0.97 \\ -0.91 & 8.83 \end{bmatrix}$$

- Se determinan las variaciones de las variables de estado:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} 0.0026 \\ 0.0010 \end{bmatrix} = \begin{bmatrix} 9.35 & -0.97 \\ -0.91 & 8.83 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta_2^{(3)} \\ \Delta V_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0.0003 \\ 0.0001 \end{bmatrix}$$

- Se actualizan los valores de las variables de estado:

$$\theta_2^{(4)} = \theta_2^{(3)} + \Delta\theta_2^{(3)}$$

$$\theta_2^{(4)} = -0.0964[\text{rad}]$$

$$V_2^{(4)} = V_2^{(3)} + \Delta V_2^{(3)}$$

$$V_2^{(4)} = 0.9392[\text{p.u.}]$$

- Se evalúan los errores de potencia con los valores actualizados de las variables de estado:

$$\begin{bmatrix} \Delta P_2^{(4)} \\ \Delta Q_2^{(4)} \end{bmatrix} = \begin{bmatrix} -0.0003 \\ 0.0003 \end{bmatrix}$$

- Se verifica el cumplimiento de la tolerancia:

$$\Delta P_2 = |-0.0003| < 1 * 10^{-3} \rightarrow (V)$$

$$\Delta Q_2 = |0.0003| < 1 * 10^{-3} \rightarrow (V)$$

- Como todos los valores de los errores de potencia cumplen con la tolerancia, se tiene que la solución son los valores de las variables de estado con los cuales se cumplió dicha condición:

$$\theta_2 = -0.0964[\text{rad}] \rightarrow -5.523^\circ$$

$$V_2 = 0.9392[\text{p.u.}]$$

- Una vez determinados los valores de magnitud y ángulo de voltaje de todas las barras del sistema se calculan las potencias de envío y recepción de las mismas, así como también las potencias de pérdidas:

$$\bar{V}_{envio} = V_1 = 1\angle 0^\circ$$

$$\bar{V}_{recepción} = V_2 = 0.9392\angle -5.523^\circ$$

$$Z_{12} = R_{12} + jX_{12} = j0.1$$

$$\bar{I}_{envio} = \frac{(\bar{V}_{envio} - \bar{V}_{recepción})}{Z_{12}}$$

$$\bar{I}_{12} = \frac{(1\angle 0^\circ - 0.9392\angle -5.523^\circ)}{j0.1}$$

$$\bar{I}_{12} = 0.904 - j0.652 [p. u.]$$

$$\bar{I}_{recepción} = \frac{(\bar{V}_{recepción} - \bar{V}_{envio})}{Z_{12}}$$

$$\bar{I}_{21} = \frac{(0.9392\angle -5.523^\circ - 1\angle 0^\circ)}{j0.1}$$

$$\bar{I}_{21} = -0.904 + j0.652 [p. u.]$$

$$S_{envio} = \bar{V}_{envio} * (\bar{I}_{envio})^*$$

$$S_{12} = \bar{V}_1 * (\bar{I}_{12})^*$$

$$S_{12} = 1\angle 0^\circ * (0.904 + j0.652)$$

$$S_{12} = 0.904 + j0.652 [p. u.]$$

$$S_{recepción} = \bar{V}_{recepción} * (\bar{I}_{recepción})^*$$

$$S_{21} = \bar{V}_2 * (\bar{I}_{21})^*$$

$$S_{21} = (0.9392\angle -5.523^\circ) * (-0.904 - j0.652)$$

$$S_{21} = -0.904 - j0.528 [p. u.]$$

$$S_{pérdidas} = S_{12} + S_{21}$$

$$S_{pérdidas} = (0.904 + j0.652 - 0.904 - j0.528)$$

$$S_{pérdidas} = j0.124 [p. u.]$$

Cabe recalcar que esta sección únicamente se presenta el ejemplo de la incorporación del modelo de carga de motor de inducción mediante la metodología de Inyección Nodal, puesto que como se muestra en las secciones 3.2.2.2, 3.2.2.3 y 3.2.2.4; al utilizar cualquiera de las tres metodologías Inyección Nodal, Flujos Sucesivos o Reestructuración de Potencias Calculadas los resultados obtenidos son los mismos.

3.3 Presentación de los Sistemas de Prueba

Los sistemas de prueba utilizados son redes eléctricas estandarizadas encontradas en libros y publicados en fuentes confiables. Estas redes sirven de referencia para realizar una comparación cuantitativa con los resultados arrojados por el programa.

Las redes utilizadas fueron las siguientes:

- Dos Sistemas de 2 barras tomados del cuaderno de la cátedra de Introducción a Sistemas Eléctricos de Potencia dictada en la Escuela Politécnica Nacional.
- Sistema de 4 barras del libro “Power System Analysis” de Grainger y Stevenson.
- Sistema de 14 barras de la revista IEEE.
- Sistema de 30 barras de la revista IEEE.

A continuación se presenta una breve descripción de los sistemas de prueba utilizados para validar el programa desarrollado en el presente trabajo de titulación.

- *Sistema de 2 barras (Línea R-L):*

Este sistema está conformado por un generador sincrónico, dos barras, una línea de transmisión y una carga. La principal característica que presenta este sistema es que la línea de transmisión que conecta el generador con la carga es modelada mediante una resistencia y una reactancia que representa una línea de transmisión de longitud media.

En la Figura 3.6 se presenta el sistema de dos barras descrito, los sistemas modelados en los softwares comerciales se muestran en el Anexo III.

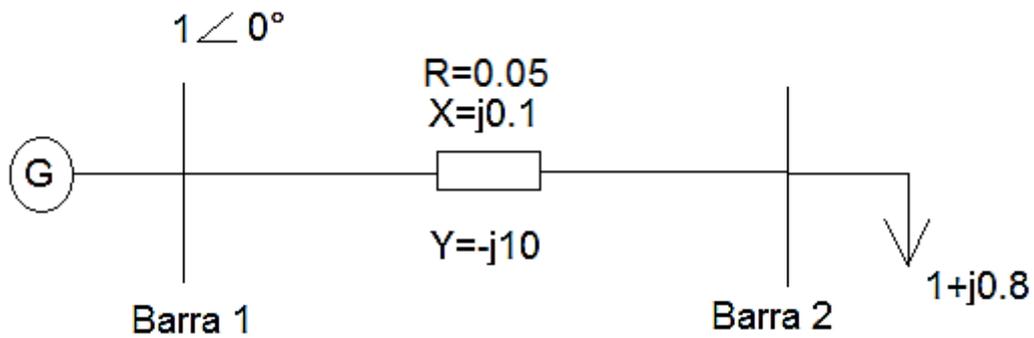


Figura 3.6. Sistema de 2 barras.

- *Sistema de 2 barras (Línea L):*

Este sistema está conformado por un generador sincrónico, dos barras, una línea de transmisión y una carga. La principal característica que presenta este sistema es que la línea de transmisión que conecta el generador con la carga es modelada únicamente mediante una reactancia que representa una línea de transmisión de longitud corta.

En la Figura 3.7 se presenta el sistema de dos barras descrito, los sistemas modelados en los softwares comerciales se muestran en el Anexo III.

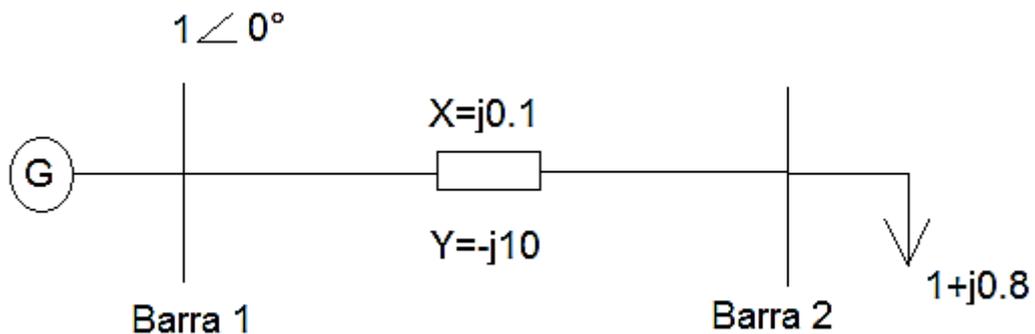


Figura 3.7. Sistema de 2 barras.

- *Sistema de 4 barras:*

El sistema de 4 barras mostrado en el libro "Power System Analysis" de Grainger y Stevenson, está compuesto por dos generadores, cuatro líneas de transmisión y cuatro cargas. En la Figura 3.8 se presenta el sistema de cuatro barras descrito, los sistemas modelados en los softwares comerciales se muestran en el Anexo III.

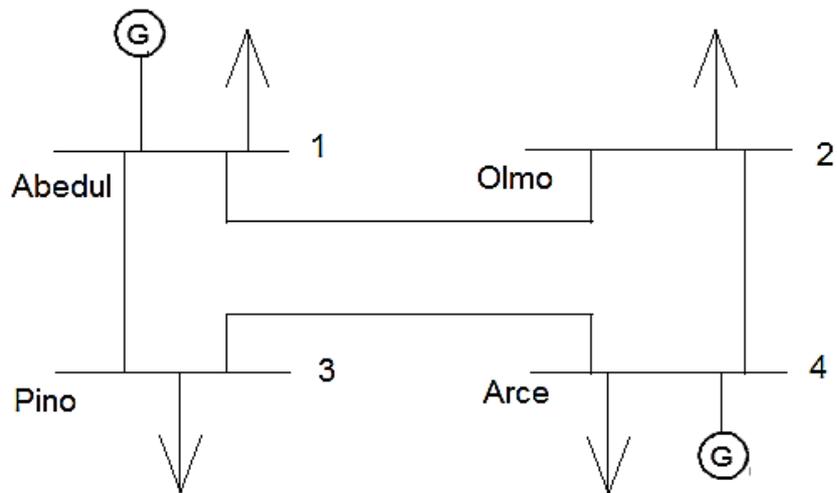


Figura 3.8. Sistema de 4 barras.

- *Sistema IEEE de 14 barras:*

El sistema IEEE de 14 barras está compuesto por dos generadores, catorce barras, tres compensadores sincrónicos, dos transformadores de dos devanados, un transformador de tres devanados, quince líneas de transmisión y un capacitor en paralelo. Este sistema es uno de los más utilizados para comprender apropiadamente los efectos que se producen en las redes eléctricas puesto que cuenta con todos los elementos eléctricos que conforman un sistema eléctrico de potencia. La Figura 3.9 muestra un bosquejo de este sistema en donde se puede apreciar que el transformador de tres devanados se modela como una impedancia conectada en estrella. Los sistemas modelados en los softwares comerciales se muestran en el Anexo III.

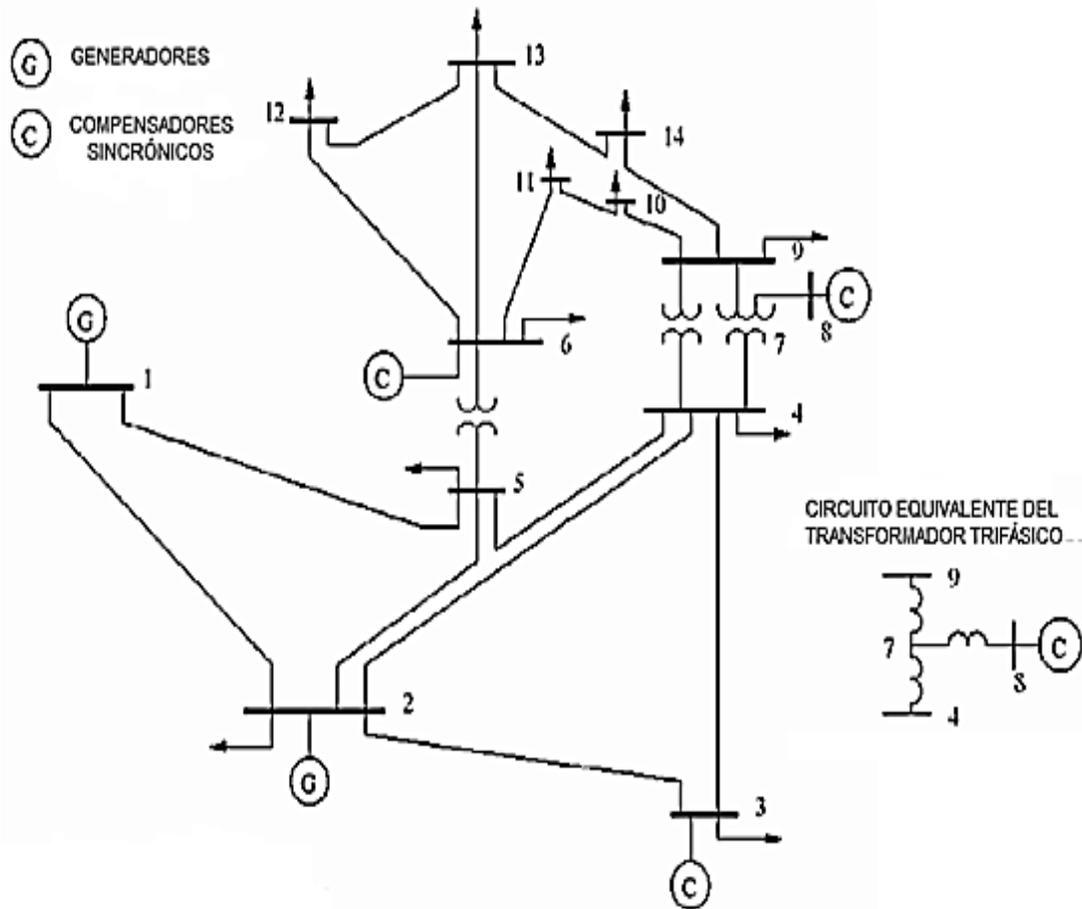


Figura 3.9. Sistema IEEE de 14 barras.

- *Sistema IEEE de 30 barras:*

El sistema IEEE de 30 barras está compuesto por seis generadores, 22 cargas, treinta barras, dos compensadores sincrónicos, tres transformadores de dos devanados, dos transformadores de tres devanados y treinta y cinco líneas de transmisión. Este sistema junto al de 14 barras son los más utilizados para comprender apropiadamente los efectos que se producen en las redes eléctricas puesto que cuenta con todos los elementos eléctricos que conforman un sistema eléctrico de potencia. La Figura 3.10 muestra un bosquejo de este sistema, donde se puede apreciar que los transformadores de tres devanados se modelan como una impedancia conectada en estrella. Los sistemas modelados en los softwares comerciales se muestran en el Anexo III.

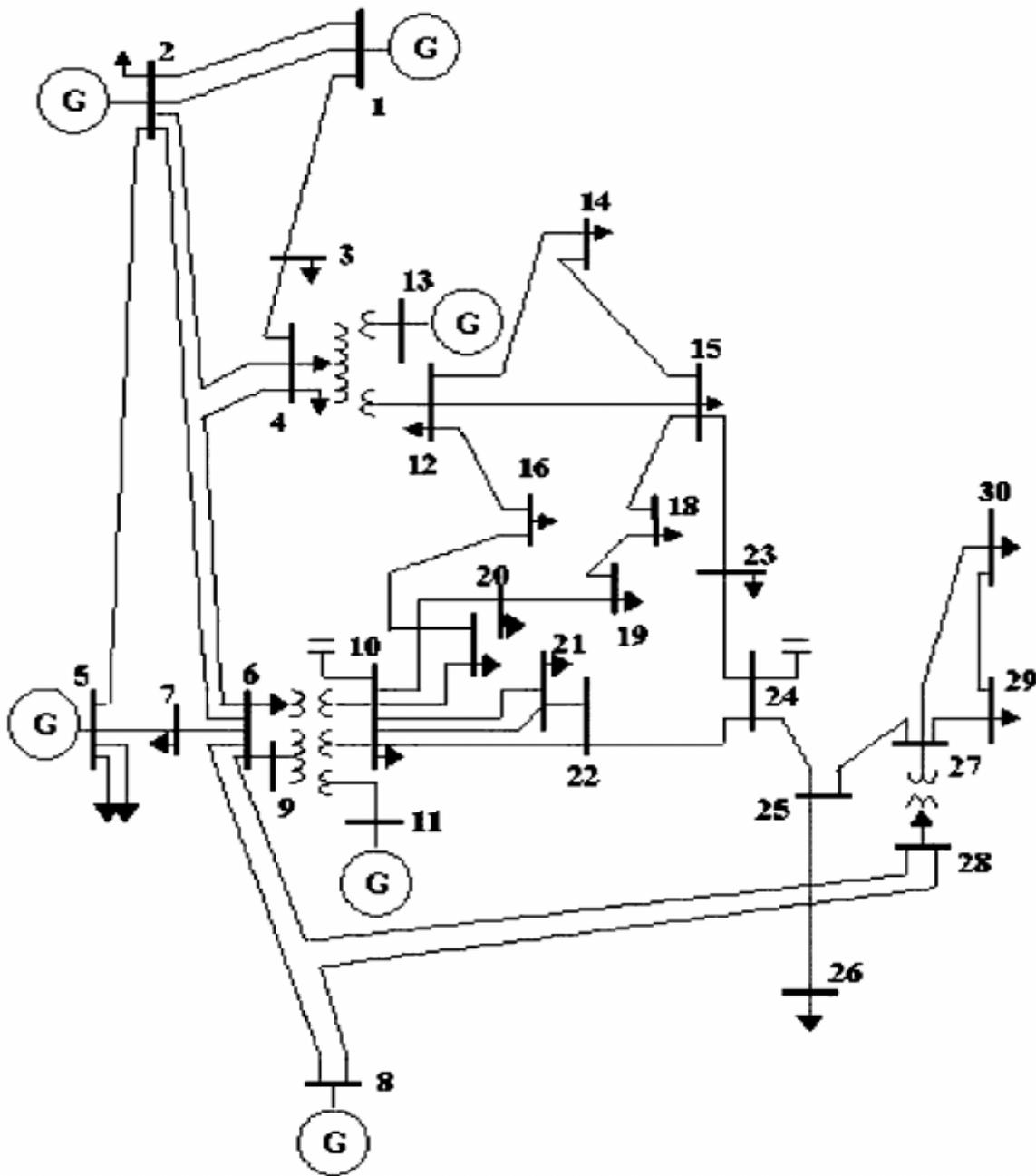


Figura 3.10. Sistema IEEE de 30 barras.

3.4 Resultados obtenidos al Incorporar Modelos de Carga en estado estable al cálculo de Flujos Potencia

A continuación se presentan los resultados obtenidos para cada uno de los sistemas de prueba descritos en la sección 3.3.

3.4.1 Resultados obtenidos en los sistemas de 2 barras

- *Sistema de 2 barras (Línea R-L):*

En las Tablas 3.8 y 3.9 se muestran los resultados de magnitud y ángulo de voltaje obtenidos mediante el programa desarrollado en Python, seguidamente en las Tablas 3.10, 3.11 y 3.12 se muestran los resultados obtenidos en los softwares comerciales DlgSILENT PowerFactory y PSAT. Finalmente en la Tabla 3.13 se muestran las potencias activas y reactivas de envío y recepción de cada una de las ramas del sistema, así como también las potencias activas y reactivas de pérdidas.

Tabla 3.8. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9264	-2,1651	0,9295	-2,1025	0,9309	-2,0532
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	9		11		11	
Tiempo [s]:	0,05		0,06		0,05	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	9		40		44	
Tiempo [s]:	0,05		0,20		0,21	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						
Iteraciones:	9		9		9	
Tiempo [s]:	0,05		0,11		0,12	

Tabla 3.9. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9264	-2,1651	0,9358	-1,8832	0,8507	-4,4266
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	9		14		53	
Tiempo [s]:	0,05		0,08		0,28	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	9		59		194	
Tiempo [s]:	0,05		0,28		0,93	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						

Iteraciones:	9	11	19
Tiempo [s]:	0,05	0,11	0,11

Tabla 3.10. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DlgSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9264	-2,1651	0,9295	-2,0996	0,9309	-2,0532

Tabla 3.11. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DlgSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9264	-2,1651	0,9358	-1,8832	0,8507	-4,4266

Tabla 3.12. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PSAT.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PSAT				
BARRA	MODELO PQ		MODELO EXPONENCIAL	
	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000
2	0,9264	-2,1651	0,9309	-2,0532

Tabla 3.13. Potencias de las Barras.

RESULTADOS DE POTENCIAS EN LAS BARRAS						
RAMA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	[MW]	[MVar]	[MW]	[MVar]	[MW]	[MVar]
DE	HASTA	MODELO PQ				

1	2	57,694	45,389	-55,000	-40,000	2,694	5,389
DE	HASTA	MODELO POLINOMIAL					
1	2	55,745	43,290	-53,254	-38,309	2,491	4,982
DE	HASTA	MODELO EXPONENCIAL					
1	2	54,549	42,393	-52,163	-37,620	2,386	4,773
DE	HASTA	MODELO LINEAL					
1	2	50,476	39,444	-48,424	-35,341	2,052	4,104
DE	HASTA	MODELO MOTOR DE INDUCCIÓN					
1	2	113,272	95,232	-102,322	-73,332	10,950	21,900

A continuación en la Fig. 3.11 se observan los perfiles de voltaje obtenidos para los distintos modelos de carga simulados.

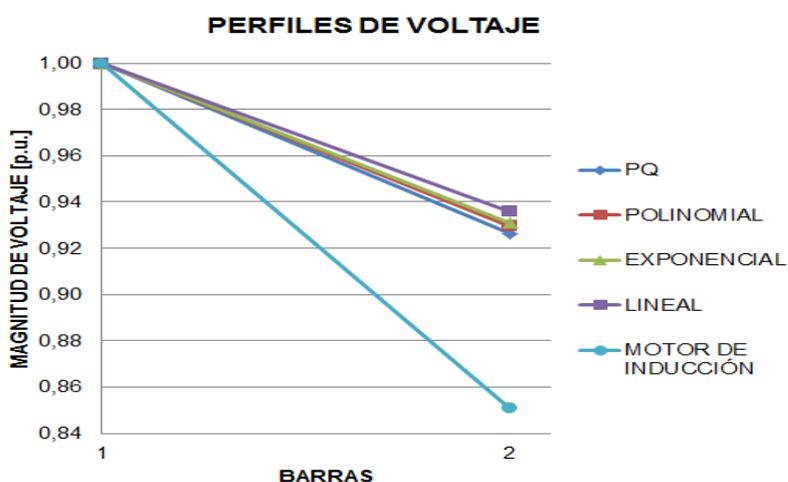


Figura 3.11. Perfiles de Voltaje del Sistema de 2 barras (Línea R-L).

- *Sistema de 2 barras (Línea L):*

En las Tablas 3.14 y 3.15 se muestran los resultados de magnitud y ángulo de voltaje obtenidos mediante el programa desarrollado en Python, seguidamente en las Tablas 3.16, 3.17 y 3.18 se muestran los resultados obtenidos en los softwares comerciales DlgSILENT PowerFactory y PSAT. Finalmente en la Tabla 3.19 se muestran las potencias activas y reactivas de envío y recepción de cada una de las ramas del sistema, así como también las potencias activas y reactivas de pérdidas.

Tabla 3.14. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ

#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9055	-6,3402	0,9114	-6,0381	0,9137	-5,8759
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	11		13		13	
Tiempo [s]:	0,06		0,08		0,07	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	11		51		55	
Tiempo [s]:	0,06		0,24		0,25	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						
Iteraciones:	11		12		12	
Tiempo [s]:	0,06		0,13		0,11	

Tabla 3.15. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9055	-6,3402	0,9213	-5,3147	0,9006	-7,3161
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	11		16		21	
Tiempo [s]:	0,06		0,10		0,11	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	11		77		90	
Tiempo [s]:	0,06		0,38		0,43	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						
Iteraciones:	11		13		21	
Tiempo [s]:	0,06		0,13		0,26	

Tabla 3.16. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DlgSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9055	-6,3402	0,9114	-6,0328	0,9137	-5,8759

Tabla 3.17. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DlgSILENT POWERFACTORY						
# BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9055	-6,3402	0,9213	-5,3147	0,9006	-7,3161

Tabla 3.18. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PSAT.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PSAT				
BARRA	MODELO PQ		MODELO EXPONENCIAL	
	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000
2	0,9055	-6,3402	0,9137	-5,8759

Tabla 3.19. Potencias de las Barras.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q	
#	[MW]	[MVar]	[MW]	[MVar]	[MW]	[MVar]	
DE	HASTA	MODELO PQ					
1	2	100,000	100,000	100,000	80,000	0,000	20,000
DE	HASTA	MODELO POLINOMIAL					
1	2	95,873	93,630	-95,873	-75,672	0,000	17,958
DE	HASTA	MODELO EXPONENCIAL					
1	2	93,540	91,094	-93,540	-74,046	0,000	17,048
DE	HASTA	MODELO LINEAL					
1	2	85,333	82,688	-85,333	-68,569	0,000	14,119
DE	HASTA	MODELO MOTOR DE INDUCCIÓN					
1	2	114,685	106,738	-114,685	-82,192	0,000	24,546

A continuación en la Fig. 3.12 se observan los perfiles de voltaje obtenidos para los distintos modelos de carga simulados.

