

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Grado

**CONTROL DE LOS FLUJOS DE POTENCIA
EN REDES ELÉCTRICAS DE ALTA
TENSIÓN MEDIANTE EL USO DE
REACTANCIAS DE LINEA**

**(High voltage electrical network power flow
control through the use of line reactances)**

Para acceder al Título de

**GRADUADO EN INGENIERÍA EN
TECNOLOGÍAS INDUSTRIALES**

Autor: Víctor Terán Castillo

Septiembre - 2019

CONTROL DE LOS FLUJOS DE POTENCIA EN REDES ELÉCTRICAS DE ALTA TENSIÓN MEDIANTE EL USO DE REACTANCIAS DE LINEA

Índice de contenido

1	INTRODUCCIÓN	5
1.1	Motivaciones del proyecto.....	6
2	ESTADO DEL ARTE	7
2.1	Concepto de simulación.....	7
2.2	Simulación de los sistemas eléctricos	10
2.2.1	Simulador digital	10
2.3	Tipos de estudios de sistemas eléctricos	12
2.3.1	Estudios en régimen transitorio electromagnético.....	12
2.3.2	Estudios en régimen transitorio electromecánico	12
2.3.3	Estudios en régimen transitorio lento	13
2.3.4	Estudios en régimen permanente	14
2.4	Modelos de elementos del sistema eléctrico	14
2.4.1	Modelos de elementos para el cálculo de flujos de cargas	14
2.4.2	Pérdidas eléctricas en la red.....	18
2.4.3	Tecnologías para reducir las pérdidas y saturación en sistemas de distribución. 18	
2.4.4	Reactancias en la red	21
3	MODELO DEL SISTEMA ELÉCTRICO	23
3.1	Introducción y sistema unifilar	23
4	METODOLOGÍA	26
4.1	Software	26
4.1.1	SimPowerSystem (Simulink).....	27
4.1.2	PowerWorld Simulator	27
4.1.3	ATP-EMTP	27
4.1.4	PSCAD/EMTDC.....	28

4.1.5	OpenDSS	28
4.1.6	PSS/E	28
4.2	PSS/E	30
4.2.1	Introducción	30
4.2.2	Creación de un caso	31
4.2.3	Simulación del flujo de cargas	35
4.2.4	Método de Gauss-Seidel	40
4.2.5	Herramientas de simulación.....	42
4.3	Python	44
4.3.1	Introducción	44
4.3.2	Python aplicado al flujo de cargas con PSS/E.....	44
4.4	SQL	48
4.4.1	Introducción	48
4.4.2	Python relacionado con SQL	49
5	Resultados	51
5.1	Análisis y representación gráfica del índice de carga de las líneas ante variaciones de generación.....	51
5.2	Tablas.....	53
5.2.1	Potencias.....	53
5.2.2	Casos	55
5.2.3	Resultados.....	55
5.3	Futuras investigaciones	57
6	Anexos.....	59
6.1	Instrucciones en Python.....	59
7	Conclusiones	78
8	Bibliografía.....	80

Lista de figuras

Figura 2.1.1: Estudio de una simulación. (Fuente y adaptado de: [3]) 10

Figura 2.3.1 Regímenes eléctricos clasificados por 3 dominios (Fuente y adaptado de: [25]) 12

Figura 2.4.1 Circuito eléctrico equivalente en π de una línea de transmisión (Fuente y adaptado de: [24]) 15

Figura 2.4.2 Circuito eléctrico equivalente del transformador monofásico. (Fuente y adaptado de: [24]) 16

Figura 2.4.3 (a) Circuito eléctrico equivalente de un generador síncrono; (b) Equivalente Norton. (Fuente y adaptado de: [24]) 17

Figura 2.4.4 . Modelo de la máquina asíncrona (Fuente y adaptado de: [24]) 17

Figura 2.4.5 Circuito saturado prereactancias (Fuente y adaptado de: [4]) 21

Figura 2.4.6 Circuito postreactancias (Fuente y adaptado de: [4]) 21

Figura 2.4.7 Reactancia de línea en una línea de transmisión (Fuente y adaptado de: [4]). 22