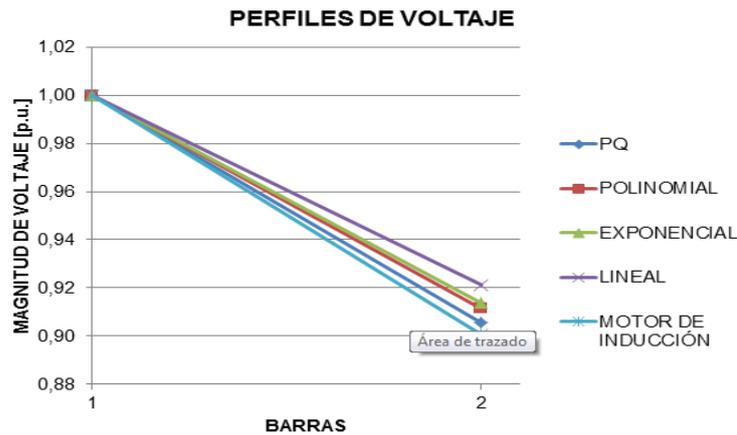


Figura 3.12. Perfiles de Voltaje del Sistema de 2 barras (Línea L).

Al comparar los resultados de los sistemas de dos barras se observa que los resultados obtenidos tanto con el programa desarrollado en Python como los resultados obtenidos en los softwares comerciales PSAT y DlgSILENT PowerFactory son prácticamente los mismos, ya que únicamente existe un error menor al 1% en la barra 2 al incorporar el modelo de carga polinomial, este error puede haberse presentado debido al número de decimales y a la tolerancia utilizada por cada software en el cálculo del flujo de potencia.

3.4.2 Resultados obtenidos en los sistemas de 4 barras

En las Tablas 3.20 y 3.21 se muestran los resultados de magnitud y ángulo de voltaje obtenidos mediante el programa desarrollado en Python, seguidamente en las Tablas 3.22 y 3.23 se muestran los resultados obtenidos en el software comercial DlgSILENT PowerFactory. Finalmente en la Tabla 3.24 se muestran las potencias activas y reactivas de envío y recepción de cada una de las ramas del sistema, así como también las potencias activas y reactivas de pérdidas.

Cabe mencionar que en este sistema no se realiza la comparación con PSAT puesto que este software no permite incorporar carga a una barra Slack.

Tabla 3.20. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9824	-0,9761	0,9828	-0,9110	0,9832	-0,8730
3	0,9690	-1,8722	0,9704	-1,7695	0,9699	-1,7809

4	1,0200	1,5231	1,0200	1,6083	1,0200	1,6290
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	8		9		9	
Tiempo [s]:	0,14		0,13		0,12	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	8		35		35	
Tiempo [s]:	0,14		0,46		0,47	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						
Iteraciones:	8		9		9	
Tiempo [s]:	0,14		0,26		0,23	

Tabla 3.21. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9824	-0,9761	0,9838	-0,7535	0,9855	0,1170
3	0,9690	-1,8722	0,9701	-1,6940	0,9793	-0,5894
4	1,0200	1,5231	1,0200	1,7409	1,0200	2,7340
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	8		10		9	
Tiempo [s]:	0,14		0,15		0,14	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	8		39		40	
Tiempo [s]:	0,14		0,52		0,53	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						
Iteraciones:	8		10		9	
Tiempo [s]:	0,14		0,26		0,12	

Tabla 3.22. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DlgSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9824	-0,9761	0,9828	-0,9110	0,9832	-0,8730
3	0,9690	-1,8722	0,9704	-1,7698	0,9699	-1,7809

4	1,0200	1,5231	1,0200	1,6082	1,0200	1,6290
---	--------	--------	--------	--------	--------	--------

Tabla 3.23. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DigSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0000	0,0000	1,0000	0,0000	1,0000	0,0000
2	0,9824	-0,9761	0,9838	-0,7535	0,9855	0,1169
3	0,9690	-1,8722	0,9701	-1,6940	0,9793	-0,5895
4	1,0200	1,5231	1,0200	1,7409	1,0200	2,7339

Tabla 3.24. Potencias de las Barras.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#		[MW]	[MVar]	[MW]	[MVar]	[MW]	[MVar]
DE	HASTA	MODELO PQ					
1	2	38,692	22,298	-38,465	-31,236	0,227	-8,938
1	3	98,118	61,212	-97,086	-63,569	1,031	-2,356
2	4	-131,535	-74,114	133,251	74,920	1,715	0,806
3	4	-102,914	-60,371	104,749	56,930	1,835	-3,441
DE	HASTA	MODELO POLINOMIAL					
1	2	36,426	21,997	-36,218	-31,032	0,208	-9,036
1	3	92,993	58,349	-92,061	-61,215	0,931	-2,867
2	4	-132,297	-73,002	134,014	73,816	1,718	0,814
3	4	-102,202	-58,543	103,986	54,826	1,784	-3,717
DE	HASTA	MODELO EXPONENCIAL					
1	2	35,042	21,483	-34,847	-30,586	0,195	-9,103
1	3	93,693	59,427	-92,742	-62,192	0,951	-2,764
2	4	-131,378	-72,236	133,068	72,906	1,690	0,671
3	4	-103,115	-58,892	104,932	55,405	1,817	-3,547
DE	HASTA	MODELO LINEAL					
1	2	30,881	20,926	-30,717	-30,190	0,165	-9,263
1	3	89,793	59,627	-88,893	-62,649	0,900	-3,022
2	4	-130,787	-70,665	132,446	71,177	1,659	0,512
3	4	-103,726	-58,543	105,554	55,047	1,827	-3,496
DE	HASTA	MODELO MOTOR DE INDUCCIÓN					
1	2	1,712	23,398	-1,630	-33,088	0,082	-9,690
1	3	36,780	44,164	-36,504	-50,827	0,276	-6,213

2	4	-135,686	-65,323	137,387	66,033	1,701	0,709
3	4	-99,096	-46,355	100,613	41,196	1,517	-5,159

A continuación en la Fig. 3.13 se muestran los perfiles de voltaje obtenidos para los distintos modelos de carga simulados.

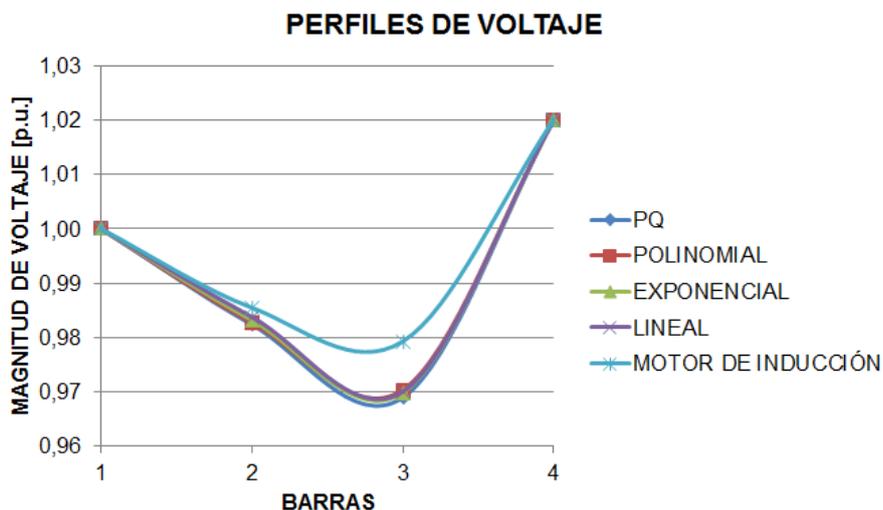


Figura 3.13. Perfiles de Voltaje del Sistema de 4 barras.

3.4.3 Resultados obtenidos en el sistema de 14 barras

En las Tablas 3.25 y 3.26 se muestran los resultados de magnitud y ángulo de voltaje obtenidos mediante el programa desarrollado en Python, seguidamente en las Tablas 3.27 y 3.28 se muestran los resultados obtenidos en DlgSILENT PowerFactory, para posteriormente en las Tablas 3.29 y 3.30 comparar los resultados arrojados por el programa desarrollado en Python y por el software comercial DlgSILENT PowerFactory mediante el cálculo de errores relativos de magnitud y ángulo de voltaje. De la misma manera en las Tablas 3.31 y 3.32 se muestran los resultados obtenidos en PSAT y la comparación de los mismos respectivamente. Finalmente en las Tablas 3.33, 3.34, 3.35, 3.36 y 3.37, se muestran las potencias de envío, recepción y pérdidas de los modelos de carga PQ (potencia constante), polinomial, exponencial, lineal y de motor de inducción respectivamente.

Tabla 3.25. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ

#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0450	-4,9826	1,0450	-5,0535	1,0450	-5,0381
3	1,0100	-12,7251	1,0100	-12,8458	1,0100	-12,8961
4	1,0177	-10,3129	1,0170	-10,4646	1,0167	-10,5256
5	1,0195	-8,7739	1,0190	-8,9088	1,0187	-8,9654
6	1,0700	-14,2209	1,0700	-14,4847	1,0700	-14,6148
7	1,0615	-13,3596	1,0607	-13,5921	1,0601	-13,7200
8	1,0900	-13,3596	1,0900	-13,5921	1,0900	-13,7200
9	1,0559	-14,9385	1,0545	-15,2140	1,0536	-15,3773
10	1,0510	-15,0973	1,0495	-15,3786	1,0489	-15,5335
11	1,0569	-14,7906	1,0559	-15,0697	1,0557	-15,2102
12	1,0551	-15,0756	1,0547	-15,3618	1,0543	-15,5131
13	1,0504	-15,1563	1,0497	-15,4403	1,0495	-15,5844
14	1,0355	-16,0336	1,0337	-16,3474	1,0336	-16,4804
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	9		9		9	
Tiempo [s]:	0,65		0,57		0,57	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	9		33		36	
Tiempo [s]:	0,65		1,90		2,03	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						
Iteraciones:	9		11		11	
Tiempo [s]:	0,65		1,28		1,26	

Tabla 3.26. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0450	-4,9826	1,0450	-5,1348	1,0450	-12,0546
3	1,0100	-12,7251	1,0100	-12,9813	1,0100	-24,9799
4	1,0177	-10,3129	1,0164	-10,6364	0,8882	-24,3775
5	1,0195	-8,7739	1,0183	-9,0697	0,8928	-22,1399
6	1,0700	-14,2209	1,0700	-14,8345	1,0700	-33,7596
7	1,0615	-13,3596	1,0603	-13,8727	0,9529	-32,9791
8	1,0900	-13,3596	1,0900	-13,8727	1,0900	-32,9791
9	1,0559	-14,9385	1,0541	-15,5507	0,8990	-37,5379
10	1,0510	-15,0973	1,0492	-15,7130	0,9195	-37,0623
11	1,0569	-14,7906	1,0556	-15,4198	0,9884	-35,3931

12	1,0551	-15,0756	1,0535	-15,7813	1,0439	-34,9164
13	1,0504	-15,1563	1,0490	-15,8395	1,0261	-35,0516
14	1,0355	-16,0336	1,0332	-16,7280	0,9339	-37,5664
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	9		9		52	
Tiempo [s]:	0,65		0,57		3,50	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	9		36		249	
Tiempo [s]:	0,65		2,07		15,30	
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS						
Iteraciones:	9		11		43	
Tiempo [s]:	0,65		1,26		2,85	

Tabla 3.27. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DgSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0450	-4,9809	1,0450	-5,0538	1,0450	-5,0827
3	1,0100	-12,7180	1,0100	-12,8420	1,0100	-12,8911
4	1,0186	-10,3242	1,0180	-10,4801	1,0176	-10,5396
5	1,0203	-8,7826	1,0197	-8,9211	1,0194	-8,9763
6	1,0700	-14,2226	1,0700	-14,4923	1,0700	-14,6196
7	1,0620	-13,3682	1,0611	-13,6061	1,0605	-13,7319
8	1,0900	-13,3682	1,0900	-13,6061	1,0900	-13,7319
9	1,0563	-14,9466	1,0549	-15,2280	1,0540	-15,3892
10	1,0513	-15,1043	1,0499	-15,3920	1,0492	-15,5442
11	1,0571	-14,7953	1,0561	-15,0805	1,0558	-15,2182
12	1,0552	-15,0774	1,0547	-15,3701	1,0543	-15,5181
13	1,0504	-15,1589	1,0497	-15,4496	1,0495	-15,5902
14	1,0358	-16,0389	1,0340	-16,3592	1,0339	-16,4893

Tabla 3.28. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DgSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ

#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0450	-4,9809	1,0450	-5,1353	1,0450	-12,0481
3	1,0100	-12,7180	1,0100	-12,9778	1,0100	-24,9630
4	1,0186	-10,3242	1,0174	-10,6523	0,8891	-24,3736
5	1,0203	-8,7826	1,0191	-9,0823	0,8936	-22,1347
6	1,0700	-14,2226	1,0700	-14,8412	1,0700	-33,7383
7	1,0620	-13,3682	1,0607	-13,8866	0,9533	-32,9639
8	1,0900	-13,3682	1,0900	-13,8866	1,0900	-32,9639
9	1,0563	-14,9466	1,0545	-15,5645	0,8994	-37,5193
10	1,0513	-15,1043	1,0495	-15,7256	0,9199	-37,0434
11	1,0571	-14,7953	1,0558	-15,4298	0,9886	-35,3735
12	1,0552	-15,0774	1,0536	-15,7883	1,0439	-34,8950
13	1,0504	-15,1589	1,0491	-15,8474	1,0262	-35,0309
14	1,0358	-16,0389	1,0334	-16,7392	0,9342	-37,5464

Tabla 3.29. Errores Relativos de Magnitud y Ángulo de Voltaje obtenidos al comparar los resultados arrojados por el programa con PowerFactory.

PORCENTAJES DE ERROR COMPARACIÓN CON DigSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	e_{relV} [%]	$e_{rel\theta}$ [%]	e_{relV} [%]	$e_{rel\theta}$ [%]	e_{relV} [%]	$e_{rel\theta}$ [%]
#						
1	0,000	0,000	0,000	0,000	0,000	0,000
2	0,000	0,034	0,000	0,006	0,000	0,877
3	0,000	0,056	0,000	0,030	0,000	0,039
4	0,088	0,109	0,098	0,148	0,088	0,133
5	0,078	0,099	0,069	0,138	0,069	0,121
6	0,000	0,012	0,000	0,052	0,000	0,033
7	0,047	0,064	0,038	0,103	0,038	0,087
8	0,000	0,064	0,000	0,103	0,000	0,087
9	0,038	0,054	0,038	0,092	0,038	0,077
10	0,029	0,046	0,038	0,087	0,029	0,069
11	0,019	0,032	0,019	0,072	0,009	0,053
12	0,009	0,012	0,000	0,054	0,000	0,032
13	0,000	0,017	0,000	0,060	0,000	0,037
14	0,029	0,033	0,029	0,072	0,029	0,054

Tabla 3.30. Errores Relativos de Magnitud y Ángulo de Voltaje obtenidos al comparar los resultados arrojados por el programa con PowerFactory.

PORCENTAJES DE ERROR COMPARACIÓN CON DigSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	e_{relV}	$e_{rel\theta}$	e_{relV}	$e_{rel\theta}$	e_{relV}	$e_{rel\theta}$
#	[%]	[%]	[%]	[%]	[%]	[%]
1	0,000	0,000	0,000	0,000	0,000	0,000
2	0,000	0,034	0,000	0,010	0,000	0,054
3	0,000	0,056	0,000	0,027	0,000	0,068
4	0,088	0,109	0,098	0,149	0,101	0,016
5	0,078	0,099	0,079	0,139	0,090	0,023
6	0,000	0,012	0,000	0,045	0,000	0,063
7	0,047	0,064	0,038	0,100	0,042	0,046
8	0,000	0,064	0,000	0,100	0,000	0,046
9	0,038	0,054	0,038	0,089	0,044	0,050
10	0,029	0,046	0,029	0,080	0,043	0,051
11	0,019	0,032	0,019	0,065	0,020	0,055
12	0,009	0,012	0,009	0,044	0,000	0,061
13	0,000	0,017	0,010	0,050	0,010	0,059
14	0,029	0,033	0,019	0,067	0,032	0,053

Tabla 3.31. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PSAT.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PSAT				
BARRA	MODELO PQ		MODELO EXPONENCIAL	
	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000
2	1,0450	-4,9826	1,0450	-5,0831
3	1,0100	-12,7251	1,0100	-12,8961
4	1,0177	-10,3129	1,0167	-10,5256
5	1,0195	-8,7739	1,0187	-8,9654
6	1,0700	-14,2210	1,0700	-14,6148
7	1,0615	-13,3596	1,0601	-13,7200
8	1,0900	-13,3596	1,0900	-13,7200
9	1,0559	-14,9385	1,0536	-15,3773
10	1,0510	-15,0973	1,0489	-15,5335
11	1,0569	-14,7906	1,0557	-15,2102
12	1,0552	-15,0756	1,0543	-15,5131
13	1,0504	-15,1563	1,0495	-15,5844
14	1,0355	-16,0336	1,0336	-16,4805

Tabla 3.32. Errores Relativos de Magnitud y Ángulo de Voltaje obtenidos al comparar los resultados arrojados por el programa con PSAT.

PORCENTAJES DE ERROR COMPARACIÓN CON PSAT				
BARRA	MODELO PQ		MODELO EXPONENCIAL	
	e_{relV}	$e_{rel\theta}$	e_{relV}	$e_{rel\theta}$
#	[%]	[%]	[%]	[%]
1	0,00	0,00	0,00	0,00
2	0,000	0,00	0,00	0,01
3	0,000	0,00	0,00	0,04
4	0,000	0,00	0,00	0,13
5	0,000	0,00	0,00	0,12
6	0,000	0,00	0,00	0,03
7	0,000	0,00	0,00	0,09
8	0,000	0,00	0,00	0,09
9	0,000	0,00	0,00	0,08
10	0,000	0,00	0,00	0,07
11	0,000	0,00	0,00	0,05
12	0,009	0,00	0,00	0,03
13	0,000	0,00	0,00	0,04
14	0,000	0,00	0,00	0,05

Tabla 3.33. Potencias en las Barras del Modelo de Carga de Potencia Constante.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO PQ					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVA _r]	[MW]	[MVA _r]	[MW]	[MVA _r]
1	2	156,883	-20,404	-152,585	27,676	4,298	7,272
1	5	75,510	3,855	-72,748	2,229	2,763	6,084
2	3	73,238	3,560	-70,914	1,602	2,323	5,162
2	4	56,131	-1,550	-54,455	3,021	1,677	1,470
2	5	41,516	1,171	-40,612	-2,099	0,904	-0,928
3	4	-23,286	4,473	23,659	-4,836	0,373	-0,363
4	5	-61,158	15,824	61,673	-14,201	0,514	1,623
4	7	28,074	-9,681	-28,074	11,384	0,000	1,703
4	9	16,080	-0,428	-16,080	1,732	0,000	1,305
5	6	44,087	12,471	-44,087	-8,050	0,000	4,421
6	11	7,353	3,560	-7,298	-3,445	0,055	0,116
6	12	7,786	2,503	-7,714	-2,354	0,072	0,149
6	13	17,748	7,217	-17,536	-6,799	0,212	0,418
7	8	0,000	-17,163	0,000	17,623	0,000	0,460

7	9	28,074	5,779	-28,074	-4,977	0,000	0,802
9	10	5,228	4,219	-5,215	-4,185	0,013	0,034
9	14	9,426	3,610	-9,310	-3,363	0,116	0,247
10	11	-3,785	-1,615	3,798	1,645	0,013	0,029
12	13	1,614	0,754	-1,608	-0,748	0,006	0,006
13	14	5,644	1,747	-5,590	-1,637	0,054	0,110

Tabla 3.34. Potencias en las Barras del Modelo de Carga Polinomial.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO POLINOMIAL					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVA _r]	[MW]	[MVA _r]	[MW]	[MVA _r]
1	2	159,028	-20,904	-154,611	28,542	4,418	7,638
1	5	76,637	4,021	-73,790	2,411	2,846	6,432
2	3	73,686	3,516	-71,334	1,765	2,351	5,281
2	4	56,999	-1,376	-55,270	3,007	1,729	1,631
2	5	42,225	1,320	-41,290	-2,150	0,935	-0,830
3	4	-22,866	4,676	23,228	-5,067	0,362	-0,391
4	5	-61,794	15,839	62,319	-14,183	0,525	1,657
4	7	28,777	-9,544	-28,777	11,321	0,000	1,777
4	9	16,476	-0,253	-16,476	1,624	0,000	1,371
5	6	45,102	12,303	-45,102	-7,695	0,000	4,608
6	11	7,675	3,927	-7,614	-3,798	0,062	0,129
6	12	8,012	2,620	-7,936	-2,461	0,076	0,159
6	13	18,214	7,551	-17,989	-7,109	0,225	0,442
7	8	0,000	-17,658	0,000	18,146	0,000	0,488
7	9	28,777	6,336	-28,777	-5,487	0,000	0,849
9	10	5,356	4,227	-5,343	-4,192	0,013	0,035
9	14	9,683	3,633	-9,561	-3,373	0,122	0,260
10	11	-3,895	-1,834	3,908	1,866	0,014	0,032
12	13	1,645	0,802	-1,638	-0,796	0,007	0,006
13	14	5,896	1,955	-5,836	-1,833	0,060	0,122

Tabla 3.35. Potencias en las Barras del Modelo de Carga Exponencial.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO EXPONENCIAL					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVA _r]	[MW]	[MVA _r]	[MW]	[MVA _r]
1	2	159,923	-21,112	-155,455	28,904	4,468	7,792
1	5	77,111	4,119	-74,229	2,462	2,882	6,581
2	3	73,873	3,498	-71,509	1,832	2,363	5,330

2	4	57,350	-1,249	-55,600	2,947	1,750	1,698
2	5	42,533	1,416	-41,583	-2,203	0,949	-0,787
3	4	-22,691	4,814	23,048	-5,215	0,358	-0,401
4	5	-61,970	15,752	62,498	-14,087	0,528	1,666
4	7	29,366	-9,417	-29,366	11,258	0,000	1,840
4	9	16,810	-0,119	-16,810	1,546	0,000	1,428
5	6	45,681	12,220	-45,681	-7,503	0,000	4,717
6	11	7,814	4,005	-7,750	-3,871	0,064	0,134
6	12	8,209	2,695	-8,128	-2,528	0,080	0,167
6	13	18,459	7,627	-18,229	-7,173	0,230	0,454
7	8	0,000	-17,977	0,000	18,484	0,000	0,507
7	9	29,366	6,720	-29,366	-5,832	0,000	0,888
9	10	5,054	3,958	-5,042	-3,926	0,012	0,031
9	14	9,378	3,454	-9,264	-3,211	0,114	0,243
10	11	-4,084	-1,946	4,099	1,982	0,015	0,036
12	13	1,571	0,798	-1,565	-0,793	0,006	0,006
13	14	5,833	1,951	-5,774	-1,832	0,059	0,120

Tabla 3.36. Potencias en las Barras del Modelo de Carga Exponencial.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO LINEAL					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVar]	[MW]	[MVar]	[MW]	[MVar]
1	2	161,485	-21,473	-156,928	29,537	4,557	8,064
1	5	77,977	4,197	-75,030	2,654	2,947	6,851
2	3	74,175	3,469	-71,793	1,942	2,382	5,411
2	4	57,957	-1,242	-56,169	3,054	1,788	1,811
2	5	43,096	1,462	-42,122	-2,170	0,974	-0,708
3	4	-22,407	4,839	22,757	-5,261	0,350	-0,422
4	5	-62,134	15,981	62,666	-14,303	0,532	1,678
4	7	29,748	-9,589	-29,748	11,481	0,000	1,891
4	9	17,030	-0,237	-17,030	1,703	0,000	1,466
5	6	46,597	12,161	-46,597	-7,265	0,000	4,896
6	11	7,746	4,074	-7,683	-3,941	0,064	0,133
6	12	8,627	2,800	-8,538	-2,616	0,088	0,184
6	13	19,024	7,702	-18,780	-7,222	0,243	0,479
7	8	0,000	-17,896	0,000	18,398	0,000	0,502
7	9	29,748	6,415	-29,748	-5,509	0,000	0,906
9	10	5,269	4,150	-5,256	-4,116	0,013	0,034
9	14	9,946	3,554	-9,819	-3,282	0,128	0,271
10	11	-3,766	-1,910	3,779	1,941	0,013	0,031
12	13	1,438	0,792	-1,432	-0,788	0,005	0,005

13	14	5,796	1,969	-5,738	-1,850	0,058	0,119
----	----	-------	-------	--------	--------	-------	-------

Tabla 3.37. Potencias en las Barras del Modelo de Carga de Motor de Inducción.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO MOTOR DE INDUCCIÓN					
DE #	HASTA #	Penvío [MW]	Qenvío [MVar]	Precepción [MW]	Qrecepción [MVar]	Pérdidas P [MW]	Pérdidas Q [MVar]
1	2	373,250	-57,065	-348,716	126,122	24,534	69,057
1	5	176,387	65,237	-159,203	0,977	17,185	66,214
2	3	120,079	1,090	-113,869	20,446	6,210	21,536
2	4	132,575	59,535	-121,216	-28,268	11,358	31,266
2	5	114,363	60,392	-105,519	-36,659	8,844	23,733
3	4	19,669	63,623	-16,701	-57,205	2,968	6,418
4	5	-69,240	13,512	70,083	-10,855	0,842	2,657
4	7	61,889	-14,769	-61,889	25,034	0,000	10,265
4	9	33,730	6,792	-33,730	1,045	0,000	7,837
5	6	81,921	-34,245	-81,921	55,895	0,000	21,650
6	11	29,502	30,039	-28,031	-26,960	1,471	3,080
6	12	11,463	5,512	-11,290	-5,151	0,174	0,362
6	13	29,756	21,173	-28,895	-19,656	0,771	1,518
7	8	0,000	-74,183	0,000	84,859	0,000	10,676
7	9	61,889	49,149	-61,889	-41,581	0,000	7,568
9	10	-14,307	-16,435	14,494	16,932	0,187	0,496
9	14	-4,345	-9,569	4,518	9,939	0,174	0,370
10	11	-23,494	-22,732	24,531	25,160	1,037	2,428
12	13	5,190	3,551	-5,109	-3,478	0,080	0,073
13	14	20,595	17,334	-19,418	-14,939	1,176	2,395

En las Fig. 3.14 y Fig. 3.15 se muestran los perfiles de voltaje de los distintos modelos de carga simulados y los máximos errores porcentuales de la magnitud de voltaje encontrados al comparar los resultados de la herramienta computacional con los resultados de DIgSILENT PowerFactory.