Figura 2.4.8 Circuito saturado prereactancias (Fuente y adaptado de: [4]) 22

Figura 2.4.9 Circuito postreactancias (Fuente y adaptado de: [4]) 22

Figura 3.1.1 Zona A del circuito Norte. 24

Figura 3.1.2 Zona B del circuito Norte. 24

Figura 3.1.3 Zona c del circuito Norte. 25

Figura 4.2.1 Figura 5. Nudo de una red genérica. (Fuente y adaptado de: [24]) 37

Figura 5.1.1 Flujograma de funcionamiento del programa de estudio 1. 51

Figura 5.1.2 Flujograma de funcionamiento del programa de estudio 2. 52

Lista de ecuaciones

Ecuación 2.4.1.1 15

Ecuación 2.4.1.2 16

Ecuación 2.4.1.3 16

Ecuación 2.4.1.4 17

Ecuación 4.2.3.1 36

Ecuación 4.2.3.2 36

Ecuación 4.2.3.3 36

Ecuación 4.2.3.4 37

Ecuación 4.2.3.5 37

Ecuación 4.2.3.6 37

Ecuación 4.2.3.7 37

Ecuación 4.2.3.8	38
Ecuación 4.2.3.9	39
Ecuación 4.2.3.10	39
Ecuación 4.2.4.1	41
Ecuación 4.2.4.2	41
Ecuación 4.2.4.3	42

Lista de tablas

Tabla 4.1.6.1 Comparativa de los programas de simulación más visuales	29
Tabla 4.2.3.1 condiciones del problema en términos de ecuaciones e incógnita.....	39
Tabla 4.2.5.1 Comparativa de los programas de simulación más eficientes.....	43
Tabla 5.2.1.1 Potencias de las centrales generadoras estudiadas	54
Tabla 5.2.3.1 Caso de mayor saturación eléctrica.....	57
Tabla 5.2.3.2 Caso de mayor saturación eléctrica, estado de las centrales.	57
Tabla 5.2.3.3 Caso de mayor saturación eléctrica, reactancias necesarias bajar saturación al 80%	57

1 INTRODUCCIÓN

Debido al auge en la fabricación y compraventa de los coches eléctricos en el mundo y en menor medida en España; el uso de energía del país tenderá a desplazarse con el tiempo de combustibles fósiles para transporte de personas a generación eléctrica, y más tarde serán sustituidos por fuentes de energía renovables.

Un aumento de dependencia de la red eléctrica para la recarga de baterías de gran capacidad o el uso de cada vez más electrodomésticos en el hogar supondrá una sobrecarga de consumo en la red inasumible por las instalaciones eléctricas actuales. Sin embargo, se puede dar el caso de que solamente ciertas ramas de la red alcancen esos picos que puedan llegar a quemar la línea mientras que el resto de la red seguiría trabajando en condiciones óptimas.

La solución, en principio más sencilla, es la construcción de nuevas líneas con más potencia. Esta propuesta rápidamente es descartada debido al coste económico que conlleva y al aumento en los plazos de ejecución debido a los estudios de impacto medioambiental que hay que realizar.

Otra de las medidas propuestas es la repotenciación de las líneas ya existentes. Dicha opción es menos costosa que la anterior, pero en muchos casos el aumento de potencia es bastante limitado. Además, con esta solución hay que contemplar el tener que dejar de utilizar la línea durante un tiempo, lo que supondría una gran desventaja.

Por último, la tercera opción para el aumento de potencia de las líneas es la gestión dinámica de estas. Desde el punto de vista de la operación, el principal desafío de la gestión dinámica basada en las normas IEEE 738 y CIGRE TB 601 es el control de los flujos de potencia en redes malladas. En este caso se plantea una metodología que permite controlar dicho flujo en base a la instalación de reactancias saturables controladas electrónicamente que modifican la reactancia total de la línea. La metodología define una estrategia para la definición de los valores de reactancia necesarios para realizar dicho control. Con esta solución avanzamos hacia las 'redes inteligentes'.

1.1 Motivaciones del proyecto

El gran aumento en los últimos años y en los próximos de consumo eléctrico en los hogares debido al auge del coche eléctrico generará, sin la correspondiente mejora de las líneas eléctricas de nuestro país, una ineficiencia en el transporte de energía eléctrica.