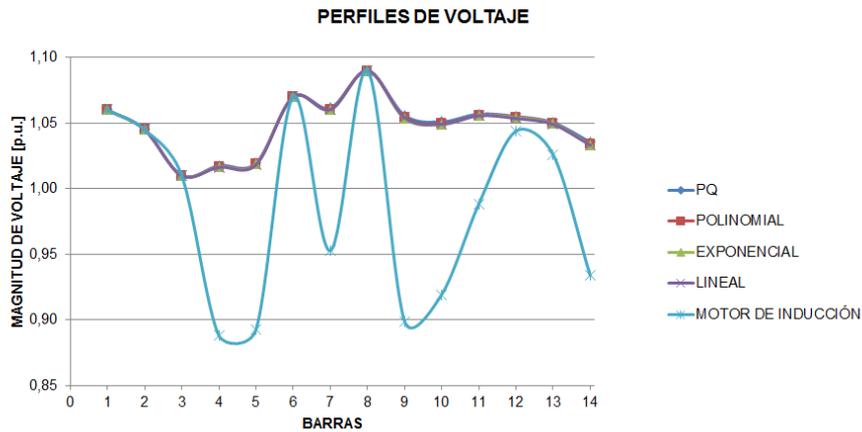


Figura 3.14. Perfiles de Voltaje del Sistema de 14 barras.

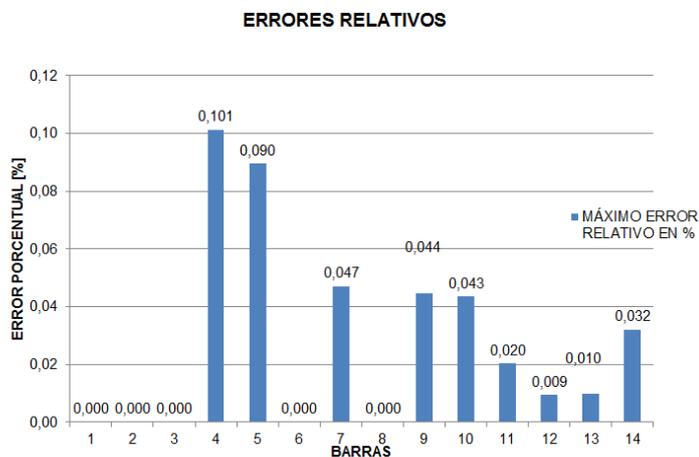


Figura 3.15. Máximos Errores relativos en Magnitud de Voltaje del Sistema de 14 barras.

3.4.4 Resultados obtenidos en el sistema de 30 barras

En las Tablas 3.38 y 3.39 se muestran los resultados de magnitud y ángulo de voltaje obtenidos mediante el programa desarrollado en Python, seguidamente en las Tablas 3.40 y 3.41 se muestran los resultados obtenidos en DigSILENT PowerFactory, para posteriormente en las Tablas 3.42 y 3.43 comparar los resultados arrojados por el programa desarrollado en Python y por el software comercial DigSILENT PowerFactory mediante el cálculo de errores relativos de magnitud y ángulo de voltaje. De la misma manera las Tablas 3.44 y 3.45 se muestran los resultados obtenidos en PSAT y la comparación de los mismos respectivamente. Finalmente en las Tablas 3.46, 3.47, 3.48, 3.49, 3.50, se muestran las potencias de envío, recepción y pérdidas de los modelos de carga PQ (potencia constante), polinomial, exponencial, lineal y de motor de inducción respectivamente.

Tabla 3.38. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0430	-5,3500	1,0430	-5,4036	1,0430	-5,2635
3	1,0207	-7,5323	1,0203	-7,6166	1,0210	-7,3998
4	1,0117	-9,2845	1,0112	-9,3896	1,0120	-9,1181
5	1,0100	-14,1671	1,0100	-14,2565	1,0100	-14,0219
6	1,0102	-11,0654	1,0099	-11,1833	1,0105	-10,8613
7	1,0024	-12,8661	1,0021	-12,9731	1,0025	-12,6870
8	1,0100	-11,8145	1,0100	-11,9397	1,0100	-11,6039
9	1,0509	-14,1098	1,0501	-14,3053	1,0512	-13,7101
10	1,0451	-15,7006	1,0437	-15,9383	1,0454	-15,1988
11	1,0820	-14,1098	1,0820	-14,3053	1,0820	-13,7101
12	1,0571	-14,9441	1,0562	-15,1784	1,0566	-14,6811
13	1,0710	-14,9441	1,0710	-15,1784	1,0710	-14,6811
14	1,0423	-15,8363	1,0410	-16,0855	1,0413	-15,5739
15	1,0377	-15,9283	1,0362	-16,1783	1,0368	-15,6416
16	1,0444	-15,5272	1,0430	-15,7687	1,0442	-15,1702
17	1,0399	-15,8623	1,0383	-16,1070	1,0399	-15,4075
18	1,0281	-16,5426	1,0265	-16,7995	1,0274	-16,1944
19	1,0256	-16,7164	1,0239	-16,9749	1,0250	-16,3292
20	1,0297	-16,5198	1,0280	-16,7741	1,0293	-16,1060
21	1,0327	-16,1433	1,0309	-16,3900	1,0326	-15,6695
22	1,0332	-16,1291	1,0315	-16,3735	1,0331	-15,6608
23	1,0272	-16,3190	1,0256	-16,5645	1,0262	-15,9947
24	1,0216	-16,4956	1,0199	-16,7313	1,0207	-16,1128
25	1,0173	-16,0677	1,0163	-16,2651	1,0168	-15,7446
26	0,9996	-16,4874	0,9986	-16,6854	0,9991	-16,1643
27	1,0232	-15,5434	1,0226	-15,7160	1,0229	-15,2570
28	1,0068	-11,6893	1,0065	-11,8138	1,0070	-11,4739
29	1,0034	-16,7734	1,0028	-16,9454	1,0032	-16,4785
30	0,9919	-17,6562	0,9914	-17,8255	0,9919	-17,3502
METODOLOGÍA DE INYECCIÓN NODAL						
Iteraciones:	6		7		7	
Tiempo [s]:	0,90		0,97		0,86	
METODOLOGÍA DE FLUJOS SUCESIVOS						
Iteraciones:	6		34		36	

Tiempo [s]:	0,90	4,04	4,23
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS			
Iteraciones:	6	9	9
Tiempo [s]:	0,90	2,14	2,14

Tabla 3.39. Resultados de Magnitud y Ángulo de Voltaje obtenidos en Python.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PYTHON						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0430	-5,3500	1,0430	-5,3241	1,0430	-13,1660
3	1,0207	-7,5323	1,0205	-7,4957	0,8843	-21,0235
4	1,0117	-9,2845	1,0115	-9,2364	0,9048	-24,2787
5	1,0100	-14,1671	1,0100	-14,1241	1,0100	-29,1240
6	1,0102	-11,0654	1,0101	-10,9955	0,9568	-26,5066
7	1,0024	-12,8661	1,0023	-12,8097	0,9122	-30,2903
8	1,0100	-11,8145	1,0100	-11,7460	1,0100	-28,2294
9	1,0509	-14,1098	1,0503	-13,9336	1,0180	-29,5969
10	1,0451	-15,7006	1,0439	-15,4706	1,0064	-31,1850
11	1,0820	-14,1098	1,0820	-13,9336	1,0820	-29,5969
12	1,0571	-14,9441	1,0557	-14,9304	1,0159	-30,6001
13	1,0710	-14,9441	1,0710	-14,9304	1,0710	-30,6001
14	1,0423	-15,8363	1,0398	-15,8618	1,0006	-31,5422
15	1,0377	-15,9283	1,0353	-15,9280	0,9962	-31,6214
16	1,0444	-15,5272	1,0430	-15,4317	1,0041	-31,1372
17	1,0399	-15,8623	1,0384	-15,6737	1,0004	-31,4013
18	1,0281	-16,5426	1,0256	-16,4925	0,9871	-32,2205
19	1,0256	-16,7164	1,0231	-16,6357	0,9850	-32,3686
20	1,0297	-16,5198	1,0274	-16,4063	0,9896	-32,1345
21	1,0327	-16,1433	1,0305	-15,9660	0,9931	-31,6865
22	1,0332	-16,1291	1,0311	-15,9526	0,9936	-31,6785
23	1,0272	-16,3190	1,0246	-16,2821	0,9854	-32,0151
24	1,0216	-16,4956	1,0190	-16,3935	0,9797	-32,1657
25	1,0173	-16,0677	1,0157	-15,9778	0,9739	-31,8605
26	0,9996	-16,4874	0,9981	-16,3960	0,9554	-32,3191
27	1,0232	-15,5434	1,0222	-15,4610	0,9793	-31,3898
28	1,0068	-11,6893	1,0066	-11,6163	0,9654	-27,3669
29	1,0034	-16,7734	1,0024	-16,6843	0,9585	-32,7350
30	0,9919	-17,6562	0,9911	-17,5545	0,9464	-33,7036
METODOLOGÍA DE INYECCIÓN NODAL						

Iteraciones:	6	9	29
Tiempo [s]:	0,90	1,15	3,78
METODOLOGÍA DE FLUJOS SUCESIVOS			
Iteraciones:	6	50	154
Tiempo [s]:	0,90	5,96	18,98
METODOLOGÍA DE REESTRUCTURACIÓN DE POTENCIAS CALCULADAS			
Iteraciones:	6	10	21
Tiempo [s]:	0,90	2,47	2,74

Tabla 3.40. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DigSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	V	θ	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0430	-5,3459	1,0430	-5,4026	1,0430	-5,4075
3	1,0207	-7,5233	1,0203	-7,6125	1,0203	-7,6176
4	1,0117	-9,2734	1,0112	-9,3845	1,0112	-9,3901
5	1,0100	-14,1609	1,0100	-14,2558	1,0100	-14,2749
6	1,0103	-11,0596	1,0100	-11,1845	1,0100	-11,1907
7	1,0024	-12,8601	1,0022	-12,9737	1,0022	-12,9853
8	1,0100	-11,8069	1,0100	-11,9395	1,0100	-11,9449
9	1,0516	-14,1077	1,0507	-14,3151	1,0509	-14,3167
10	1,0464	-15,6986	1,0449	-15,9507	1,0452	-15,9495
11	1,0820	-14,1077	1,0820	-14,3150	1,0820	-14,3166
12	1,0562	-14,8977	1,0552	-15,1444	1,0553	-15,1505
13	1,0710	-14,8977	1,0710	-15,1444	1,0710	-15,1505
14	1,0449	-15,8466	1,0436	-16,1129	1,0436	-16,1295
15	1,0434	-16,0720	1,0419	-16,3435	1,0420	-16,3590
16	1,0444	-15,5044	1,0430	-15,7592	1,0432	-15,7625
17	1,0408	-15,8520	1,0392	-16,1108	1,0396	-16,1081
18	1,0323	-16,6295	1,0306	-16,9066	1,0307	-16,9200
19	1,0289	-16,7713	1,0271	-17,0488	1,0272	-17,0614
20	1,0325	-16,5595	1,0308	-16,8319	1,0309	-16,8423
21	1,0343	-16,1463	1,0324	-16,4079	1,0327	-16,4064
22	1,0349	-16,1342	1,0330	-16,3937	1,0333	-16,3934
23	1,0315	-16,4156	1,0299	-16,6805	1,0299	-16,6953
24	1,0240	-16,5327	1,0223	-16,7847	1,0223	-16,7938
25	1,0190	-16,0775	1,0179	-16,2882	1,0180	-16,2911
26	1,0014	-16,4958	1,0003	-16,7074	1,0003	-16,7104

27	1,0244	-15,5376	1,0238	-15,7217	1,0239	-15,7205
28	1,0070	-11,6853	1,0066	-11,8174	1,0067	-11,8228
29	1,0046	-16,7647	1,0040	-16,9481	1,0042	-16,9411
30	0,9932	-17,6454	0,9926	-17,8254	0,9928	-17,8123

Tabla 3.41. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PowerFactory.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN DgSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	V	θ	V	θ	V	θ
#	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]	[p.u.]	[$^{\circ}$]
1	1,0600	0,0000	1,0600	0,0000	1,0600	0,0000
2	1,0430	-5,3459	1,0430	-5,4704	1,0430	-13,3374
3	1,0207	-7,5233	1,0198	-7,7196	0,8830	-21,2853
4	1,0117	-9,2734	1,0107	-9,5161	0,9035	-24,6082
5	1,0100	-14,1609	1,0100	-14,3727	1,0100	-29,4110
6	1,0103	-11,0596	1,0096	-11,3337	0,9560	-26,8990
7	1,0024	-12,8601	1,0019	-13,1112	0,9117	-30,6411
8	1,0100	-11,8069	1,0100	-12,0967	1,0100	-28,6394
9	1,0516	-14,1077	1,0500	-14,5611	1,0176	-30,2826
10	1,0464	-15,6986	1,0437	-16,2487	1,0061	-32,0204
11	1,0820	-14,1077	1,0820	-14,5611	1,0820	-30,2826
12	1,0562	-14,8977	1,0544	-15,4217	1,0145	-31,1454
13	1,0710	-14,8977	1,0710	-15,4216	1,0710	-31,1454
14	1,0449	-15,8466	1,0421	-16,4449	1,0028	-32,1726
15	1,0434	-16,0720	1,0405	-16,6787	1,0013	-32,4177
16	1,0444	-15,5044	1,0419	-16,0482	1,0030	-31,8096
17	1,0408	-15,8520	1,0380	-16,4006	0,9999	-32,1851
18	1,0323	-16,6295	1,0290	-17,2506	0,9904	-33,0256
19	1,0289	-16,7713	1,0253	-17,4000	0,9872	-33,1811
20	1,0325	-16,5595	1,0290	-17,1737	0,9911	-32,9521
21	1,0343	-16,1463	1,0306	-16,7305	0,9931	-32,5070
22	1,0349	-16,1342	1,0313	-16,7127	0,9937	-32,4942
23	1,0315	-16,4156	1,0282	-17,0142	0,9890	-32,7940
24	1,0240	-16,5327	1,0205	-17,1023	0,9812	-32,9254
25	1,0190	-16,0775	1,0169	-16,5463	0,9751	-32,4782
26	1,0014	-16,4958	0,9992	-16,9653	0,9566	-32,9358
27	1,0244	-15,5376	1,0231	-15,9425	0,9802	-31,9203
28	1,0070	-11,6853	1,0063	-11,9752	0,9649	-27,7809
29	1,0046	-16,7647	1,0034	-17,1658	0,9594	-33,2631
30	0,9932	-17,6454	0,9921	-18,0352	0,9473	-34,2299

Tabla 3.42. Errores Relativos de Magnitud y Ángulo de Voltaje obtenidos al comparar los resultados arrojados por el programa con PowerFactory.

PORCENTAJES DE ERROR COMPARACIÓN CON DigSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO POLINOMIAL		MODELO EXPONENCIAL	
	e_{relV} [%]	$e_{rel\theta}$ [%]	e_{relV} [%]	$e_{rel\theta}$ [%]	e_{relV} [%]	$e_{rel\theta}$ [%]
1	0,000	0,000	0,000	0,000	0,000	0,000
2	0,000	0,077	0,000	0,018	0,000	2,663
3	0,001	0,119	0,003	0,054	0,071	2,859
4	0,001	0,120	0,002	0,054	0,075	2,897
5	0,000	0,044	0,000	0,005	0,000	1,772
6	0,011	0,053	0,006	0,011	0,050	2,943
7	0,007	0,047	0,009	0,005	0,030	2,297
8	0,000	0,065	0,000	0,002	0,000	2,855
9	0,064	0,015	0,057	0,068	0,033	4,237
10	0,126	0,013	0,117	0,077	0,018	4,707
11	0,000	0,015	0,000	0,068	0,000	4,237
12	0,084	0,312	0,092	0,224	0,119	3,098
13	0,000	0,312	0,000	0,224	0,000	3,098
14	0,255	0,065	0,253	0,170	0,221	3,445
15	0,545	0,894	0,550	1,011	0,496	4,385
16	0,003	0,147	0,001	0,060	0,093	3,758
17	0,089	0,065	0,087	0,024	0,033	4,349
18	0,408	0,523	0,402	0,634	0,323	4,289
19	0,322	0,327	0,315	0,434	0,217	4,292
20	0,273	0,240	0,271	0,343	0,157	4,372
21	0,150	0,019	0,142	0,109	0,013	4,492
22	0,156	0,032	0,148	0,123	0,024	4,469
23	0,420	0,589	0,417	0,696	0,355	4,197
24	0,242	0,225	0,235	0,318	0,155	4,055
25	0,168	0,061	0,161	0,142	0,117	3,354
26	0,174	0,051	0,167	0,132	0,122	3,268
27	0,118	0,037	0,113	0,036	0,094	2,949
28	0,020	0,034	0,013	0,030	0,033	2,951
29	0,123	0,052	0,116	0,016	0,095	2,731
30	0,126	0,062	0,119	0,001	0,095	2,594

Tabla 3.43. Errores Relativos de Magnitud y Ángulo de Voltaje obtenidos al comparar los resultados arrojados por el programa con PowerFactory.

PORCENTAJES DE ERROR COMPARACIÓN CON DigSILENT POWERFACTORY						
BARRA	MODELO PQ		MODELO LINEAL		MODELO MOTOR DE INDUCCIÓN	
	e_{relV} [%]	$e_{rel\theta}$ [%]	e_{relV} [%]	$e_{rel\theta}$ [%]	e_{relV} [%]	$e_{rel\theta}$ [%]
1	0,000	0,000	0,000	0,000	0,000	0,000
2	0,000	0,077	0,000	2,674	0,000	1,285
3	0,001	0,119	0,071	2,901	0,147	1,230
4	0,001	0,120	0,079	2,940	0,144	1,339
5	0,000	0,044	0,000	1,730	0,000	0,976
6	0,011	0,053	0,051	2,984	0,084	1,459
7	0,007	0,047	0,037	2,300	0,055	1,145
8	0,000	0,065	0,000	2,899	0,000	1,432
9	0,064	0,015	0,032	4,309	0,039	2,264
10	0,126	0,013	0,021	4,788	0,030	2,600
11	0,000	0,015	0,000	4,309	0,000	2,264
12	0,084	0,312	0,123	3,186	0,138	1,751
13	0,000	0,312	0,000	3,185	0,000	1,751
14	0,255	0,065	0,224	3,545	0,219	1,959
15	0,545	0,894	0,504	4,501	0,509	2,456
16	0,003	0,147	0,102	3,842	0,110	2,114
17	0,089	0,065	0,035	4,432	0,050	2,435
18	0,408	0,523	0,326	4,395	0,333	2,438
19	0,322	0,327	0,210	4,393	0,223	2,449
20	0,273	0,240	0,157	4,468	0,151	2,481
21	0,150	0,019	0,011	4,569	0,000	2,524
22	0,156	0,032	0,022	4,548	0,010	2,510
23	0,420	0,589	0,354	4,303	0,364	2,375
24	0,242	0,225	0,145	4,145	0,153	2,307
25	0,168	0,061	0,115	3,436	0,123	1,902
26	0,174	0,051	0,113	3,355	0,125	1,872
27	0,118	0,037	0,092	3,020	0,092	1,662
28	0,020	0,034	0,033	2,997	0,052	1,490
29	0,123	0,052	0,097	2,805	0,094	1,588
30	0,126	0,062	0,098	2,665	0,095	1,538

Tabla 3.44. Resultados de Magnitud y Ángulo de Voltaje obtenidos en PSAT.

RESULTADOS DE LA SIMULACIÓN OBTENIDA EN PSAT				
BARRA	MODELO PQ		MODELO EXPONENCIAL	
	V	θ	V	θ
#	[p.u.]	[°]	[p.u.]	[°]
1	1,0600	0,0000	1,0600	0,0000
2	1,0430	-5,3500	1,0430	-5,4070
3	1,0207	-7,5323	1,0203	-7,6196
4	1,0117	-9,2845	1,0113	-9,3925
5	1,0100	-14,1671	1,0100	-14,2735
6	1,0102	-11,0654	1,0099	-11,1863
7	1,0024	-12,8661	1,0022	-12,9821
8	1,0100	-11,8146	1,0100	-11,9417
9	1,0509	-14,1098	1,0502	-14,3021
10	1,0451	-15,7006	1,0440	-15,9314
11	1,0820	-14,1098	1,0820	-14,3021
12	1,0571	-14,9441	1,0563	-15,1788
13	1,0710	-14,9441	1,0710	-15,1788
14	1,0423	-15,8363	1,0410	-16,0956
15	1,0377	-15,9283	1,0363	-16,1859
16	1,0444	-15,5272	1,0433	-15,7662
17	1,0399	-15,8623	1,0387	-16,0986
18	1,0281	-16,5426	1,0266	-16,8054
19	1,0256	-16,7164	1,0240	-16,9801
20	1,0297	-16,5198	1,0282	-16,7775
21	1,0327	-16,1433	1,0313	-16,3823
22	1,0332	-16,1291	1,0318	-16,3670
23	1,0272	-16,3190	1,0256	-16,5711
24	1,0216	-16,4956	1,0199	-16,7326
25	1,0173	-16,0677	1,0164	-16,2596
26	0,9996	-16,4874	0,9987	-16,6793
27	1,0232	-15,5434	1,0227	-15,7066
28	1,0068	-11,6893	1,0065	-11,8155
29	1,0034	-16,7734	1,0030	-16,9283
30	0,9919	-17,6562	0,9917	-17,8000

Tabla 3.45. Errores Relativos de Magnitud y Ángulo de Voltaje obtenidos al comparar los resultados arrojados por el programa con PSAT.

PORCENTAJES DE ERROR COMPARACIÓN CON PSAT				
BARRA	MODELO PQ		MODELO EXPONENCIAL	
	e_{relV}	$e_{rel\theta}$	e_{relV}	$e_{rel\theta}$
#	[%]	[%]	[%]	[%]
1	0,000	0,000	0,000	0,000
2	0,000	0,000	0,000	2,654
3	0,001	0,000	0,069	2,885
4	0,003	0,000	0,069	2,921
5	0,000	0,000	0,000	1,763
6	0,003	0,000	0,059	2,905
7	0,004	0,000	0,030	2,273
8	0,000	0,000	0,000	2,829
9	0,000	0,000	0,095	4,139
10	0,001	0,000	0,134	4,598
11	0,000	0,000	0,000	4,139
12	0,000	0,000	0,028	3,279
13	0,000	0,000	0,000	3,279
14	0,003	0,000	0,029	3,241
15	0,003	0,000	0,048	3,363
16	0,003	0,000	0,086	3,780
17	0,001	0,000	0,116	4,293
18	0,004	0,000	0,078	3,636
19	0,003	0,000	0,098	3,833
20	0,002	0,000	0,107	4,002
21	0,001	0,000	0,126	4,351
22	0,004	0,000	0,126	4,315
23	0,003	0,000	0,059	3,478
24	0,003	0,000	0,078	3,704
25	0,002	0,000	0,039	3,167
26	0,000	0,000	0,040	3,088
27	0,003	0,000	0,020	2,862
28	0,000	0,000	0,050	2,891
29	0,001	0,000	0,020	2,657
30	0,001	0,000	0,020	2,527

Tabla 3.46. Potencias en las Barras del Modelo de Carga de Potencia Constante.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO PQ					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVA _r]	[MW]	[MVA _r]	[MW]	[MVA _r]
1	2	173,223	-21,093	-168,040	30,778	5,184	9,686
1	3	87,729	4,566	-84,614	2,401	3,115	6,967
2	4	43,620	3,895	-42,604	-4,687	1,015	-0,791
3	4	82,214	-3,601	-81,356	5,196	0,858	1,595
2	5	82,386	1,751	-79,434	6,245	2,952	7,996
2	6	60,334	0,440	-58,387	1,527	1,947	1,967
4	6	72,154	-16,354	-71,520	17,642	0,635	1,288
5	7	-14,766	11,691	14,938	-13,324	0,171	-1,633
6	7	38,119	-2,975	-37,738	2,424	0,381	-0,550
6	8	29,568	-8,145	-29,458	7,610	0,110	-0,534
6	9	27,718	-8,173	-27,718	9,801	0,000	1,628
6	10	15,836	0,161	-15,836	1,122	0,000	1,283
9	11	0,000	-15,715	0,000	16,180	0,000	0,465
9	10	27,718	5,914	-27,718	-5,114	0,000	0,800
4	12	44,206	14,244	-44,206	-9,558	0,000	4,686
12	13	0,000	-10,493	0,000	10,631	0,000	0,138
12	14	7,859	2,402	-7,785	-2,247	0,074	0,155
12	15	17,898	6,796	-17,681	-6,368	0,217	0,428
12	16	7,249	3,354	-7,195	-3,240	0,054	0,113
14	15	1,585	0,647	-1,579	-0,641	0,006	0,005
16	17	3,695	1,440	-3,688	-1,413	0,008	0,028
15	18	6,020	1,598	-5,981	-1,519	0,039	0,079
18	19	2,781	0,619	-2,776	-0,609	0,005	0,010
19	20	-6,724	-2,791	6,741	2,825	0,017	0,034
10	20	9,022	3,707	-8,941	-3,525	0,082	0,182
10	17	5,327	4,424	-5,312	-4,387	0,014	0,037
10	21	15,786	10,012	-15,675	-9,772	0,111	0,240
10	22	7,619	4,601	-7,566	-4,492	0,053	0,109
21	22	-1,825	-1,428	1,826	1,429	0,001	0,001
15	23	5,040	2,912	-5,008	-2,848	0,031	0,064
22	24	5,740	3,063	-5,695	-2,992	0,046	0,071
23	24	1,808	1,248	-1,802	-1,236	0,006	0,012
24	25	-1,203	2,016	1,213	-1,998	0,010	0,017
25	26	3,545	2,367	-3,500	-2,300	0,045	0,067
25	27	-4,758	-0,368	4,782	0,414	0,024	0,046
28	27	18,064	5,034	-18,064	-3,746	0,000	1,287
27	29	6,190	1,669	-6,104	-1,506	0,086	0,163
27	30	7,092	1,663	-6,930	-1,358	0,162	0,305

29	30	3,704	0,606	-3,670	-0,542	0,034	0,063
8	28	-0,542	-0,392	0,544	-3,954	0,002	-4,345
6	28	18,666	-0,037	-18,608	-1,080	0,058	-1,118

Tabla 3.47. Potencias en las Barras del Modelo de Carga Polinomial.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO POLINOMIAL					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVAr]	[MW]	[MVAr]	[MW]	[MVAr]
1	2	174,884	-21,479	-169,599	31,467	5,285	9,989
1	3	88,685	4,705	-85,502	2,515	3,184	7,220
2	4	44,191	4,037	-43,149	-4,744	1,042	-0,707
3	4	83,081	-3,730	-82,205	5,380	0,877	1,650
2	5	82,708	1,726	-79,733	6,367	2,975	8,093
2	6	61,000	0,498	-59,009	1,601	1,991	2,099
4	6	72,522	-16,764	-71,879	18,082	0,643	1,318
5	7	-14,467	11,763	14,635	-13,403	0,168	-1,640
6	7	37,835	-3,054	-37,459	2,489	0,376	-0,566
6	8	29,615	-9,012	-29,504	8,485	0,112	-0,527
6	9	28,391	-7,906	-28,391	9,600	0,000	1,694
6	10	16,217	0,394	-16,217	0,953	0,000	1,347
9	11	0,000	-16,118	0,000	16,608	0,000	0,490
9	10	28,391	6,518	-28,391	-5,671	0,000	0,846
4	12	45,150	14,505	-45,150	-9,615	0,000	4,890
12	13	0,000	-11,179	0,000	11,335	0,000	0,157
12	14	7,993	2,475	-7,916	-2,314	0,077	0,161
12	15	18,217	7,030	-17,990	-6,584	0,226	0,446
12	16	7,379	3,504	-7,322	-3,385	0,057	0,119
14	15	1,627	0,680	-1,621	-0,674	0,006	0,006
16	17	3,745	1,538	-3,737	-1,509	0,008	0,029
15	18	6,094	1,662	-6,054	-1,581	0,040	0,081
18	19	2,819	0,667	-2,814	-0,656	0,005	0,010
19	20	-6,829	-2,813	6,847	2,848	0,018	0,035
10	20	9,170	3,753	-9,086	-3,565	0,084	0,188
10	17	5,494	4,477	-5,479	-4,438	0,015	0,039
10	21	16,126	10,334	-16,009	-10,081	0,117	0,252
10	22	7,753	4,736	-7,698	-4,623	0,055	0,114
21	22	-2,057	-1,555	2,058	1,557	0,001	0,001
15	23	5,023	2,985	-4,992	-2,921	0,032	0,064
22	24	5,640	3,066	-5,595	-2,997	0,045	0,069
23	24	1,764	1,301	-1,758	-1,288	0,006	0,012
24	25	-1,443	1,954	1,454	-1,935	0,011	0,019

25	26	3,542	2,365	-3,498	-2,298	0,045	0,067
25	27	-4,997	-0,429	5,023	0,480	0,027	0,051
28	27	18,272	5,117	-18,272	-3,798	0,000	1,319
27	29	6,178	1,664	-6,092	-1,502	0,086	0,163
27	30	7,070	1,654	-6,909	-1,350	0,161	0,304
29	30	3,686	0,599	-3,652	-0,536	0,033	0,063
8	28	-0,496	-0,233	0,499	-4,110	0,003	-4,343
6	28	18,830	-0,106	-18,771	-1,007	0,059	-1,113

Tabla 3.48. Potencias en las Barras del Modelo de Carga Exponencial.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO EXPONENCIAL					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVAr]	[MW]	[MVAr]	[MW]	[MVAr]
1	2	170,543	-20,466	-165,521	29,669	5,022	9,203
1	3	86,260	4,614	-83,248	1,977	3,012	6,592
2	4	42,801	3,943	-41,823	-4,848	0,978	-0,905
3	4	80,781	-3,213	-79,954	4,721	0,827	1,508
2	5	81,858	1,794	-78,943	6,045	2,914	7,839
2	6	59,162	0,537	-57,289	1,202	1,873	1,739
4	6	70,670	-16,009	-70,062	17,205	0,608	1,196
5	7	-15,257	11,765	15,436	-13,380	0,179	-1,614
6	7	38,675	-2,972	-38,282	2,454	0,392	-0,518
6	8	29,498	-7,442	-29,390	6,902	0,108	-0,541
6	9	25,952	-8,253	-25,952	9,698	0,000	1,445
6	10	14,829	0,091	-14,829	1,033	0,000	1,124
9	11	0,000	-15,585	0,000	16,043	0,000	0,457
9	10	25,952	5,888	-25,952	-5,183	0,000	0,705
4	12	43,444	14,522	-43,444	-9,966	0,000	4,556
12	13	0,000	-10,888	0,000	11,036	0,000	0,149
12	14	7,926	2,552	-7,849	-2,393	0,076	0,159
12	15	17,711	7,149	-17,495	-6,723	0,216	0,426
12	16	6,425	3,545	-6,379	-3,449	0,046	0,096
14	15	1,451	0,734	-1,446	-0,729	0,005	0,005
16	17	2,773	1,590	-2,768	-1,572	0,005	0,018
15	18	5,559	1,744	-5,525	-1,675	0,034	0,069
18	19	2,297	0,768	-2,294	-0,761	0,004	0,007
19	20	-7,402	-2,720	7,423	2,760	0,020	0,040
10	20	9,809	3,702	-9,715	-3,492	0,094	0,210
10	17	6,348	4,333	-6,331	-4,287	0,018	0,046
10	21	16,561	10,163	-16,440	-9,904	0,120	0,259
10	22	8,063	4,647	-8,005	-4,528	0,058	0,119

21	22	-1,458	-1,571	1,459	1,572	0,000	0,001
15	23	4,766	3,071	-4,736	-3,010	0,030	0,060
22	24	6,547	2,956	-6,491	-2,870	0,056	0,087
23	24	1,471	1,373	-1,466	-1,362	0,005	0,010
24	25	-0,993	1,805	1,000	-1,791	0,008	0,013
25	26	3,540	2,364	-3,496	-2,297	0,045	0,067
25	27	-4,541	-0,572	4,563	0,615	0,022	0,042
28	27	17,729	5,164	-17,729	-3,916	0,000	1,248
27	29	6,144	1,657	-6,059	-1,496	0,085	0,161
27	30	7,022	1,645	-6,863	-1,345	0,159	0,300
29	30	3,654	0,594	-3,621	-0,533	0,033	0,062
8	28	-0,610	-0,474	0,612	-3,872	0,002	-4,347
6	28	18,398	0,168	-18,341	-1,292	0,056	-1,124

Tabla 3.49. Potencias en las Barras del Modelo de Carga Lineal.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO LINEAL					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVA _r]	[MW]	[MVA _r]	[MW]	[MVA _r]
1	2	172,421	-20,906	-167,286	30,446	5,135	9,540
1	3	87,346	4,749	-84,257	2,125	3,089	6,873
2	4	43,434	4,077	-42,427	-4,890	1,008	-0,813
3	4	81,705	-3,389	-80,857	4,955	0,847	1,565
2	5	82,231	1,764	-79,290	6,186	2,941	7,950
2	6	59,920	0,588	-57,999	1,300	1,921	1,888
4	6	71,171	-16,427	-70,552	17,660	0,619	1,233
5	7	-14,910	11,839	15,085	-13,463	0,175	-1,624
6	7	38,378	-3,047	-37,992	2,513	0,387	-0,534
6	8	29,558	-8,373	-29,448	7,840	0,110	-0,533
6	9	26,733	-7,977	-26,733	9,494	0,000	1,518
6	10	15,271	0,333	-15,271	0,861	0,000	1,194
9	11	0,000	-16,006	0,000	16,489	0,000	0,483
9	10	26,733	6,512	-26,733	-5,757	0,000	0,755
4	12	44,405	14,748	-44,405	-9,990	0,000	4,758
12	13	0,000	-11,525	0,000	11,692	0,000	0,167
12	14	8,249	2,652	-8,166	-2,479	0,083	0,172
12	15	18,336	7,371	-18,104	-6,914	0,232	0,457
12	16	6,589	3,660	-6,541	-3,559	0,048	0,101
14	15	1,455	0,752	-1,449	-0,747	0,005	0,005
16	17	2,841	1,697	-2,836	-1,678	0,005	0,019
15	18	5,678	1,812	-5,642	-1,739	0,036	0,072
18	19	2,435	0,822	-2,431	-0,814	0,004	0,008

19	20	-7,565	-2,757	7,586	2,799	0,021	0,042
10	20	10,068	3,770	-9,969	-3,548	0,099	0,222
10	17	6,210	4,340	-6,193	-4,296	0,017	0,045
10	21	17,345	10,604	-17,213	-10,320	0,132	0,284
10	22	8,381	4,826	-8,319	-4,697	0,062	0,129
21	22	-1,923	-1,773	1,924	1,774	0,001	0,002
15	23	4,790	3,120	-4,759	-3,058	0,030	0,062
22	24	6,395	2,923	-6,341	-2,840	0,053	0,083
23	24	1,449	1,427	-1,444	-1,416	0,005	0,011
24	25	-1,277	1,754	1,285	-1,739	0,009	0,015
25	26	3,524	2,355	-3,479	-2,288	0,044	0,066
25	27	-4,809	-0,616	4,834	0,663	0,025	0,048
28	27	17,998	5,265	-17,998	-3,997	0,000	1,288
27	29	6,147	1,663	-6,062	-1,502	0,085	0,161
27	30	7,017	1,651	-6,858	-1,351	0,159	0,300
29	30	3,644	0,596	-3,612	-0,535	0,033	0,062
8	28	-0,552	-0,302	0,555	-4,042	0,002	-4,344
6	28	18,610	0,104	-18,553	-1,222	0,057	-1,118

Tabla 3.50. Potencias en las Barras del Modelo de Carga de Motor de Inducción.

RESULTADOS DE POTENCIAS EN LAS BARRAS							
RAMA		MODELO MOTOR DE INDUCCIÓN					
DE	HASTA	Penvío	Qenvío	Precepción	Qrecepción	Pérdidas P	Pérdidas Q
#	#	[MW]	[MVA _r]	[MW]	[MVA _r]	[MW]	[MVA _r]
1	2	418,616	-60,868	-388,098	146,423	30,518	85,556
1	3	227,694	85,906	-203,709	-2,131	23,985	83,776
2	4	122,138	51,097	-112,844	-26,284	9,294	24,813
3	4	93,132	-77,118	-90,673	83,507	2,459	6,389
2	5	146,743	0,627	-137,397	34,236	9,347	34,863
2	6	137,517	18,939	-127,182	8,675	10,335	27,614
4	6	45,343	-125,415	-42,771	133,582	2,572	8,167
5	7	43,197	67,121	-40,260	-61,606	2,936	5,516
6	7	79,486	27,621	-77,407	-22,724	2,078	4,897
6	8	32,189	-129,843	-29,857	137,133	2,332	7,291
6	9	25,813	-18,006	-25,813	20,159	0,000	2,153
6	10	14,577	-2,783	-14,577	4,039	0,000	1,256
9	11	0,000	-31,300	0,000	33,267	0,000	1,966
9	10	25,813	11,142	-25,813	-10,303	0,000	0,839
4	12	42,419	-14,767	-42,419	20,247	0,000	5,480
12	13	0,000	-39,970	0,000	42,137	0,000	2,167
12	14	7,695	2,418	-7,617	-2,257	0,078	0,161
12	15	17,250	6,730	-17,030	-6,297	0,220	0,433

12	16	6,274	3,074	-6,229	-2,980	0,045	0,094
14	15	1,417	0,657	-1,412	-0,652	0,005	0,005
16	17	2,729	1,180	-2,725	-1,163	0,005	0,017
15	18	5,435	1,485	-5,401	-1,415	0,034	0,070
18	19	2,201	0,515	-2,197	-0,508	0,003	0,007
19	20	-7,303	-2,892	7,324	2,935	0,022	0,043
10	20	9,624	3,857	-9,524	-3,635	0,099	0,222
10	17	6,295	4,688	-6,275	-4,637	0,020	0,051
10	21	16,431	10,229	-16,303	-9,952	0,129	0,277
10	22	8,040	4,733	-7,978	-4,604	0,062	0,129
21	22	-1,197	-1,248	1,198	1,248	0,000	0,001
15	23	4,807	2,964	-4,775	-2,899	0,032	0,065
22	24	6,780	3,356	-6,713	-3,252	0,067	0,104
23	24	1,575	1,299	-1,569	-1,288	0,006	0,012
24	25	-0,418	1,967	0,426	-1,953	0,008	0,014
25	26	3,549	2,373	-3,500	-2,300	0,049	0,073
25	27	-3,975	-0,420	3,993	0,455	0,018	0,035
28	27	17,303	5,136	-17,303	-3,839	0,000	1,297
27	29	6,202	1,692	-6,107	-1,513	0,095	0,179
27	30	7,108	1,692	-6,929	-1,356	0,178	0,335
29	30	3,707	0,613	-3,671	-0,544	0,037	0,070
8	28	-0,143	20,446	0,462	-23,619	0,319	-3,173
6	28	17,888	-19,247	-17,765	18,483	0,123	-0,764

En las Fig. 3.16 y Fig. 3.17 se muestran los perfiles de voltaje de los distintos modelos de carga simulados y los máximos errores porcentuales de la magnitud de voltaje encontrados al comparar los resultados de la herramienta computacional con los resultados de DIgSILENT PowerFactory.

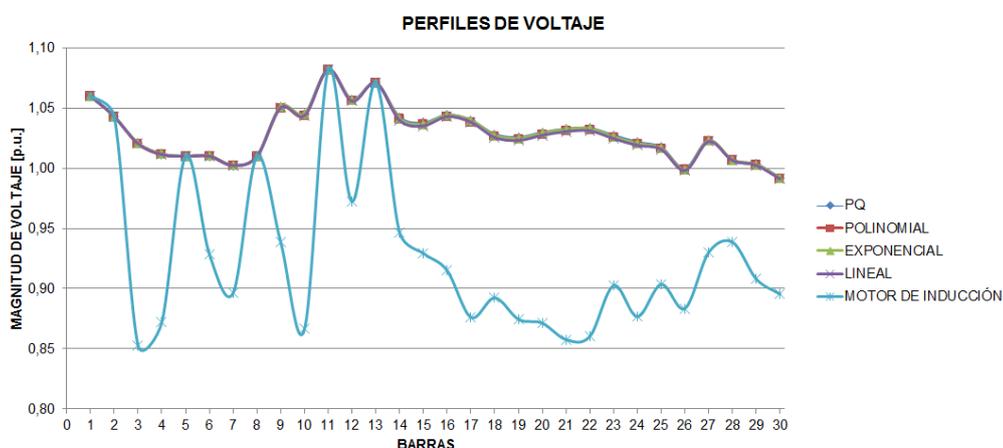


Figura 3.16. Perfiles de Voltaje del Sistema de 30 barras.

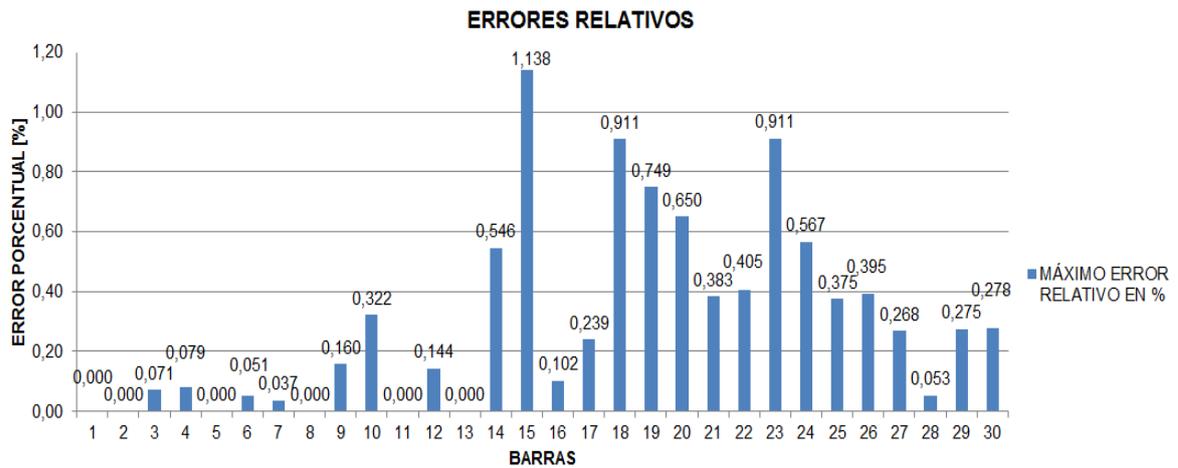


Figura 3.17. Máximos Errores relativos en Magnitud de Voltaje del Sistema de 30 barras.

3.4.5 Análisis de Resultados

En esta sección se presenta el análisis de los resultados obtenidos al resolver flujos de potencia con y sin la incorporación de modelos de carga. Este análisis se realiza mediante la comparación de los resultados de las variables de estado magnitud y ángulo de voltaje ya que en base a estas se calculan los flujos de potencia en las ramas del sistema.

De los resultados de la sección 3.4 se observa que al incorporar modelos de carga en el cálculo de flujos de potencia tanto el número de iteraciones realizadas como el tiempo de ejecución aumentan. Se observa también que la metodología de Reestructuración de Potencias Calculadas utiliza un menor número de iteraciones para obtener el resultado del flujo de potencia. Por otra parte se aprecia que la metodología de Inyección Nodal es la más rápida debido a que los tiempos de ejecución por iteración son más pequeños en función de que el Jacobiano permanece constante durante el proceso. También se observa que el efecto de la incorporación de los modelos de carga se ve reflejado en el hecho de que las magnitudes y ángulos de voltaje cambian respecto de los resultados obtenidos en flujos de potencia simples, en cuanto a las potencias en las ramas se observan que en la mayoría de los casos analizados incorporando modelos las potencias activas y reactivas de envío y recepción así como las pérdidas disminuyen su valor excepto en el caso que considera modelo de motor de inducción en el cual aumentan.

Para realizar el análisis de los resultados en la Tabla 3.51 se muestran los máximos errores obtenidos al comparar los resultados arrojados por el programa desarrollado en Python con los resultados de los softwares comerciales PSAT y DlgSILENT PowerFactory.

Tabla 3.51. Máximos errores obtenidos.

SISTEMA DE 2 BARRAS (Línea R-L)		
MODELO DE CARGA	$error_V$	$error_\theta$
Tipo	[p.u.]	[°]
Potencia Constante	0,0000	0,0000
Polinomial	0,0000	0,0029
Exponencial	0,0000	0,0000
Lineal	0,0000	0,0000
Motor de Inducción	0,0000	0,0000
SISTEMA DE 2 BARRAS (Línea L)		
Potencia Constante	0,0000	0,0000
Polinomial	0,0000	0,0053
Exponencial	0,0000	0,0000
Lineal	0,0000	0,0000
Motor de Inducción	0,0000	0,0000
SISTEMA DE 4 BARRAS		
Potencia Constante	0,0000	0,0000
Polinomial	0,0000	0,0003
Exponencial	0,0000	0,0000
Lineal	0,0000	0,0000
Motor de Inducción	0,0000	0,0001
SISTEMA DE 14 BARRAS		
Potencia Constante	0,0009	0,0113
Polinomial	0,0010	0,0155
Exponencial	0,0009	0,0446
Lineal	0,0010	0,0159
Motor de Inducción	0,0009	0,0169
SISTEMA DE 30 BARRAS		
Potencia Constante	0,0057	0,1437
Polinomial	0,0057	0,1652
Exponencial	0,0052	0,7507
Lineal	0,0052	0,7781
Motor de Inducción	0,0051	0,8354

Al analizar los resultados de la Tabla 3.51 se observa que los errores obtenidos son aceptables. Por otra parte se aprecia que los resultados obtenidos mediante el programa desarrollado en Python son similares a los resultados obtenidos en los softwares comerciales PSAT y DlgSILENT PowerFactory validando así las metodologías de Inyección Nodal, Flujos Sucesivos y Reestructuración de Potencias Calculadas utilizadas para incorporar modelos de carga en estado estable al cálculo de flujos de potencia.