La imposibilidad de la construcción de nuevas líneas por motivos políticos y socioeconómicos, así como el coste y perjuicio para los clientes por tener que dejar de utilizar líneas para su repotenciación, crean la necesidad de explotar la línea de forma dinámica para así, poder aumentar la potencia de transporte de estas.

La gestión dinámica de capacidad de las infraestructuras de transporte y distribución de energía eléctrica se dimensionan, con carácter general, en base a un nivel de capacidad estático basado en condiciones ambientales desfavorables. En el caso de REE se utiliza un rating estático estacional que define un nivel de capacidad diferente para cada estación del año. El P.O. 1.2 define también la posibilidad de establecer niveles dinámicos de capacidad si se utiliza una medida directa en tiempo real de las condiciones ambientales.

Desde el punto de vista de la operación, el principal desafío de la gestión dinámica basada en las normas IEEE 738 y CIGRE TB 601 es el control de los flujos de potencia en redes malladas. En este Proyecto se plantea una metodología que permite controlar dicho flujo en base a la instalación de reactancias saturables controladas electrónicamente que modifican la reactancia total de la línea. La metodología define una estrategia para la definición de los valores de reactancia necesarios para realizar dicho control.

De forma adicional, en el Proyecto se realiza el análisis de soluciones comerciales disponibles. La metodología desarrollada se aplica a una red de distribución de 132 kV.