La razón por la cual se presentan errores entre el programa desarrollado y los softwares comerciales PSAT y DIgSILENT PowerFactory no puede ser determinada debido a que se desconoce la manera en la que los softwares comerciales realizan la incorporación de modelos de carga en el cálculo de flujos de potencia.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- En el presente trabajo de titulación se incorporaron al cálculo de flujos de potencia los modelos de carga polinomial, exponencial, lineal y de motor de inducción mediante las metodologías de Inyección Nodal, Flujos Sucesivos y Reestructuración de Potencias Calculadas. Se demuestra que las tres metodologías para todos los modelos arrojan resultados similares validando de esta manera el programa desarrollado en Python.
- Al utilizar las metodologías de Inyección Nodal, Flujos Sucesivos y Reestructuración de Potencias Calculadas para resolver flujos de potencia considerando la incorporación de modelos de carga se observa que el número de iteraciones y el tiempo de ejecución aumentan con respecto al número de iteraciones y tiempo de ejecución obtenidos al resolver flujos de potencia sin incorporar modelos de carga, determinándose de esta manera que la metodología más rápida para incorporar modelos de carga en la solución del flujo de potencia es la de Inyección Nodal.
- Luego de analizar los resultados obtenidos en el cálculo de flujos de potencia con y sin la incorporación de modelos de carga, se encontró una variación leve en magnitudes de voltaje (segundo y tercer decimal) y una variación considerable en los valores de los ángulos de voltaje y las potencias de las ramas del sistema.
- Después de analizar los resultados obtenidos en el cálculo de flujos de potencia con y sin la incorporación de modelos de carga, se encontró que la incorporación de modelos de carga en el cálculo de flujos de potencia agrega mayor precisión a los resultados obtenidos, permitiendo así analizar de mejor manera lo que en realidad sucede en el sistema.
- Se encontró que las metodologías de Inyección Nodal y Reestructuración de Potencias Calculadas tienen grandes ventajas frente a la metodología de Flujos Sucesivos, ya que tanto los tiempos de cálculo del algoritmo como el número de iteraciones son más pequeños y presentan la misma exactitud en los resultados.
- Se encontró que a pesar de que la metodología de Inyección Nodal presenta una mayor rapidez en el cálculo del flujo de potencia, la metodología de Reestructuración de Potencias Calculadas presenta una mejor convergencia ya que se realiza un menor número de iteraciones para obtener los resultados.

- El programa desarrollado en el presente trabajo de titulación puede ser utilizado por los estudiantes de Ingeniería Eléctrica para conocer cómo se modela la carga en estado estable, y como los modelos deben ser incorporados al cálculo de flujos de potencia, además el programa puede ser revisado y modificado a conveniencia ya que se encuentra desarrollado en el software libre de código abierto Python.

4.2 Recomendaciones

- Se recomienda seguir utilizando Python para el desarrollo de trabajos de titulación futuros, ya que al ser un software libre ofrece grandes ventajas como la creación y modificación de programas por diferentes personas, además es un software completo que posee un gran número de herramientas.
- Se recomienda el uso de los modelos de carga polinomial y exponencial debido a que existe un mayor número de estudios al respecto, razón por la cual resulta más fácil obtener o determinar los parámetros de la carga que se desee modelar.
- Se recomienda plantear proyectos para determinar los parámetros de las cargas del sistema eléctrico ecuatoriano, ya que de esta manera se podrá incorporar modelos de carga a las bases de datos utilizadas en estudios eléctricos.
- Se recomienda que los estudiantes de Ingeniería Eléctrica de la Escuela Politécnica Nacional conozcan la manera y los efectos de modelar carga dentro de los estudios de flujo de potencia, para ello se propone la realización de una práctica sobre el tema en el laboratorio de Introducción a Sistemas Eléctricos de Potencia.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Milanovic, A. Borghetti, J. Matevosyan, A. Halley, and W. Hung, "Modelling and Aggregation of Loads in Flexible Power Networks, presentado por CIGRE WG C4.605" Feb. 2014.
- [2] A. Bonavita, "Software Libre." .
- [3] J. J. Grainger and W. D. Stevenson, *Power System Analysis*. McGraw_Hill, 1994.
- [4] "Technical Reference Documentation about Asynchronous Machine." DlgSILENT PowerFactory.
- [5] A. R. Bin Azizan, "Simulation of Dynamic Load Effect on Power System Frequency," University Tun Hussein Onn Malaysia, 2013.
- [6] P. Kundur, *Power System Stability and Control*. New York: McGraw_Hill, 1994.
- [7] L. M. Hajagos and B. Danai, "Laboratory Measurements and Models of Modern Loads and their Effect on Voltage Stability Studies," May 1998.
- [8] J. R. Caiza Quiroz, "Determinación Experimental de las Características de los Modelos de Carga de Artefactos Eléctricos Modernos de Uso Residencial," Escuela Politécnica Nacional, Quito, 2017.
- [9] F. Nozari, D. Kankam, and W. Price, "Aggregation of Induction Motors for Transient Stability Load Modeling."
- [10] R. Patiño Bedolla, "Análisis y Control de Flujos de Potencia en estado estable," Universidad Autónoma de Nuevo León, 1995.
- [11] G. Venkataramanan *et al.*, "Final Project Report Load Modeling Transmission Research," Mar. 2010.
- [12] M. E. El-Hawary and L. G. Dias, "Incorporation of load models in load flow studies: Form of model effects," Apr. 1986.
- [13] K. Sun, "Load Modeling," Spring, 2015.
- [14] A. Polycarpou, S. M. Stavrinou, A. G. Petoussis, and M. A. Michael, "The Impact of Load Modeling on Power Flow Studies for the Cyprus Power System," Oct. 2014.
- [15] L. Quezada and M. Leonardo, "Modelos de carga y sus efectos en estudios sistémicos: Aplicaciones al SING," Universidad de Chile, Santiago de Chile, 2013.
- [16] M. E. El-Hawary and L. G. Dias, "A comparison of load models and their effect on the convergence of Newton's power flows," Jan. 1990.
- [17] M. E. El-Hawary and L. G. Dias, "Bus sensitivity to load model parameters in load flow studies," Jul. 1987.
- [18] M. H. Haque, "Load flow Solution of distribution systems with voltage dependent load models," Oct. 1995.

- [19] J. Stanko and B. Ivanovic, "Application of combined Newton Raphson method to large load flow models," 2015.
- [20] P. Borne and D. Patherton, *Advances in Systems: Control and Information Engineering*. Pergamon Press.
- [21] M. E. El-Hawary and L. G. Dias, "Effects of Load Modeling in Security Constrained OPF studies," Feb. 1991.
- [22] R. González Duque, "Python para todos." .
- [23] "Power Systems Test Case Archive." University of Washington.
- [24] J. Játiva, G. Constante, and G. Cabrera, "Flujo de Potencia por Newton-Raphson con el Jacobiano Calculado en las Ecuaciones de Errores de Potencia," Jan. 2014.
- [25] C. Wen-Chen, *Load Model of Power System*. University of Texas at Arlington, 1986.
- [26] F. Quilumba, W.-J. Lee, Y. Wang, and R. Szabados, "Load Model Development for Next Generation Appliances," 2011.
- [27] H. Ying-Yi and C. Li, "Impacts of Load Models on Solutions of Newton Optimal Power Flow," 1993.
- [28] P. Aree, "Power Flow Computation Considering Nonlinear Characteristics of Composite Load Model," 2014.
- [29] R. Zimmerman and C. Murillo, *MATPOWER 6.0 User's Manual*. 2016.
- [30] F. Milano, *Power System Modelling and Scripting*, vol. 0. Berlin: Springer, 2010.
- [31] M. González, "Incorporación de Dispositivos FACTS y Transmisión HVDC en el Cálculo de Flujos de Potencia," Escuela Politécnica Nacional, Quito, 2016.

6. ANEXOS

ANEXO I. Práctica Propuesta para el Laboratorio de Introducción a Sistemas Eléctricos de Potencia (IEE602).

LABORATORIO DE INTRODUCCIÓN A SISTEMAS ELÉCTRICOS DE POTENCIA PRÁCTICA N° 01

1. TEMA

CÁLCULO DE FLUJOS DE POTENCIA CONSIDERANDO LA INCORPORACIÓN DE MODELOS DE CARGA.

2. OBJETIVOS

- 2.1. Estudiar los modelos de carga en estado estable utilizados en el cálculo de flujos de potencia.
- 2.2. Observar los efectos de la incorporación de modelos de carga en el cálculo de flujos de potencia.
- 2.3. Comparar los resultados y establecer diferencias al calcular flujos de potencia con y sin la incorporación de modelos de carga.

3. MARCO TEÓRICO

3.1. Modelo de Carga

Un modelo de carga es la representación analítica y matemática basada en un circuito equivalente o en un componente físico que puede ser utilizado para el análisis, la predicción o estimación de las características de carga pertinentes en los estudios de sistemas de potencia.

3.2. Tipos de Modelos de Carga

Los modelos de carga pueden representarse tanto en estado estable como en estado dinámico, dependiendo del tipo de estudio que se esté realizando.

Los modelos de carga frecuentemente usados para estudios en estado estable y dinámico son los siguientes:

Modelos de carga en estado estable:

- Modelo de carga Exponencial.
- Modelo de carga Polinomial.
- Modelo de carga Lineal.
- Modelo de carga Completo.
- Modelo de carga del Motor de Inducción.
- Modelo de carga con Interfaz de Electrónica de Potencia.

Modelos de carga dinámicos:

- Modelo de carga Dinámico con Motor de Inducción.
- Modelo de carga Dinámico Exponencial.
- Modelo de carga Compuesto.
- Modelo de Carga Función de Transferencia con Motor de Inducción.
- Modelo de Carga de Distribución.
- Modelo de carga con Barra de carga de potencia.
- Modelo de carga de Sistema de Almacenamiento de Energía Distribuido (DESS)

4. TRABAJO PREPARATORIO

4.1. Consultar las ecuaciones, modelo matemático y restricciones del modelo de carga polinomial.

4.2. Consultar las ecuaciones, modelo matemático y restricciones del modelo de carga exponencial.

4.3. Consultar las ecuaciones, modelo matemático y restricciones del modelo de carga lineal.

4.4. Consultar los parámetros para representar mediante modelos de carga polinomial y exponencial las siguientes cargas:

- DVD.
- Celular.

- Microondas.
- Equipo de Sonido.
- Televisor LED.

4.5. Traer armado en DlgSILENT PowerFactory el sistema de cuatro barras mostrado en la página 317 del libro “Power System Analysis” de Grainger y Stevenson.

5. EQUIPO Y MATERIALES

- Computadores del Laboratorio de Sistemas Eléctricos de Potencia
- Software DlgSILENT PowerFactory.

6. PROCEDIMIENTO

6.1. Correr el flujo de potencia para el sistema de cuatro barras.

6.2. Incorporar los parámetros del modelo de carga polinomial consultados en la sección 4.4 del Trabajo Preparatorio en las cargas del sistema de cuatro barras y ejecutar la simulación del flujo de potencia considerando modelos de carga.

6.3. Incorporar los parámetros del modelo de carga exponencial consultados en la sección 4.4 del Trabajo Preparatorio en las cargas del sistema de cuatro barras y ejecutar la simulación del flujo de potencia considerando modelos de carga.

7. INFORME

7.1. Realizar un tutorial en el cual se explique cómo incorporar modelos de carga en el cálculo de Flujos de potencia utilizando DlgSILENT PowerFactory.

7.2. En el sistema de cuatro barras, ejecutar el cálculo de flujos de potencia con y sin la incorporación del modelo de carga polinomial utilizando los siguientes parámetros:

MODELO DE CARGA POLINOMIAL						
N° BARRA	p_1	p_2	p_3	q_1	q_2	q_3
1	-0,725	1,863	-0,137	-0,630	1,815	-0,185
2	-1,304	3,093	-0,789	-0,966	2,641	-0,675
3	-0,135	1,235	-0,100	0,219	0,742	0,040
4	0,000	0,000	1,000	-0,121	0,797	0,325

7.3. En el sistema de cuatro barras, ejecutar el cálculo de flujos de potencia con y sin la incorporación del modelo de carga exponencial utilizando los siguientes parámetros:

MODELO EXPONENCIAL		
N° BARRA	k_{pu}	k_{qu}
1	0,740	0,857
2	1,323	1,431
3	0,686	0,748
4	0,000	0,232

7.4. Conclusiones y recomendaciones (Se aconseja escribir por lo menos cuatro conclusiones de la práctica)

7.5. Bibliografía.

8. REFERENCIAS

[1] C. Jiménez, "Incorporación de Modelos de Carga en Estable para el Cálculo de Flujos de Potencia", Escuela Politécnica Nacional, Quito, 2018.

[2] J. Caiza, "Determinación Experimental de las Características de los Modelos de Carga de Artefactos Eléctricos Modernos de Uso Residencial", Escuela Politécnica Nacional, Quito, 2017.

[3] J. J. Grainger y W. D. Stevenson, "Power System Analysis", McGraw Hill, 1994.

Elaborado por: Sr. Cristian Jiménez

Revisado por: Dr. Hugo Arcos

ANEXO II. Parámetros de los Modelos de Carga utilizados en las Simulaciones

Tabla A2.1. Parámetros del modelo de carga polinomial.

SISTEMA DE 2 BARRAS						
MODELO POLINOMIAL						
N° BARRA	p_1	p_2	p_3	q_1	q_2	q_3
#	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[p.u.]
2	-0,725	1,863	-0,137	-0,630	1,815	-0,185
SISTEMA DE 4 BARRAS DE GRAINGER Y STEVENSON						
N° BARRA	p_1	p_2	p_3	q_1	q_2	q_3
1	-0,725	1,863	-0,137	-0,630	1,815	-0,185
2	-1,304	3,093	-0,789	-0,966	2,641	-0,675
3	-0,135	1,235	-0,100	0,219	0,742	0,040
4	0,000	0,000	1,000	-0,121	0,797	0,325
SISTEMA IEEE DE 14 BARRAS						
N° BARRA	p_1	p_2	p_3	q_1	q_2	q_3
2	0,000	0,000	1,000	-0,630	1,815	-0,185
3	0,000	0,000	1,000	-0,966	2,641	-0,675
4	-0,135	1,235	-0,100	0,219	0,742	0,040
5	0,174	0,066	0,760	-0,121	0,797	0,325
6	0,000	0,000	1,000	-0,883	2,595	-0,712
9	-0,793	2,073	-0,280	-0,693	2,010	-0,317
10	-1,223	3,059	-0,837	-0,883	2,595	-0,712
11	-0,119	1,293	-0,174	0,306	0,682	0,012
12	0,255	0,050	0,695	-0,066	0,793	0,274
13	-0,833	2,072	-0,240	-0,721	1,998	-0,277
14	-0,167	1,329	-0,162	0,209	0,798	-0,007
SISTEMA IEEE DE 30 BARRAS						
N° BARRA	p_1	p_2	p_3	q_1	q_2	q_3
2	0,000	0,000	1,000	-0,630	1,815	-0,185
3	0,174	0,066	0,760	-0,121	0,797	0,325
4	-0,135	1,235	-0,100	0,219	0,742	0,040
5	0,000	0,000	1,000	-0,966	2,641	-0,675
7	-0,793	2,073	-0,280	-0,693	2,010	-0,317
8	0,000	0,000	1,000	-0,883	2,595	-0,712
10	-0,119	1,293	-0,174	0,306	0,682	0,012
12	0,255	0,050	0,695	-0,066	0,793	0,274
14	-0,833	2,072	-0,240	-0,721	1,998	-0,277
15	-0,167	1,329	-0,162	0,209	0,798	-0,007
16	0,211	0,084	0,705	-0,083	0,778	0,305
17	0,265	0,084	0,651	-0,040	0,768	0,271
18	-0,912	2,264	-0,352	-0,790	2,170	-0,379

19	-1,187	3,033	-0,846	-0,851	2,570	-0,719
20	-1,187	3,033	-0,846	-0,851	2,570	-0,719
21	-0,173	1,397	-0,224	0,246	0,794	-0,041
23	-0,918	2,196	-0,278	-0,790	2,099	-0,309
24	-1,232	3,047	-0,815	-0,897	2,592	-0,695
26	-1,328	3,116	-0,788	-0,987	2,661	-0,674
29	-0,192	1,385	-0,193	0,188	0,848	-0,036
30	-1,268	3,065	-0,796	-0,934	2,614	-0,681

Tabla A2.2. Parámetros del modelo de exponencial.

SISTEMA DE 2 BARRAS		
MODELO EXPONENCIAL		
N° BARRA	k_{pu}	k_{qu}
#	[p.u.]	[p.u.]
2	0,740	0,857
SISTEMA DE 4 BARRAS DE GRAINGER Y STEVENSON		
N° BARRA	k_{pu}	k_{qu}
1	0,740	0,857
2	1,323	1,431
3	0,686	0,748
4	0,000	0,232
SISTEMA IEEE DE 14 BARRAS		
N° BARRA	k_{pu}	k_{qu}
2	0,000	0,857
3	0,000	1,431
4	0,686	0,748
5	0,236	0,232
6	0,000	0,932
9	1,404	1,513
10	0,292	0,260
11	0,777	0,895
12	1,369	1,478
13	0,695	0,753
14	0,278	0,257
SISTEMA IEEE DE 30 BARRAS		
N° BARRA	k_{pu}	k_{qu}
2	0,000	0,857
3	1,323	1,431
4	0,686	0,748
5	0,236	0,232
7	0,813	0,932
8	0,000	1,513

10	0,691	0,750
12	0,292	0,260
14	0,777	0,895
15	1,369	1,478
16	0,695	0,753
17	0,278	0,257
18	0,323	0,281
19	0,827	0,947
20	1,427	1,537
21	0,702	0,756
23	0,779	0,897
24	1,377	1,486
26	1,312	1,420
29	0,701	0,756
30	1,344	1,452

Tabla A2.3. Parámetros del modelo de lineal.

SISTEMA DE 2 BARRAS				
MODELO LINEAL				
N° BARRA	a_0	a_2	b_0	b_2
#	[p.u.]	[p.u.]	[p.u.]	[p.u.]
2	-0,863	1,863	-0,815	1,815
SISTEMA DE 4 BARRAS DE GRAINGER Y STEVENSON				
N° BARRA	a_0	a_2	b_0	b_2
1	-0,863	1,863	-0,815	1,815
2	-2,093	3,093	-1,641	2,641
3	-0,235	1,235	0,258	0,742
4	1,000	0,000	0,203	0,797
SISTEMA IEEE DE 14 BARRAS				
N° BARRA	a_0	a_2	b_0	b_2
2	1,000	0,000	-0,815	1,815
3	1,000	0,000	-1,641	2,641
4	-0,235	1,235	0,258	0,742
5	-1,073	2,073	-1,010	2,010
6	1,000	0,000	-1,595	2,595
9	-0,293	1,293	0,318	0,682
10	0,950	0,050	0,207	0,793
11	-1,072	2,072	-0,998	1,998
12	-2,063	3,063	-1,607	2,607
13	-0,385	1,385	0,152	0,848
14	-0,329	1,329	0,202	0,798
SISTEMA IEEE DE 30 BARRAS				

N° BARRA	a_0	a_2	b_0	b_2
2	1,000	0,000	-0,815	1,815
3	-2,093	3,093	-1,641	2,641
4	-0,235	1,235	0,258	0,742
5	0,934	0,066	0,203	0,797
7	-1,073	2,073	-1,010	2,010
8	1,000	0,000	-1,595	2,595
10	-0,293	1,293	0,318	0,682
12	0,950	0,050	0,207	0,793
14	-1,072	2,072	-0,998	1,998
15	-2,063	3,063	-1,607	2,607
16	-0,329	1,329	0,202	0,798
17	0,916	0,084	0,222	0,778
18	0,916	0,084	0,232	0,768
19	-1,264	2,264	-1,170	2,170
20	-2,033	3,033	-1,570	2,570
21	-2,065	3,065	-1,614	2,614
23	-0,397	1,397	0,206	0,794
24	-1,196	2,196	-1,099	2,099
26	-2,047	3,047	-1,592	2,592
29	-2,116	3,116	-1,661	2,661
30	-0,385	1,385	0,152	0,848

Tabla A2.4. Parámetros del modelo de lineal.

PARA TODOS LOS SISTEMAS							
MODELO MOTOR DE INDUCCIÓN							
N° BARRA	R_s	X_{ys}	X_m	R_{yr}	X_r	s	S
#	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[p.u.]	[MVA]
i	0,01300	0,14000	2,40000	0,00900	0,12000	0,01551	175,35

ANEXO III. Sistemas de Prueba Modelados en DigSILENT PowerFactory y PSAT

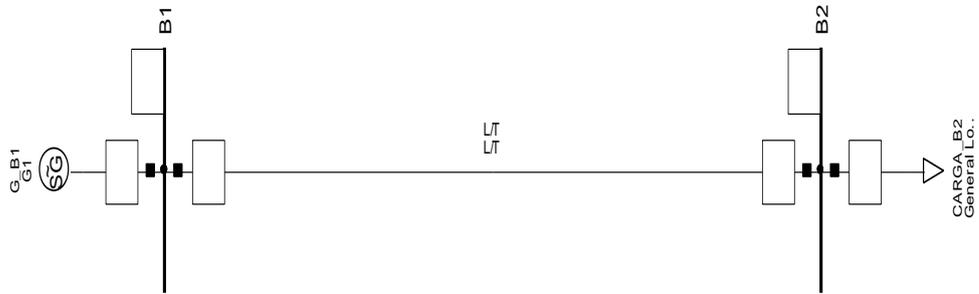


Figura A3.1. Sistema de Prueba de 2 barras modelado en DigSILENT PowerFactory.

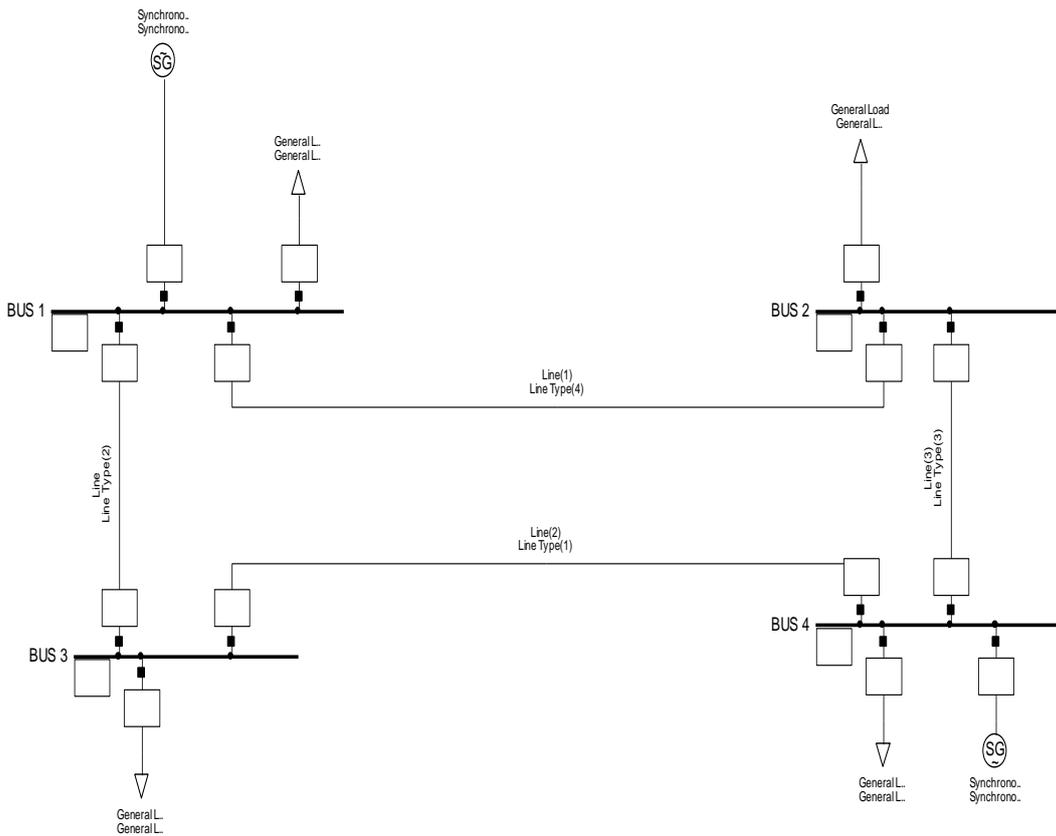


Figura A3.2. Sistema de Prueba de 4 barras modelado en DigSILENT PowerFactory.

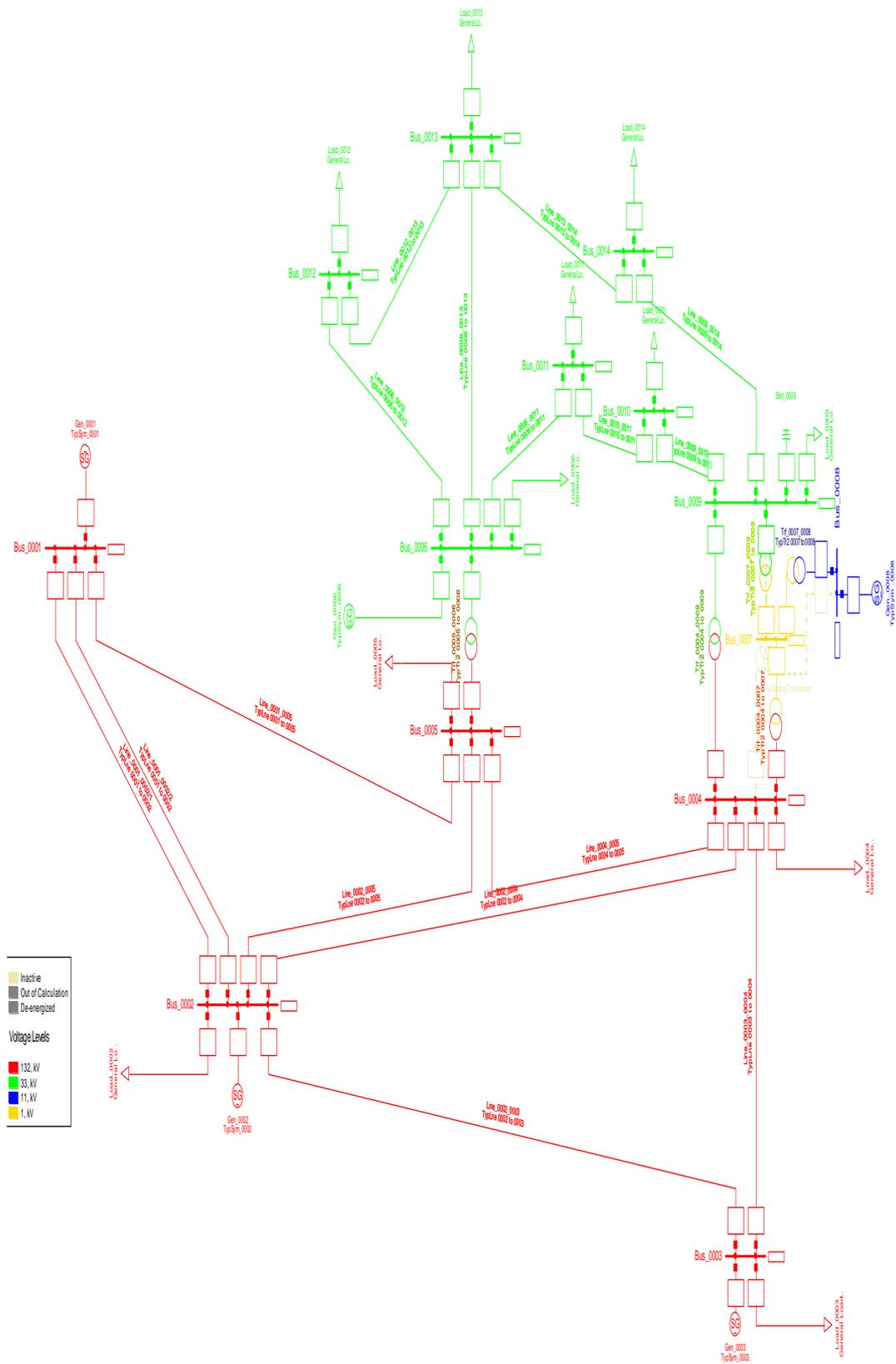


Figura A3.3. Sistema de Prueba de 14 barras modelado en DlgSILENT PowerFactory.

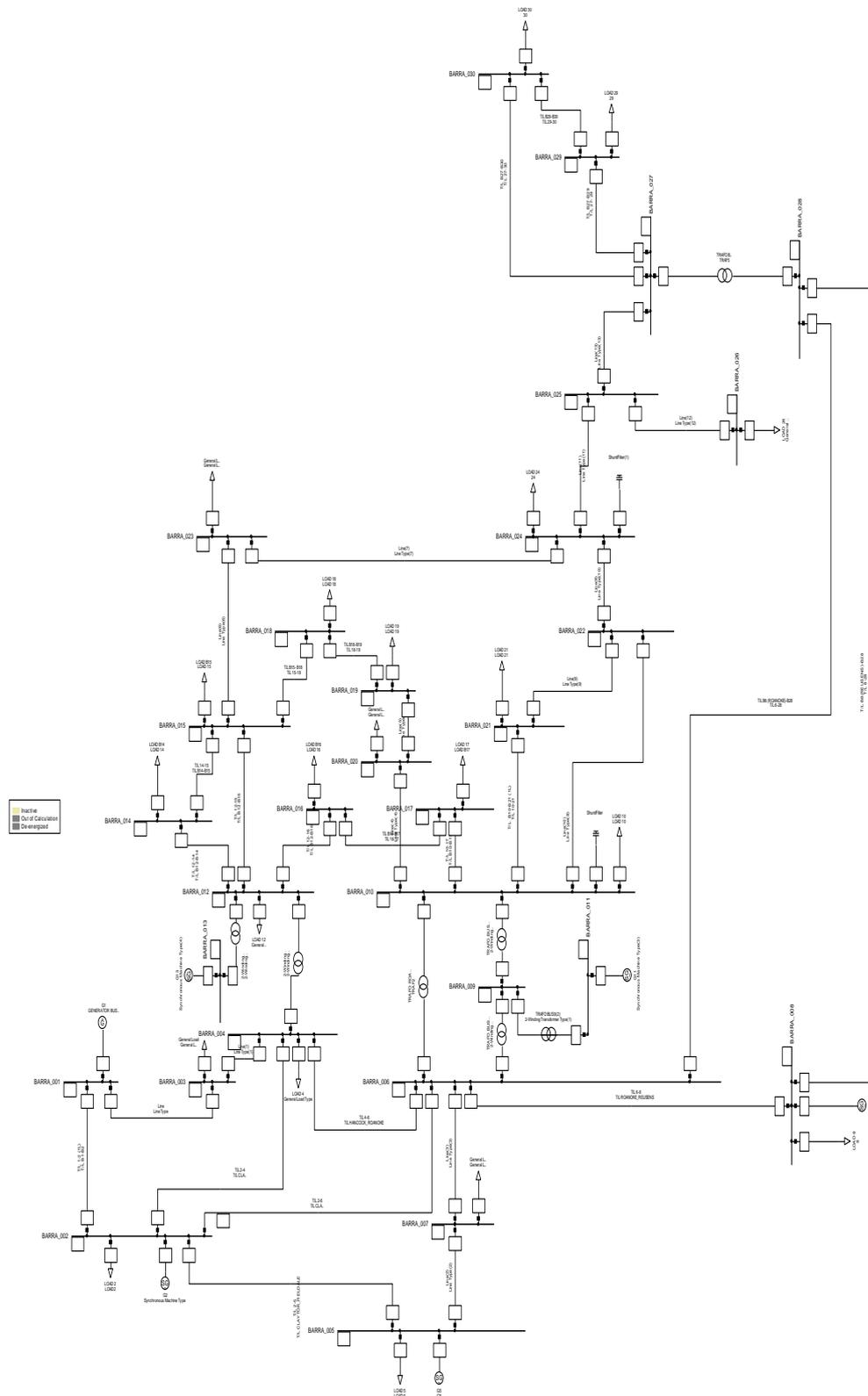


Figura A3.4. Sistema de Prueba de 30 barras modelado en DIgSILENT PowerFactory.

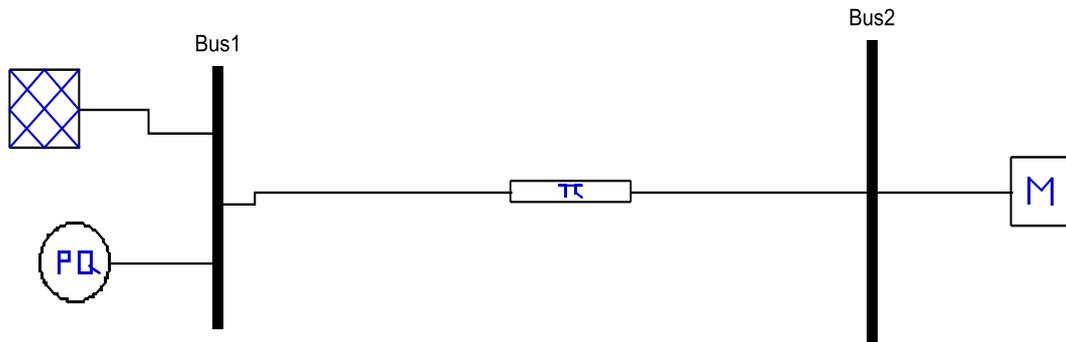


Figura A3.5. Sistema de Prueba de 2 barras modelado en PSAT.

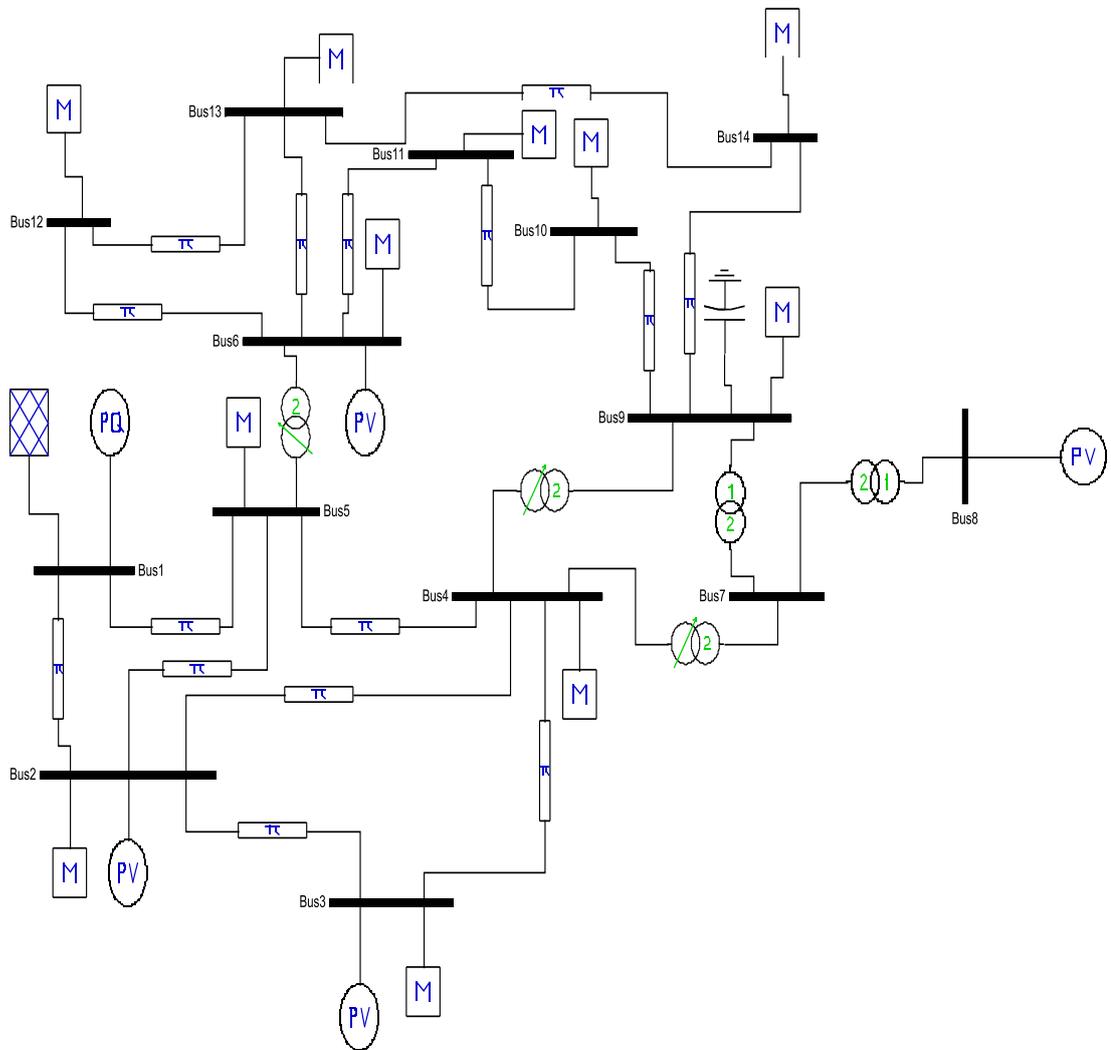


Figura A3.6. Sistema de Prueba de 14 barras modelado en PSAT.

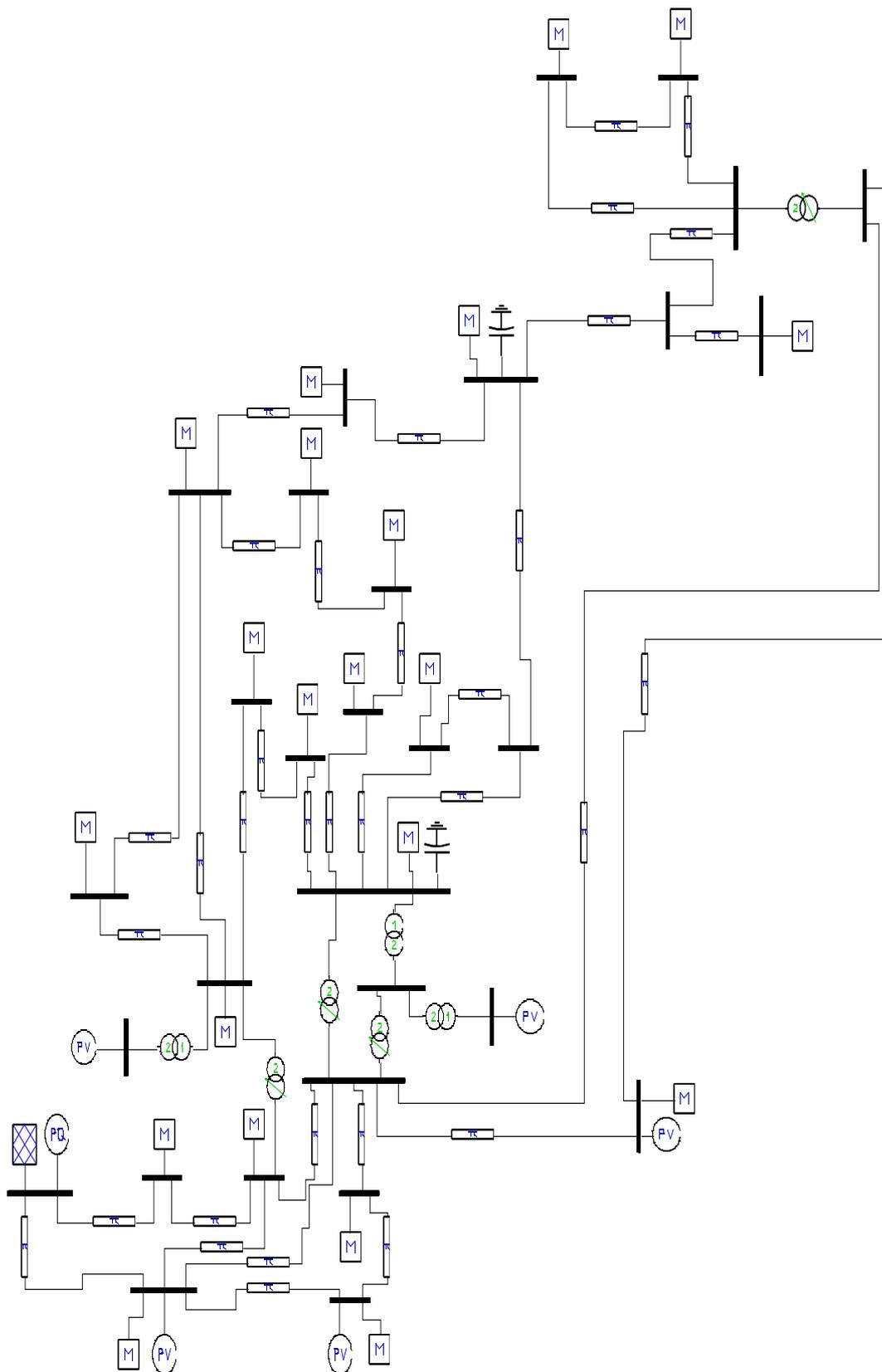


Figura A3.7. Sistema de Prueba de 30 barras modelado en PSAT.

ANEXO IV. Descripción y Ejecución del Código desarrollado en Python

En esta sección se presenta el código desarrollado en Python, el mismo que fue programado en forma de bloques para un mejor entendimiento y una mejor visualización del mismo. A continuación se muestran los bloques del programa y una descripción de la función que realizan los mismos:

- **case_.py:**

```

"""
SEP de 4 barras propuesto en el libro "Power System Analysis"
de Grainger y Stevenson pag:318-320
"""

from numpy import matrix
"""
Datos de las barras
"""
# ***** BUSBAR DATA *****
# #NUMBER Type Voltage Angle PLoad QLoad Bsh MinV MaxV
BD =matrix(
    ' 1 1 1.000 0.000 0.500 0.3099 0.00 0.95 1.05 ;'
    ' 2 3 1.000 0.000 1.700 1.0535 0.00 0.95 1.05 ;'
    ' 3 3 1.000 0.000 2.000 1.2394 0.00 0.95 1.05 ;'
    ' 4 2 1.020 0.000 0.800 0.4958 0.00 0.95 1.05 ;'
)

"""
Datos de las branches: L/T y Transformadores
"""
# ***** LINE DATA *****
# FROM TO R[pu] X[pu] B[pu] STATUS TAP
LD =matrix(
    ' 1 2 0.01008 0.05040 0.1025 1 0.00 ;'
    ' 1 3 0.00744 0.03720 0.0775 1 0.00 ;'
    ' 2 4 0.00744 0.03720 0.0775 1 0.00 ;'
    ' 3 4 0.01272 0.06360 0.1275 1 0.00 ;'
)

"""
Datos de los Generadores
"""
# ***** GENERATOR DATA *****
# #NUMBER PGEN QGEN QMIN QMAX STATUS
GD = matrix(
    ' 1 0.00 0.00 -1.00 1.00 1 ;'
    ' 4 3.18 0.00 -1.00 1.50 1 ;'
)

"""
Datos del Modelo de Carga Polinomial
"""
# ***** LOAD MODELLING DATA (POLINOMIAL MODEL) *****
# NUMBER Type UZ UI UP WZ WI WP
PLMD = matrix(
    ' 1 1 -0.725 1.863 -0.137 -0.630 1.815 -0.185 ;'
    ' 2 3 -1.304 3.093 -0.789 -0.966 2.641 -0.675 ;'
    ' 3 3 -0.135 1.235 -0.100 0.219 0.742 0.040 ;'
    ' 4 2 0.000 0.000 1.000 -0.121 0.797 0.325 ;'
)

```

Figura A4.1. Código de la función case4gs.py.

Este bloque permite el ingreso de los datos del sistema a simularse. Los datos ingresan en forma de matriz, cada archivo debe contener tres matrices si se desea ejecutar el flujo de potencia sin considerar la incorporación de modelos de carga o cuatro matrices si se desea incorporar modelos de carga en el cálculo de flujos de potencia. Las matrices a ingresarse son las siguientes:

- ✓ BD: (Buses Data) contiene los datos de las barras del sistema.
- ✓ LD: (Lines Data) contiene los datos de las ramas del sistema.
- ✓ GD:(Generators Data) contiene los datos de los generadores del sistema.

- ✓ PLMD:(Polinomial Load Model Data) contiene los parámetros para modelar las cargas mediante el modelo de carga polinomial.
- ✓ ELMD:(Exponencial Load Model Data) contiene los parámetros para modelar las cargas mediante el modelo de carga exponencial.
- ✓ LLMD:(Lineal Load Model Data) contiene los parámetros para modelar las cargas mediante el modelo de carga lineal.
- ✓ IMLMD:(Induction Motor Load Model Data) contiene los parámetros para modelar las cargas mediante el modelo de carga de motor de inducción.

- **sys_var_idx.py:**

```

"""
Este archivo contiene los índices
de las columnas de las matrices de ingreso
"""
#Polinomial Load model indexing
BUSL_ID = 0 # Bus number
BUSL_TP = 1 # Bus type
UZ=2      # % of impedance in active power
UI=3      # % of current in active power
UP=4      # % of power in active power
WZ=5      # % of impedance in reactive power
WI=6      # % of current in reactive power
WP=7      # % of power in reactive power
# Branches indexing
global FR_B,TO_B
FR_B = 0 #From bus
TO_B = 1 #To bus
BR_R = 2 #Branch per unit resistance
BR_X = 3 #Branch per unit reactance
BR_S = 4 #Branch total susceptance
BR_ST = 5 #Branch service status: 1 on service, 0 out of service
TAP_H = 6 #Transformer TAP 1
TAP_L = 7 #Transformer TAP 2

# Bus indexing
global BUS_ID
BUS_ID = 0 # Bus number
BUS_TP = 1 # Bus type
VolM = 2 # Voltage Magnitude
VolA = 3 # Voltage Angle
PL = 4 # Load real power.
QL = 5 # Load reactive load.
SH_B = 6 # Total Shun compensation
V_MIN = 7 # Minimum Bus Voltage magnitude
V_MAX = 8 # Maximum Bus Voltage magnitude

# Generator indexing
global GEN_B
GEN_B = 0 # Generator Bus
PG = 1 # Real Power Output
QG = 2 # Reactive Power Output
QMIN = 3 # Generator minimum reactive power
QMAX = 4 # Generator maximum reactive power
GEN_ST = 5 # Generator status in/out

#Busbar type
global SL,PV,PQ
SL = 1 # Generator Bus
PV = 2 # Real Power Output
PQ = 3 # Reactive Power Output

```

Figura A4.2. Código de la función sys_var_idx.py.

Esta función contiene la representación y significado de cada una de las columnas de las matrices de los datos de entrada. Cabe notar que se empieza a enumerar desde cero debido a que Python considera el conteo de índices desde cero.

- **info.py:**

```

"""
INCORPORACIÓN DE MODELOS DE CARGA EN ESTADO ESTABLE
información para impresión
"""

def info(*args):
    pver = {'Name': 'INCORPORACIÓN DE MODELOS DE CARGA EN ESTADO ESTABLE',
            'Version': '1.0.0',
            'Date': '02-Mayo-2018'}
    return pver

```

Figura A4.3. Código de la función info.py.

Este archivo contiene información como nombre, versión y fecha de presentación del programa.

- **loadcase.py:**

```

import sys
from os.path import basename, splitext, exists

basestring = str

def loadcase(casefile):

    info = 0

    if isinstance(casefile, basestring):
        if casefile.endswith('.py'):
            rootname, extension = splitext(casefile)
        else:
            rootname = casefile
            if exists(casefile + '.py'):
                extension = '.py'
            else:
                info = 2
                fname = basename(rootname)

    if info == 0:
        if extension == '.py':
            exec(compile(open(rootname + extension).read(),
                            rootname + extension, 'exec'))

            data = locals()
            branch_data = data['LD']
            busbar_data = data['BD']
            generator_data = data['GD']

    return branch_data, busbar_data, generator_data

def loadcase_load(casefile):

    info = 0

    if isinstance(casefile, basestring):
        if casefile.endswith('.py'):
            rootname, extension = splitext(casefile)
        else:
            rootname = casefile
            if exists(casefile + '.py'):
                extension = '.py'
            else:
                info = 2
                fname = basename(rootname)

    if info == 0:
        if extension == '.py':
            exec(compile(open(rootname + extension).read(),
                            rootname + extension, 'exec'))

            data = locals()
            branch_data = data['LD']
            busbar_data = data['BD']
            generator_data = data['GD']
            plmd_data = data['PLMD']

    return branch_data, busbar_data, generator_data, plmd_data

```



Figura A4.4. Código de la función loadcase.py.

Este archivo permite cargar un determinado caso de estudio, es decir permite la lectura de datos de las matrices de entrada.

- **ext2int.py:**

```

"""
Este archivo contiene las funciones ext_to_int y ext_to_int_load
usadas dentro de las funciones runpqpf y runloadmodelingpf
para considerar los índices de los datos ingresados desde cero
y poder trabajar dentro de Python
"""
from sys_var_idx import *

def ext_to_int(br_matrix,bbd_matrix,gend_matrix):

    nbb = bbd_matrix.shape[0]
    j = 0
    for i in range(nbb):
        change = bbd_matrix[i,BUS_ID]
        bbd_matrix[i,BUS_ID] = i
        for j in range(br_matrix.shape[0]):
            if br_matrix[j,FR_B] == change:
                br_matrix[j,FR_B] = i
            if br_matrix[j,TO_B] == change:
                br_matrix[j,TO_B] = i
        for j in range(gend_matrix.shape[0]):
            if gend_matrix[j,GEN_B] == change:
                gend_matrix[j,GEN_B] = i

    return

def ext_to_int_load(br_matrix,bbd_matrix,gend_matrix,plmd_matrix):

    nbb = bbd_matrix.shape[0]
    j = 0
    for i in range(nbb):
        change = bbd_matrix[i,BUS_ID]
        bbd_matrix[i,BUS_ID] = i
        for j in range(br_matrix.shape[0]):
            if br_matrix[j,FR_B] == change:
                br_matrix[j,FR_B] = i
            if br_matrix[j,TO_B] == change:
                br_matrix[j,TO_B] = i
        for j in range(gend_matrix.shape[0]):
            if gend_matrix[j,GEN_B] == change:
                gend_matrix[j,GEN_B] = i
        for j in range(plmd_matrix.shape[0]):
            if plmd_matrix[j,BUSL_ID] == change:
                plmd_matrix[j,BUSL_ID] = i
            if plmd_matrix[j,BUSL_TP] == change:
                plmd_matrix[j,BUSL_TP] = i
            if plmd_matrix[j,UZ] == change:
                plmd_matrix[j,UZ] = i
            if plmd_matrix[j,UI] == change:
                plmd_matrix[j,UI] = i
            if plmd_matrix[j,UP] == change:
                plmd_matrix[j,UP] = i
            if plmd_matrix[j,WZ] == change:
                plmd_matrix[j,WZ] = i
            if plmd_matrix[j,WI] == change:
                plmd_matrix[j,WI] = i
            if plmd_matrix[j,WP] == change:
                plmd_matrix[j,WP] = i

    return

```



Figura A4.5. Código de la función ext2int.py.

Esta función permite convertir los índices exteriores contabilizados desde 1 a índices utilizados internamente por Python contados desde 0, es decir permite contabilizar los índices desde cero para trabajar sin problemas en Python.

- **data_conv.py:**

Esta función permite subdividir las matrices de datos de entrada, toma los valores de todas las filas de una misma columna y crea una matriz n*1, permitiendo de esta manera separar los datos y operar con ellos dentro del código desarrollado en Python.

El código implementado para esta función se muestra en a Figura A4.6.

```

"""
Este archivo contiene las funciones para subdividir las columnas
de los datos ingresados en submatrices.
Estas funciones se usan dentro de las funciones runppqf y runloadmodelingpf
"""
from numpy import shape, zeros, transpose, pi
from sys_var_idx import *
from nrpf_settings import load_fact

def brd_conv(br_matrix):

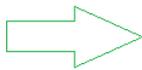
    frombus = br_matrix[:, FR_B]
    tobus = br_matrix[:, TO_B]
    tlresis = br_matrix[:, BR_R]
    tlreact = br_matrix[:, BR_X]
    tlsuscp = br_matrix[:, BR_S]
    tlstat = br_matrix[:, BR_ST]
    trtap = br_matrix[:, TAP_H]
    return frombus, tobus, tlresis, tlreact, tlsuscp, tlstat, trtap

def bbd_conv(bbd_matrix):

    nbb = bbd_matrix.shape[0]
    busnum = bbd_matrix[:, BUS_ID]
    bustype = bbd_matrix[:, BUS_TP]
    vm = bbd_matrix[:, VoM]
    va = bbd_matrix[:, VoLA]*pi/180
    pload = bbd_matrix[:, PL]*load_fact
    qload = bbd_matrix[:, QL]*load_fact
    bshunt = bbd_matrix[:, SH_B]

    return nbb, bustype, vm, va, pload, qload, bshunt, busnum

```



CONTINUACIÓN

```

def gend_conv(gend_matrix, nbb, bustype):

    ngn = gend_matrix.shape[0]
    pgen = zeros((nbb,1))
    qgen = zeros((nbb,1))
    genbus = gend_matrix[:, GEN_B]
    qmax = gend_matrix[:, QMAX]
    qmin = gend_matrix[:, QMIN]
    genstat = gend_matrix[:, GEN_ST]
    j = 0

    for i in gend_matrix[:, GEN_B]:
        i = int(i)
        if gend_matrix[j, GEN_ST] != 0:
            pgen[i,0] = gend_matrix[j, PG]
            qgen[i,0] = gend_matrix[j, QG]

        else:
            bustype[i,0] = PQ
            pgen[i,0] = 0
            qgen[i,0] = 0

        j+=1

    return ngn, pgen, qgen, genbus, qmax, qmin, genstat, bustype

def plmd_conv(plmd_matrix, nbb):

    btypeload = zeros((nbb,1))
    uz = zeros((nbb,1))
    ui = zeros((nbb,1))
    up = zeros((nbb,1))
    wz = zeros((nbb,1))
    wi = zeros((nbb,1))
    wp = zeros((nbb,1))
    j=0
    for i in plmd_matrix[:, BUSL_ID]:
        i = int(i)

        uz[i,0] = plmd_matrix[j, UZ]
        ui[i,0] = plmd_matrix[j, UI]
        up[i,0] = plmd_matrix[j, UP]
        wz[i,0] = plmd_matrix[j, WZ]
        wi[i,0] = plmd_matrix[j, WI]
        wp[i,0] = plmd_matrix[j, WP]
        btypeload[i,0] = plmd_matrix[j, BUSL_TP]

        j+=1

    return uz, ui, up, wz, wi, wp, btypeload

```

Figura A4.6. Código de la función data_convpy.

- nrpf_settings.py:

```

"""
Este archivo contiene los parámetros a tomarse en cuenta
dentro de la solución del flujo de potencia
"""
global itmax, tol, qgen_lim, load_fact

itmax = 50
tol = 10e-10

qgen_lim = 1
load_fact = 1.0

```

Figura A4.7. Código de la función nrpf_settings.py.

Esta función contiene los parámetros con cuales se resuelve el cálculo del flujo de potencia mediante Newton_Raphson.

- **Ybus_form.py:**

```

"""
Este archivo contiene la función Ybus usada dentro de las funciones runpqpf
y runloadmodelingpf para calcular la matriz de admitancias Y de barra
"""

from sys_var_idx import *
from numpy import ones, conj, r_, array, nonzero
from scipy.sparse import csr_matrix

def Ybus(branch,bus):

    # transformación a matrices tipo array
    branch = array(branch) #transforma a valores numéricos
    bus = array(bus)

    nbb = bus.shape[0] #numero de barras
    nbr = branch.shape[0] # numero de branches

    stat = branch[:,BR_ST] #si las líneas están o no operativas
    Ys = stat / (branch[:,BR_R] + 1j * branch[:,BR_X]) #admitancia
    Bc = stat * branch[:,BR_S] #suceptancias paralelo
    tap = ones(nbr)
    i = nonzero(branch[:,TAP_H])
    tap[i] = branch[i,TAP_H] #valores de los taps

    Ytt = Ys + 1j * Bc / 2
    Yff = Ytt / (tap * conj(tap))
    Yft = - Ys / conj(tap)
    Ytf = - Ys / tap

    Ysh = (1j * bus[:, SH_B]) #compensaciones shunt

    f = branch[:, FR_B] #contiene los índices de salida
    t = branch[:, TO_B] #contiene los índices de llegada

    Cf = csr_matrix((ones(nbr), (range(nbr),f)), (nbr,nbb))
    Ct = csr_matrix((ones(nbr), (range(nbr),t)), (nbr,nbb))

    i = r_[range(nbr), range(nbr)]

    Yf = csr_matrix((r_[Yff, Yft], (i, r_[f, t])), (nbr,nbb))
    Yt = csr_matrix((r_[Ytf, Ytt], (i, r_[f, t])), (nbr,nbb))

    Ybus = Cf.T * Yf + Ct.T * Yt + \
        csr_matrix((Ysh, (range(nbb), range(nbb))), (nbb,nbb))

    col = 0; row = 0
    C = []
    for row in range(nbb):
        i = 0
        for col in range(nbb):
            if Ybus[row,col] != 0:
                C.append(1)
            else:
                C.append(0)
            i = i + 1
    C = array(C).reshape(nbb,nbb)
    return Ybus,C

```



Figura A4.8. Código de la función Ybus_form.py.

Esta función calcula la matriz de admitancias de barra del sistema. Retorna la matriz de admitancias de barra y una matriz C que indica las ramas que tienen conexión.

- **nrpf_core.py:**

Esta función contiene el algoritmo de cálculo de la metodología de Newton - Raphson para la solución del flujo de potencia.

El código desarrollado se muestra en la Figura A4.9:

```

"""
Este archivo contiene las funciones runpf y runpf_load
usadas dentro de las funciones runppf y runloadmodelingpf
para realizar el método de Newton Raphson
"""

# Librerías de Python
from numpy import pi,alltrue,savetxt
from scipy.sparse.linalg import spsolve
from scipy.sparse import csr_matrix,csc_matrix
from time import time
# Funciones creadas en Python
from nrpf_func import *
from nrpf_jacb import jacobian_form
from nrpf_svud import sv_updating
from nrpf_settings import *

def runpf(br_data,bb_data,gn_data,Ybus,C):
    YR = Ybus.real
    YI = Ybus.imag
    [frombus,tobus,tlresis,tlreact,tluscp,brstat,trtap] = br_data
    [nbb,bustype,vm,va,Pload,Qload,bsh,busn] = bb_data
    [ngn,Pgen,Qgen,genbus,Qmax,Qmin,GenSTAT,bustype] = gn_data

    check_gen = 0
    bus_pointer = 0
    convergence = False
    it = 1

    #*****
    #Cálculo de potencias especificadas
    #*****

    Pnet,Qnet = net_powers(Pgen,Qgen,Pload,Qload)

    while not convergence:
        #*****
        #Cálculo de potencias calculadas
        #*****

        Pcal,Qcal = cal_powers(vm,va,YR,YI,nbb,C)

        #*****
        #Cálculo de variaciones de potencia
        #*****

        time_ref6 = time()

        nmax = jacobian_dim(nbb)

        PQmismatch,PM_pointer = power_mism(nmax,Pcal,Qcal,Pnet,Qnet,bustype,nbb)

    # *****
    #Comprobación de la convergencia
    # *****
    max_error = max(abs(PQmismatch))

    if alltrue(abs(PQmismatch)<= tol):
        convergence = True
        break
    elif it >= itmax:
        break

    it += 1

    #*****
    #Cálculo del jacobiano
    #*****
    J,J_pointer = jacobian_form(nmax,vm,va,YR,YI,Pcal,Qcal,bustype,nbb,C)
    J = csr_matrix(J)
    D = spsolve(J,PQmismatch)

    # *****
    #Actualización de valores
    # *****

    vm,va,Dpointer = sv_updating(vm,va,D,nbb)

    # *****
    #Cálculo de la potencia generada
    # *****
    Pg = Pcal + Pload
    Qg = Qcal + Qload

    general_res = (it,max_error)
    return vm,va,Pg,Qg,it

def runpf_load(br_data,bb_data,gn_data,plmd_data,Ybus,C):
    YR = Ybus.real
    YI = Ybus.imag
    [frombus,tobus,tlresis,tlreact,tluscp,brstat,trtap] = br_data
    [nbb,bustype,vm,va,Pload,Qload,bsh,busn] = bb_data
    [ngn,Pgen,Qgen,genbus,Qmax,Qmin,GenSTAT,bustype] = gn_data
    [uz,ui,up,wz,wi,wp,btypeload]=plmd_data
    check_gen = 0
    bus_pointer = 0
    convergence = False
    it = 1

    while not convergence:
        #*****
        #Cálculo de potencias esperadas
        #*****

        Fnet,Qnet = net_powers_load(Pgen,Qgen,Pload,Qload,vm,nbb,C,uz,ui,up,wz,wi,wp,btypeload)

        #*****
        #Cálculo de potencias calculadas
        #*****

        Fcal,Qcal = cal_powers(vm,va,YR,YI,nbb,C)

        #*****
        #Cálculo de variaciones de potencia
        #*****

        time_ref6 = time()

        rmax = jacobian_dim(nbb)

        FQmismatch,PM_pointer = power_mism(nmax,Fcal,Qcal,Fnet,Qnet,bustype,nbb)

    # *****
    #Comprobación de la convergencia
    # *****
    max_error = max(abs(FQmismatch))

    if alltrue(abs(FQmismatch)<= tcl):
        convergence = True
        break
    elif it >= itmax:
        break

    it += 1

    #*****
    #Cálculo del jacobiano
    #*****
    J,J_pointer = jacobian_form(nmax,vm,va,YR,YI,Pcal,Qcal,bustype,nbb,C)
    J = csr_matrix(J)
    D = spsolve(J,FQmismatch)

    # *****
    #Actualización de valores
    # *****
    vm,va,Dpointer = sv_updating(vm,va,D,nbb)

    #*****
    #Cálculo de la potencia generada
    # *****
    Pg=zeros((nbb,1))
    Qg=zeros((nbb,1))
    for i in range(nbb):
        if int(bustype[i,0])==2:
            Pg[i,0] = abs(Pcal[i,0] + Fload[i,0])
            Qg[i,0] = Qcal[i,0] + Qload[i,0] - ((vm[i,0])**2*bsh[i,0])
        else:
            Pg[i,0] = abs(Pcal[i,0] - Fnet[i,0])
            Qg[i,0] = Qcal[i,0] - Qnet[i,0] - ((vm[i,0])**2*bsh[i,0])
    general_res = (it,max_error)
    return vm,va,Pg,Qg,it,Pnet,Qnet

```

Figura A4.9. Código de la función nrpf_core.py.

- **nrpf_func.py:**

Esta función contiene el desarrollo matemático de las funciones utilizadas en nrpf_core para el cálculo del flujo de potencia.

```

"""
Este archivo contiene las funciones usadas dentro del archivo nrpf_core
para resolver el método de Newton Raphson
"""
from numpy import zeros,cos,sin
from sys_var_idx import SL,PQ,PV
def net_powers(pgen,qgen,pload,qload):
    p_sch = pgen - pload
    q_sch = qgen - qload
    return p_sch,q_sch

def cal_powers(vm,va,G,B,nbb,C):

    pcal = zeros((nbb,1))
    qcal = zeros((nbb,1))
    for i in range(nbb):
        vmagi = vm[i,0]
        vangj = va[i,0]

        for k in range(nbb):
            if C[i,k] == 1:
                if i == k:
                    pcal[i,0] += vmagi**2*G[i,i]
                    qcal[i,0] += -vmagi**2*B[i,i]
                else:
                    vmagj = vm[k,0]
                    vangj = va[k,0]
                    term1 = vmagi*vmagj
                    angle = vangj - vangj
                    pcal[i] += term1*(G[i,k]*cos(angle)+B[i,k]*sin(angle))
                    qcal[i] += term1*(G[i,k]*sin(angle)-B[i,k]*cos(angle))

    return pcal,qcal

def jacobian_dim(nbb):
    nmax = 2*nbb
    return nmax

def power_mism(nmax,pcal,qcal,psch,qsch,bustype,nbb):
    Pmism = psch - pcal
    Qmism = qsch - qcal

    for i in range(nbb):
        if bustype[i] == SL:
            Pmism[i] = 0
            Qmism[i] = 0
        elif bustype[i] == PV:
            Qmism[i] = 0

    PQmismatch = zeros(nmax)
    k = 0
    for i in range(nbb):
        PQmismatch[k] = Pmism[i]
        PQmismatch[k+1] = Qmism[i]
        k = k + 2

    pm_pointer = 2*nbb
    return PQmismatch,pm_pointer

def net_powers_load (pgen,qgen,pload,qload,vm,nbb,C,uz,ui,up,wz,wi,wp,btypeload):

    p_o =pgen - pload
    q_o = qgen - qload
    p_sch = zeros((nbb,1))
    q_sch = zeros((nbb,1))

    for i in range(nbb):
        btypeloadi=btypeload[i,0]
        vmagi=vm[i,0]
        poi=p_o[i,0]
        qoi=q_o[i,0]
        uzi=uz[i,0]
        uii=ui[i,0]
        upi=up[i,0]
        wzi=wz[i,0]
        wii=wi[i,0]
        wpi=wp[i,0]
        for k in range (nbb):
            if C[i,k]==1:
                if i==k:
                    if btypeloadi==0:
                        p_sch[i,0]=poi
                        q_sch[i,0]=qoi
                    if btypeloadi==1:
                        vo=vmagi
                        p_sch[i,0]=poi
                        q_sch[i,0]=qoi*(wzi*((vmagi/vo)**2)+wii*(vmagi/vo)+wpi)
                    else:
                        vo=1
                        p_sch[i,0]=poi*(uzi*((vmagi/vo)**2)+uii*(vmagi/vo)+upi)
                        q_sch[i,0]=qoi*(wzi*((vmagi/vo)**2)+wii*(vmagi/vo)+wpi)

    return p_sch,q_sch

```

Figura A4.10. Código de la función nrpf_func.py.

- nrpf_jacb.py:

Esta función calcula el Jacobiano del sistema eléctrico de potencia que se esté simulando.

```

"""
Este archivo forma el jacobiano
"""
from numpy import zeros
from math import sin,cos
from sys_var_idx import SL,PV,PQ

def jacobian_form(nmax,vm,va,G,B,pcal,qcal,bustype,nbb,C):
    J = zeros((nmax,nmax))
    row = 0
    for i in range(nbb):
        vmagi = vm[i,0]
        vangj = va[i,0]
        col = 0
        for j in range(nbb):
            vmagj = vm[j,0]
            vangj = va[j,0]
            if i == j:
                J[row ,col ] = -qcal[i,0] - vmagi**2*B[i,i]
                J[row ,col+1] = pcal[i,0] + vmagi**2*G[i,i]
                J[row+1,col ] = pcal[i,0] - vmagi**2*G[i,i]
                J[row+1,col+1] = qcal[i,0] - vmagi**2*B[i,i]
            else:
                if C[i,j] == 1:
                    angle = vangj-vangi
                    J[row ,col ] = vmagi*vmagj*(G[i,j]*sin(angle)-B[i,j]*cos(angle))
                    J[row+1,col ] = -vmagi*vmagj*(G[i,j]*cos(angle)+B[i,j]*sin(angle))
                    J[row ,col+1] = -J[row+1,col ]
                    J[row+1,col+1] = J[row ,col ]
                col += 2
            row += 2
        row = 0; col = 0
    for i in range(nbb):
        if bustype[i] != PQ:
            J[:,col+1] = 0
            J[row+1,:] = 0
            J[row+1,col+1] = 1
        if bustype[i] == SL:
            J[:,col] = 0
            J[row,:] = 0
            J[row,col] = 1
        row += 2
        col += 2
    j_pointer = 2*nbb
    return J,j_pointer

```

Figura A4.11. Código de la función nrpf_jacb.py.

- nrpf_svud.py:

Esta función actualiza las variables de estado.

```

"""
Este archivo contiene la función que actualiza los valores
dentro del proceso iterativo de Newton Raphson
"""
def sv_updating(vm,va,D,nbb):
    j = 0
    for i in range(nbb):
        va[i] += D[j]
        vm[i] += D[j+1]
        j += 2
    dpointer = 2*nbb
    return vm,va, dpointer

```

Figura A4.12. Código de la función nrpf_svud.py.

- **nrpf_branchpf.py:**

Una vez determinados los valores de las variables de estado del sistema, esta función permite calcular el valor de las potencias activas y reactivas de envío, recepción y pérdidas de todas las ramas del sistema.

```

"""
Este archivo contiene la función branch_pqflow, la cual calcula
la potencia de todas las barras del SEP.
"""
from numpy import sin,cos

def branch_pqflow(vm,va,frombus,tobus,branchR,branchX,branchB,branchSTAT,branchTAP):
    Ssend = []
    Srecv = []
    Sloss = []
    nbr = frombus.shape[0]

    for i in range(nbr):
        send = (frombus[i,0],0)
        recv = (tobus[i,0],0)

        Vsend = vm[send]*(cos(va[send])+1j*sin(va[send]))
        Vrecv = vm[recv]*(cos(va[recv])+1j*sin(va[recv]))

        if branchTAP[i,0] != 0:
            y = 1/(branchTAP[i,0]**2*(branchR[i,0] + 1j*branchX[i,0]))
            Y = (branchTAP[i,0])*y
            Ysend = (1-branchTAP[i,0])*y
            Yrec = branchTAP[i,0]*(branchTAP[i,0]-1)*y

            Isend = (Vsend - Vrecv)*Y + Vsend*Ysend
            S = Vsend*Isend.conjugate()
            Ssend.append(S)

            Irecv = (Vrecv - Vsend)*Y + Vrecv*Yrec
            S = Vrecv*Irecv.conjugate()
            Srecv.append(S)

            Slosses = Ssend[i] + Srecv[i]
            Sloss.append(Slosses)

        else:
            Z = branchR[i,0] + 1j*branchX[i,0]

            Isend = (Vsend - Vrecv)/Z + Vsend*(1j*branchB[i,0])*0.5
            S = Vsend*Isend.conjugate()
            Ssend.append(S)

            Irecv = (Vrecv - Vsend)/Z + Vrecv*(1j*branchB[i,0])*0.5
            S = Vrecv*Irecv.conjugate()
            Srecv.append(S)

            Slosses = Ssend[i] + Srecv[i]
            Sloss.append(Slosses)

    return nbr,Ssend,Srecv,Sloss

```

Figura A4.13. Código de la función nrpf_branchpf.py.

- **nrpf_printpf.py:**

Esta función permite imprimir los resultados del flujo de potencia con y sin la incorporación de modelos de carga, permite visualizar en forma ordenada los valores obtenidos luego de ejecutar el cálculo del flujo de potencia.

El código implementado se muestra en las Figuras A4.14 y A4.15.

- runpf.py:

Esta función permite ejecutar el cálculo del flujo de potencia, para ello va llamando a las funciones explicadas anteriormente. El código implementado se muestra en las Figuras A4.16 y A4.17.

```

"""
Este archivo ejecuta el flujo de potencia con y sin
la incorporación de modelos de carga.
Este archivo utiliza la metodología de inyección nodal
para el modelo de carga polinomial
"""

# Importación de Funciones propias de Python
import numpy #Módulo numérico de Python
import scipy #contiene el paquete de funciones de Python
from sys import stdout,version_info #Módulo de interpretación de Python
from os.path import dirname,join #I. retorna el directorio 2.une dos o mas enlçes
from time import time #para obtener el tiempo de simulación
# Importación de Funciones creadas en Python
from data_conv import *
from info import *
from Ybus_form import *
from nrpf_core import *
from ext2int import *
from nrpf_printpf import *
from loadcase import *
from nrpf_settings import *

def runqpf(casedata = None):
    """
    Ejecuta el flujo de potencia de Newton Raphson para el modelo de carga PQ
    """
    # Cargar el caso de estudio a simularse
    #*****

    if casedata is None:
        casedata = join(dirname(__file__),'Case14')
    #*****

    # Imprimir la versión del programa que se está utilizando
    #*****

    v = info('all')
    stdout.write('\nFLUJO DE POTENCIA SIN INCORPORACIÓN DE MODELOS DE CARGA \nVersion %s \n%s' % (v["Version"],v["Date"]))
    stdout.write('\nPython version: %s.%s.%s'%version_info[:3])
    stdout.write('\nRunning with Scipy Version: %s'%scipy.__version__)
    stdout.write(' and Numpy version %s'%numpy.__version__)

    #*****
    # Cargar los datos del caso de estudio
    #*****
    LD,BD,GD = loadcase(casedata)
    tt = time()

    #*****
    # Empezar a contar los índices desde cero
    #*****
    ext_to_int(LD,BD,GD)

    #*****
    # Divide en submatrices cada columna de los datos de entrada
    #*****
    br_data = brd_conv(LD)
    bb_data = bbd_conv(BD)
    gn_data = gend_conv(GD,bb_data[0],bb_data[1])

    #*****
    # Cálculo de la Y de barra
    #*****
    [Yb,C] = Ybus(LD,BD)

    #*****
    # Cálculo del Flujo de potencia mediante Newton Raphson
    #*****
    time_ref = time()
    stdout.write('\n\nStarting Power Flow Calculation Algorithm for Case File: %s \n'%(casedata))
    VM,VA,PGEN,QGEN,it = runpf(br_data,bb_data,gn_data,Yb,C)
    time_end = time()

    if it<itmax:
        stdout.write('\nNewton-Raphson Power Flow algorithm succesfully converged')
        stdout.write('\nin %i iterations in %.2f seconds'%(it,time_end-time_ref))
    else:
        no_convergece = True
        stdout.write('\nPower Mismatches could not be reduced to %.2e after %i iteration(s)'%(tol,it))

    printpf(VM,VA,PGEN,QGEN,it,br_data,bb_data,gn_data)
    tf = time()

    print('Total elapsed time: %.2f'%(tf-tt))
    return

```

Figura A4.16. Código de la función runpf.py.

Continuación del código de la función runpf.py

```

def runloadmodelingpf(casedata = None):
    """
    Ejecuta el flujo de potencia de Newton Raphson para el modelo de carga polinomial
    """
    #*****
    # Cargar el caso de estudio a simularse
    #*****

    if casedata is None:
        casedata = join(dirname(__file__), 'Case14')

    #*****
    # Imprimir la versión del programa que se está utilizando
    #*****

    v = info('all')
    stdout.write('\nFLUJO DE POTENCIA CON LA INCORPORACIÓN DE MODELOS DE CARGA \nVersion %s \n%s' % (v["Version"],v["Date"]))
    stdout.write('\nMETODOLOGÍA DE INYECCIÓN NODAL')
    stdout.write('\nMODELO DE CARGA POLINOMIAL')
    stdout.write('\nPython version: %s.%s.%s'%version_info[:3])
    stdout.write('\nRunning with Scipy Version: %s '%scipy.__version__)
    stdout.write(' and Numpy version %s'%numpy.__version__)

    #*****
    # Cargar los datos del caso de estudio
    #*****

    LD,BD,GD,PLMD = loadcase_load(casedata)

    tt = time()

    #*****
    # Empezar a contar los índices desde cero
    #*****

    ext_to_int_load(LD,BD,GD,PLMD)

    #*****
    # Divide en submatrices cada columna de los datos de entrada
    #*****

    br_data = brd_conv(LD)
    bb_data = bbd_conv(BD)
    gn_data = gend_conv(GD,bb_data[0],bb_data[1])
    plmd_data = plmd_conv(PLMD,bb_data[0])

    #*****
    # Cálculo de la Y de barra
    #*****
    [Yb,C] = Ybus(LD,BD)

    #*****
    # Cálculo del Flujo de potencia mediante Newton Raphson
    #*****
    time_ref = time()
    stdout.write('\n\nStarting Power Flow Calculation Algorithm for Case File: %s \n'%(casedata))
    VM,VA,PGEN,QGEN,it,Pnet,Qnet = runpf_load(br_data,bb_data,gn_data,plmd_data,Yb,C)
    time_end = time()

    if it<itmax:
        stdout.write('\nNewton-Raphson Power Flow algorithm succesfully converged')
        stdout.write('\nin %i iterations in %.2f seconds'%(it,time_end-time_ref))
    else:
        no_convergece = True
        stdout.write('\nPower Mismatches could not be reduced to %.2e after %i iteration(s)'%(tol,it))

    printpf_load(VM,VA,PGEN,QGEN,it,br_data,bb_data,gn_data,Pnet,Qnet)
    tf = time()

    print('Total elapsed time: %.2f'%(tf-tt))
    return

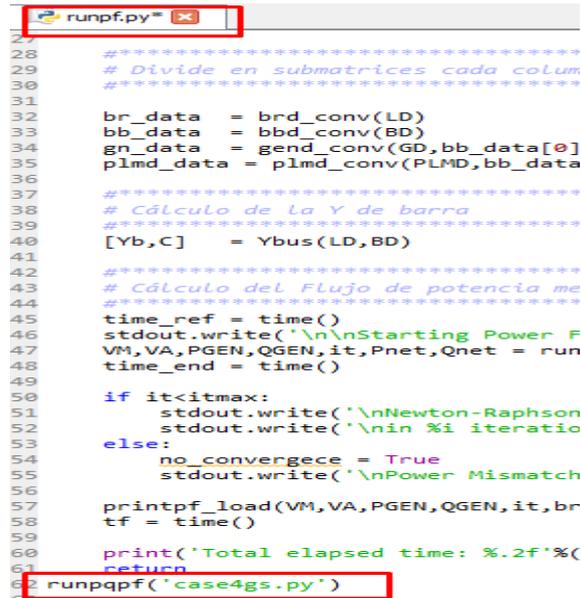
```

Figura A4.17. Código de la función runpf.py.

- Ejecución del Programa Desarrollado:

El programa puede ejecutarse de dos maneras:

1. Desde la función runpf.py escribiendo el comando *runpq* ('case_.py') si se desea ejecutar un flujo de potencia sin la incorporación de modelos de carga, o escribiendo el comando *runloadmodelingpf* ('case_.py') si se desea ejecutar un flujo de potencia considerando la incorporación de modelos de carga.

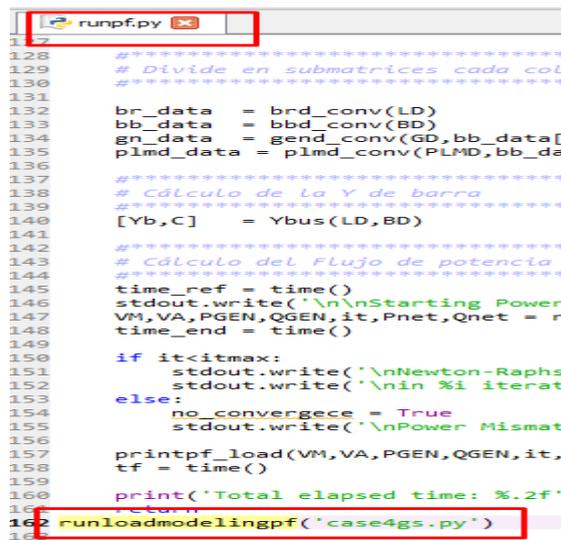


```

27
28 #*****
29 # Divide en submatrices cada colum
30 #*****
31
32 br_data = brd_conv(LD)
33 bb_data = bbd_conv(BD)
34 gn_data = gend_conv(GD,bb_data[0]
35 plmd_data = plmd_conv(PLMD,bb_data
36
37 #*****
38 # Cálculo de la Y de barra
39 #*****
40 [Yb,C] = Ybus(LD,BD)
41
42 #*****
43 # Cálculo del Flujo de potencia me
44 #*****
45 time_ref = time()
46 stdout.write('\n\nStarting Power F
47 VM,VA,PGEN,QGEN,it,Pnet,Qnet = run
48 time_end = time()
49
50 if it<itmax:
51     stdout.write('\nNewton-Raphson
52     stdout.write('\nin %i iteratio
53 else:
54     no_convergece = True
55     stdout.write('\nPower Mismatch
56
57 printpf_load(VM,VA,PGEN,QGEN,it,br
58 tf = time()
59
60 print('Total elapsed time: %.2f'%(
61     return
62 runpqpf('case4gs.py')

```

Figura A4.18. Ejecución de un flujo de potencia sin considerar la incorporación de modelos de carga desde el archivo runpf.py.



```

128 #*****
129 # Divide en submatrices cada col
130 #*****
131
132 br_data = brd_conv(LD)
133 bb_data = bbd_conv(BD)
134 gn_data = gend_conv(GD,bb_data[
135 plmd_data = plmd_conv(PLMD,bb_da
136
137 #*****
138 # Cálculo de la Y de barra
139 #*****
140 [Yb,C] = Ybus(LD,BD)
141
142 #*****
143 # Cálculo del Flujo de potencia
144 #*****
145 time_ref = time()
146 stdout.write('\n\nStarting Power
147 VM,VA,PGEN,QGEN,it,Pnet,Qnet = r
148 time_end = time()
149
150 if it<itmax:
151     stdout.write('\nNewton-Raphs
152     stdout.write('\nin %i iterat
153 else:
154     no_convergece = True
155     stdout.write('\nPower Mismatch
156
157 printpf_load(VM,VA,PGEN,QGEN,it,
158 tf = time()
159
160 print('Total elapsed time: %.2f'
161     return
162 runloadmodelingpf('case4gs.py')

```

Figura A4.19. Ejecución de un flujo de potencia sin considerar la incorporación de modelos de carga desde el archivo runpf.py.

Los resultados que se obtienen para el caso de 4 barras son los siguientes:

RESULTADOS SIN INCORPORAR MODELOS DE CARGA **RESULTADOS INCORPORANDO MODELOS DE CARGA**

```

FLUJO DE POTENCIA SIN INCORPORACIÓN DE MODELOS DE CARGA
Version 1.0.0
02-Mayo-2018
Python version: 3.5.2
Running with Scipy Version: 0.17.1 and Numpy version 1.11.1

Starting Power Flow Calculation Algorithm for Case File: case4gs.py

Newton-Raphson Power Flow algorithm successfully converged
in 8 iterations in 0.12 seconds
total iterations done: 8

=====
|                               SYSTEM SUMMARY                               |
=====
Network Elements  Power Summary  P[MW]  Q[Mvar]
-----
Buses:           4  Total Gen. Power:  504.81  295.93
Branches:        4  Total Load:         500.00  309.86
Generators:      2  Losses:             4.81   -13.93
Dispatched:     2  Shunt(inj):         0.00   0.00
Loads:           4
Shunts:          0

                Minimum                Maximum
-----
Voltage Magnitude:  0.969 p.u. @ bus 3  1.020 p.u. @ bus 4

=====
|                               BUS RESULTS                               |
=====
Bus #  Voltage Mag[p.u] Angle[Deg]  Generation P[MW] Q[MVar]  Load P[MW] Q[MVar]
-----
1  1.0000  0.0000  186.81  114.50  50.00  30.99
2  0.9824  -0.9761  0.00  0.00  170.00  105.35
3  0.9690  -1.8722  0.00  0.00  200.00  123.94
4  1.0200  1.5231  318.00  181.43  80.00  49.58

=====
|                               POWER FLOWS                               |
=====
From To  Psend  Qsend  Precv  Qrecv  Ploss  Qloss
# # [MW] [Mvar] [MW] [Mvar] [MW] [Mvar]
-----
1 2  38.692  22.298  -38.465  -31.236  0.227  -8.938
1 3  98.118  61.212  -97.086  -63.569  1.031  -2.356
2 4 -131.535 -74.114  133.251  74.920  1.715  0.806
3 4 -102.914 -60.371  104.749  56.930  1.835  -3.441

Total elapsed time: 0.13

```

```

FLUJO DE POTENCIA CON LA INCORPORACIÓN DE MODELOS DE CARGA
Version 1.0.0
02-Mayo-2018
METODOLOGÍA DE INYECCIÓN MODAL
MODELO DE CARGA POLINOMIAL
Python version: 3.5.2
Running with Scipy Version: 0.17.1 and Numpy version 1.11.1

Starting Power Flow Calculation Algorithm for Case File: case4gs.py

Newton-Raphson Power Flow algorithm successfully converged
in 9 iterations in 0.13 seconds
total iterations done: 9

=====
|                               SYSTEM SUMMARY                               |
=====
Network Elements  Power Summary  P[MW]  Q[Mvar]
-----
Buses:           4  Total Gen. Power:  497.47  289.56
Branches:        4  Total Load:         492.83  304.36
Generators:      2  Losses:             4.64   -14.80
Dispatched:     2  Shunt(inj):         0.00   0.00
Loads:           4
Shunts:          0

                Minimum                Maximum
-----
Voltage Magnitude:  0.970 p.u. @ bus 3  1.020 p.u. @ bus 4

=====
|                               BUS RESULTS                               |
=====
Bus #  Voltage Mag[p.u] Angle[Deg]  Generation P[MW] Q[MVar]  Load P[MW] Q[MVar]
-----
1  1.0000  0.0000  179.47  111.34  50.05  30.99
2  0.9828  -0.9110  0.00  -0.00  168.51  104.03
3  0.9704  -1.7695  0.00  -0.00  194.26  119.76
4  1.0200  1.6083  318.00  178.22  80.00  49.58

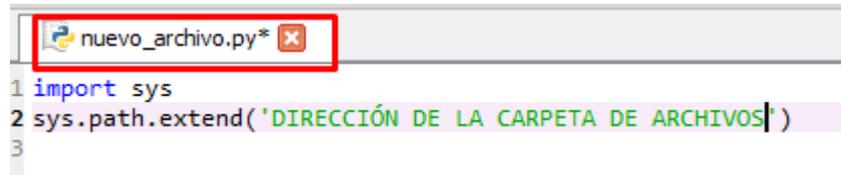
=====
|                               POWER FLOWS                               |
=====
From To  Psend  Qsend  Precv  Qrecv  Ploss  Qloss
# # [MW] [Mvar] [MW] [Mvar] [MW] [Mvar]
-----
1 2  36.426  21.997  -36.218  -31.032  0.208  -9.036
1 3  92.993  58.349  -92.061  -61.215  0.931  -2.867
2 4 -132.297 -73.002  134.014  73.816  1.718  0.814
3 4 -102.202 -58.543  103.986  54.826  1.784  -3.717

Total elapsed time: 0.14

```

Figura A4.20. Resultados del sistema de 4 barras con y sin la incorporación de modelos de carga

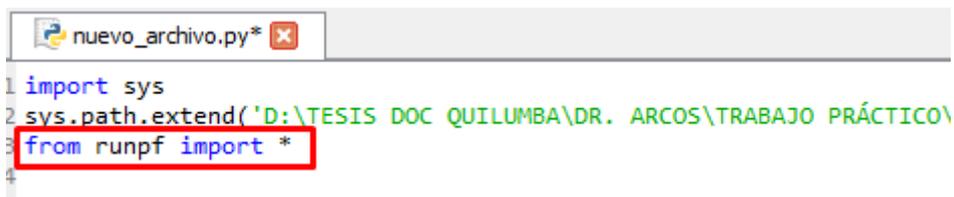
- Desde un nuevo archivo de Python, para lo cual se debe extender el directorio por defecto para que incluya la carpeta donde se encuentran los archivos del programa. La extensión del directorio se realiza mediante las siguientes instrucciones:



```
nuevo_archivo.py*
1 import sys
2 sys.path.extend('DIRECCIÓN DE LA CARPETA DE ARCHIVOS')
3
```

Figura A4.21. Instrucciones para extender el directorio en Python.

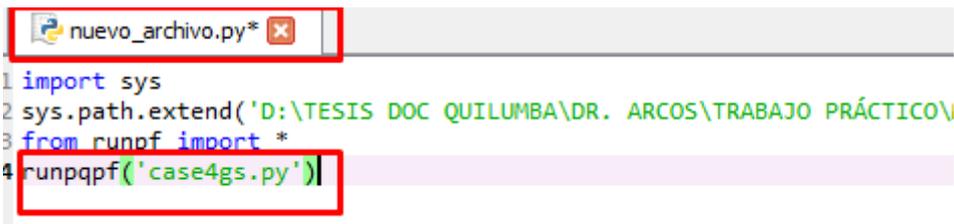
Luego de extender el directorio de Python, se importa la función `runpf` para poder ejecutar el programa. La instrucción que permite importar la función `runpf` es la siguiente:



```
nuevo_archivo.py*
1 import sys
2 sys.path.extend('D:\TESIS DOC QUILUMBA\DR. ARCOS\TRABAJO PRÁCTICO\
3 from runpf import *
4
```

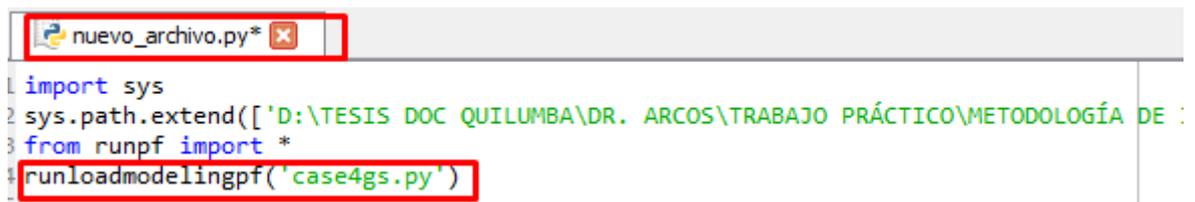
Figura A4.22. Instrucción para importar la función `runpf`.

Cuando ya se haya importado la función `runpf` se procede como en el caso anterior.



```
nuevo_archivo.py*
1 import sys
2 sys.path.extend('D:\TESIS DOC QUILUMBA\DR. ARCOS\TRABAJO PRÁCTICO\
3 from runpf import *
4 runqpf('case4gs.py')
```

Figura A4.23. Ejecución de un flujo de potencia sin considerar la incorporación de modelos de carga desde un `nuevo_archivo.py`.



```
nuevo_archivo.py*
1 import sys
2 sys.path.extend(['D:\TESIS DOC QUILUMBA\DR. ARCOS\TRABAJO PRÁCTICO\METODOLOGÍA DE :
3 from runpf import *
4 runloadmodelingpf('case4gs.py')
```

Figura A4.23. Ejecución de un flujo de potencia sin considerar la incorporación de modelos de carga desde un `nuevo_archivo.py`.

Los resultados que se obtienen para el caso de 4 barras son los siguientes:

RESULTADOS SIN INCORPORAR MODELOS DE CARGA

RESULTADOS INCORPORANDO MODELOS DE CARGA

```

FLUJO DE POTENCIA SIN INCORPORACIÓN DE MODELOS DE CARGA
Version 1.0.0
02-Mayo-2018
Python version: 3.5.2
Running with Scipy Version: 0.17.1 and Numpy version 1.11.1

Starting Power Flow Calculation Algorithm for Case File: case4gs.py

Newton-Raphson Power Flow algorithm succesfully converged
in 8 iterations in 0.12 seconds
total iterations done: 8
=====
|                               SYSTEM SUMMARY                               |
=====
Network Elements   Power Summary   P[MW]   Q[Mvar]
-----
Buses:            4   Total Gen. Power: 504.81  295.93
Branches:         4   Total Load:      500.00  309.86
Generators:       2   Losses:          4.81   -13.93
Dispatched:      2   Shunt(inj):      0.00   0.00
Loads:            4
Shunts:           0

                        Minimum           Maximum
-----
Voltage Magnitude:    0.969 p.u. @ bus 3   1.020 p.u. @ bus 4

=====
|                               BUS RESULTS                               |
=====
Bus      Voltage      Generation      Load
#      Mag[p.u] Angle[Deg] P[MW] Q[MVar] P[MW] Q[MVar]
-----
1      1.0000   0.0000  186.81 114.50   50.00  30.99
2      0.9824  -0.9761   0.00  0.00   170.00 105.35
3      0.9690  -1.8722   0.00  0.00   200.00 123.94
4      1.0200   1.5231  318.00 181.43   80.00  49.58

=====
|                               POWER FLOWS                               |
=====
From To Psend Qsend Precv Qrecv Ploss Qloss
# # [MW] [Mvar] [MW] [Mvar] [MW] [Mvar]
-----
1 2 38.692 22.298 -38.465 -31.236 0.227 -8.938
1 3 98.118 61.212 -97.086 -63.569 1.031 -2.356
2 4 -131.535 -74.114 133.251 74.920 1.715 0.806
3 4 -102.914 -60.371 104.749 56.930 1.835 -3.441

Total elapsed time: 0.13

```

```

FLUJO DE POTENCIA CON LA INCORPORACIÓN DE MODELOS DE CARGA
Version 1.0.0
02-Mayo-2018
METODOLOGÍA DE INYECCIÓN NODAL
MODELO DE CARGA POLINOMIAL
Python version: 3.5.2
Running with Scipy Version: 0.17.1 and Numpy version 1.11.1

Starting Power Flow Calculation Algorithm for Case File: case4gs.py

Newton-Raphson Power Flow algorithm succesfully converged
in 9 iterations in 0.13 seconds
total iterations done: 9
=====
|                               SYSTEM SUMMARY                               |
=====
Network Elements   Power Summary   P[MW]   Q[Mvar]
-----
Buses:            4   Total Gen. Power: 497.47  289.56
Branches:         4   Total Load:      492.83  304.36
Generators:       2   Losses:          4.64   -14.80
Dispatched:      2   Shunt(inj):      0.00   0.00
Loads:            4
Shunts:           0

                        Minimum           Maximum
-----
Voltage Magnitude:    0.970 p.u. @ bus 3   1.020 p.u. @ bus 4

=====
|                               BUS RESULTS                               |
=====
Bus      Voltage      Generation      Load
#      Mag[p.u] Angle[Deg] P[MW] Q[MVar] P[MW] Q[MVar]
-----
1      1.0000   0.0000  179.47 111.34   50.05  30.99
2      0.9828  -0.9110   0.00 -0.00   168.51 104.03
3      0.9704  -1.7695   0.00 -0.00   194.26 119.76
4      1.0200   1.6083  318.00 178.22   80.00  49.58

=====
|                               POWER FLOWS                               |
=====
From To Psend Qsend Precv Qrecv Ploss Qloss
# # [MW] [Mvar] [MW] [Mvar] [MW] [Mvar]
-----
1 2 36.426 21.997 -36.218 -31.032 0.208 -9.036
1 3 92.993 58.349 -92.061 -61.215 0.931 -2.867
2 4 -132.297 -73.002 134.014 73.816 1.718 0.814
3 4 -102.202 -58.543 103.986 54.826 1.784 -3.717

Total elapsed time: 0.14

```

Figura A4.24. Resultados del sistema de 4 barras con y sin la incorporación de modelos de carga

ANEXO V. Archivos Desarrollados en Excel para Agrupar Cargas

- Metodología para agrupar cargas:

ESCUELA POLITÉCNICA NACIONAL													
TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE ING. ELÉCTRICO													
INCORPORACIÓN DE MODELOS DE CARGA EN ESTADO ESTABLE PARA EL CÁLCULO DE FLUJOS DE POTENCIA													
METODOLOGÍA PARA AGRUPAR CARGAS													
Autor: Cristian Javier Jiménez Salazar													
DATOS DE ENTRADA													
CARGAS BARRA 1	MODELO POLINOMIAL						MODELO EXPONENCIAL		MODELO LINEAL				
	POTENCIA ACTIVA			POTENCIA REACTIVA			kp	kq	POTENCIA ACTIVA		POTENCIA REACTIVA		
	Zp	Ip	Pp	Zq	Iq	Pq			a0	a1	b0	b1	
Foco Ahorrador	0,144	0,723	0,133	0,050	0,941	0,009	0,898	1,021	0,277	0,723	0,059	0,941	
Celular	-2,121	3,329	-0,207	-1,819	3,045	-0,226	0,399	0,497	-2,329	3,329	-2,045	3,045	
DVD	0,294	-0,405	1,111	0,157	0,055	0,788	0,005	0,196	1,405	-0,405	0,945	0,055	
Equipo de sonido	-2,747	6,137	-2,390	-2,361	5,579	-2,217	1,286	1,442	-5,137	6,137	-4,579	5,579	
Laptop	0,824	-1,465	1,641	0,829	-1,249	1,419	-0,167	-0,026	2,465	-1,465	2,249	-1,249	
TV LED	1,257	-2,254	1,997	0,843	-1,603	1,761	-0,204	-0,201	3,254	-2,254	2,603	-1,603	
Microondas	-1,903	5,088	-2,185	-1,129	3,855	-1,727	2,570	2,674	-4,088	5,088	-2,855	3,855	

CÁLCULOS Y DESARROLLO													
SUPOSICIÓN DE % DE CARGA				COMPOSICIÓN DE CARGA EN %									
N° BARRA	COMERCIAL	INDUSTRIAL	RESIDENCIAL	CARGA	COMERCIAL	INDUSTRIAL	RESIDENCIAL						
BARRA 1	47	18	35	Foco Ahorrador	50	50	10						
				Celular	5	20	40						
				DVD	5	0	10						
				Equipo de sonido	15	25	10						
				Laptop	10	0	10						
				TV LED	10	0	10						
				Microondas	5	5	10						

CARGA	Yj	Zp	Ip	Pp	Zq	Iq	Pq	kp	kq	a0	a1	b0	b1
Foco Ahorrador	36	5,200	26,018	4,782	1,801	33,860	0,339	32,332	36,752	9,982	26,018	2,140	33,860
Celular	19,95	-42,322	66,408	-4,136	-36,291	60,753	-4,512	7,954	9,911	-46,458	66,408	-40,803	60,753
DVD	5,85	1,721	-2,368	6,497	0,919	0,320	4,610	0,029	1,146	8,218	-2,368	5,530	0,320
Equipo de sonido	15,05	-41,337	92,356	-35,968	-35,540	83,962	-33,372	19,351	21,698	-77,305	92,356	-68,912	83,962
Laptop	8,2	6,756	-12,013	13,457	6,800	-10,238	11,638	-1,368	-0,212	20,213	-12,013	18,438	-10,238
TV LED	8,2	10,305	-18,480	16,375	6,911	-13,148	14,436	-1,671	-1,645	26,680	-18,480	21,348	-13,148
Microondas	6,75	-12,842	34,344	-14,751	-7,620	26,024	-11,655	17,347	18,051	-27,594	34,344	-19,274	26,024

RESULTADOS													
CÁLCULO DE LOS PARÁMETROS A INGRESARSE EN LA SIMULACIÓN													
Zp	Ip	Pp	Zq	Iq	Pq	kp	kq	a0	a1	b0	b1		
-0,725	1,863	-0,137	-0,630	1,815	-0,185	0,740	0,857	-0,863	1,863	-0,815	1,815		
COMPROBACIÓN		SUMA DE P	1,000	SUMA DE Q	1,000	SUMA DE P		1,000	SUMA DE Q	1,000			

Figura A5.1. Metodología para agrupar cargas desarrollada en Excel.

ORDEN DE EMPASTADO



ESCUELA POLITÉCNICA NACIONAL
"CAMPUS POLITÉCNICO JOSÉ ALBÍN DARLAMA BACARTE"

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ORDEN DE EMPASTADO

De acuerdo con lo estipulado en el Art. 27 del Instructivo para la Implementación de la Unidad de Titulación en las Carreras y Programas Vigentes de la Escuela Politécnica Nacional, aprobado por Consejo Politécnico en sesión extraordinaria del 29 de abril de 2015 y por delegación del Decano, una vez verificado el cumplimiento de formato de presentación establecido, se autoriza la Impresión y encuadernación final del Trabajo de Titulación presentado por:

CRISTIAN JAVIER JIMÉNEZ SALAZAR

Fecha de autorización: 19 de julio de 2018



PLATE