2 ESTADO DEL ARTE

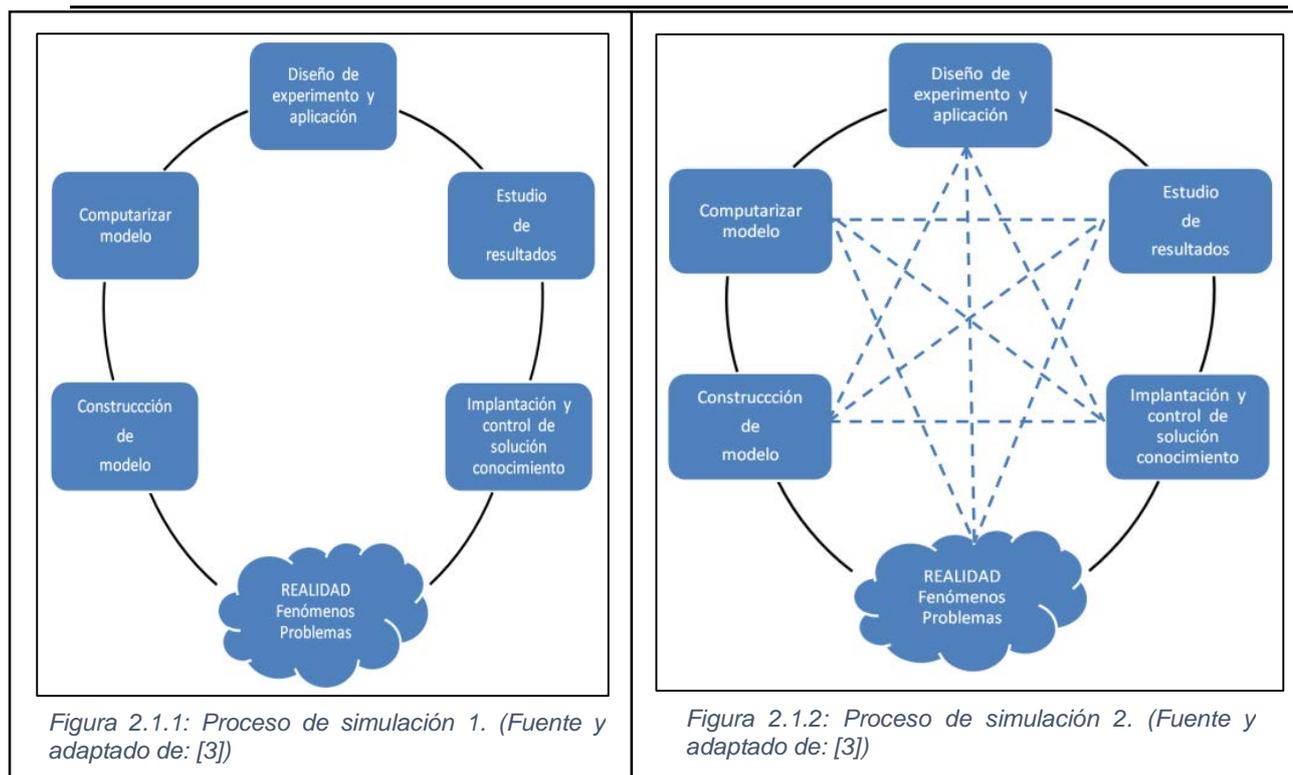
2.1 Concepto de simulación

Se ha definido a la simulación, en general, como la experimentación en un modelo y, a la digital, como aquella en que es posible “introducir” el modelo en una computadora digital, utilizando algún lenguaje apropiado para ello.

Al aproximarse a alguna parte de la realidad en que acontecen fenómenos o existen problemas, el proceso para poder llevar a cabo una simulación que coadyuve a generar conocimiento o soluciones, requiere la realización de ciertas etapas, como las que conceptualizamos a continuación:

- Construcción del modelo susceptible de poder ser “introducido” en una computadora digital.
- Introducción del modelo a la computadora, usando un lenguaje apropiado.
- Diseño del experimento y aplicación de este en el modelo computarizado.
- Estudio y análisis de los resultados.
- Implantación y control de alternativas de solución en la realidad, o del conocimiento generado.

Estas etapas, por lo general se llevan a cabo en el orden que se menciona en la Figura 2.1.1 (Proceso de simulación 1), pero es indispensable reconocer que al realizar cada una de ellas, es necesario tomar en cuenta a las otras, para mantener una aproximación sistémica de todo el proceso; es por esto que una representación del mismo, un modelo conceptual de dicho proceso, puede representarse como el que se propone en la Figura 2.1.2 (Proceso de simulación 2).



Justificaciones de la utilización de los procesos de simulación:

- Los sistemas reales son tan complejos debido a la inmensidad de variables que tienen en cuenta que hacen imposible describirlos mediante modelos matemáticos simples, es decir, de resolución analítica sencilla. Una simulación computarizada es la única forma de resolver problemas de tal inmensidad.
- La simulación permite comparar entre distintos diseños posibles con cero costos (o uno mucho más bajo que real), permitiendo encontrar la solución que mejor se adecua a las necesidades existentes.
- Se puede analizar el intervalo de tiempo deseado con el fin de sacar conclusiones acertadas acerca del comportamiento del sistema bajo las condiciones de funcionamiento a estudiar, desde los primeros microsegundos hasta varios minutos, e incluso periodos más largos.
- Hace posible alcanzar un conocimiento profundo sobre qué variables son las más importantes para el resultado y cómo interactúan entre sí.
- Permite experimentar nuevos procedimientos, estructuras, etc., sin interrumpir la operación del sistema real.

- Se pueden manipular nuevas situaciones de las que se tienen un conocimiento y experiencia limitados, para analizar y preparar eventos futuros.

Desventajas de los procesos de simulación

- Es posible que para obtener un valor sencillo sea necesario desarrollar modelos de simulación tan extensos que no sea rentable este proceso. Además, la fiabilidad de este resultado vendrá determinada por la extensión de las variables para tener en cuenta, la complejidad del sistema y el correcto diseño de su simulación.
- Un modelo de desarrollo del sistema incorrecto puede suponer un resultado irreal estando el fallo no en las pruebas hechas sino en el diseño de todo el circuito.
- Puede darse el caso de obtener un resultado no interpretable debido a la complejidad del proceso simulado.
- La computarización de un sistema de gran complejidad puede hacerse costosa en términos de tiempo, coste económico, y extensión de programación. Por suerte este inconveniente se está superando gracias al desarrollo de paquetes de software de aplicación específica que incorporan modelos básicos de todos los elementos a considerar en el proceso de simulación.
- No es posible validar los resultados de manera concluyente.

El flujograma siguiente expone los pasos que componen un estudio típico de simulación y las relaciones que existen entre ellos. Cabe destacar se trata de un flujograma referido a un análisis secuencial, pero también es posible diseñar un proceso de manera iterativa.

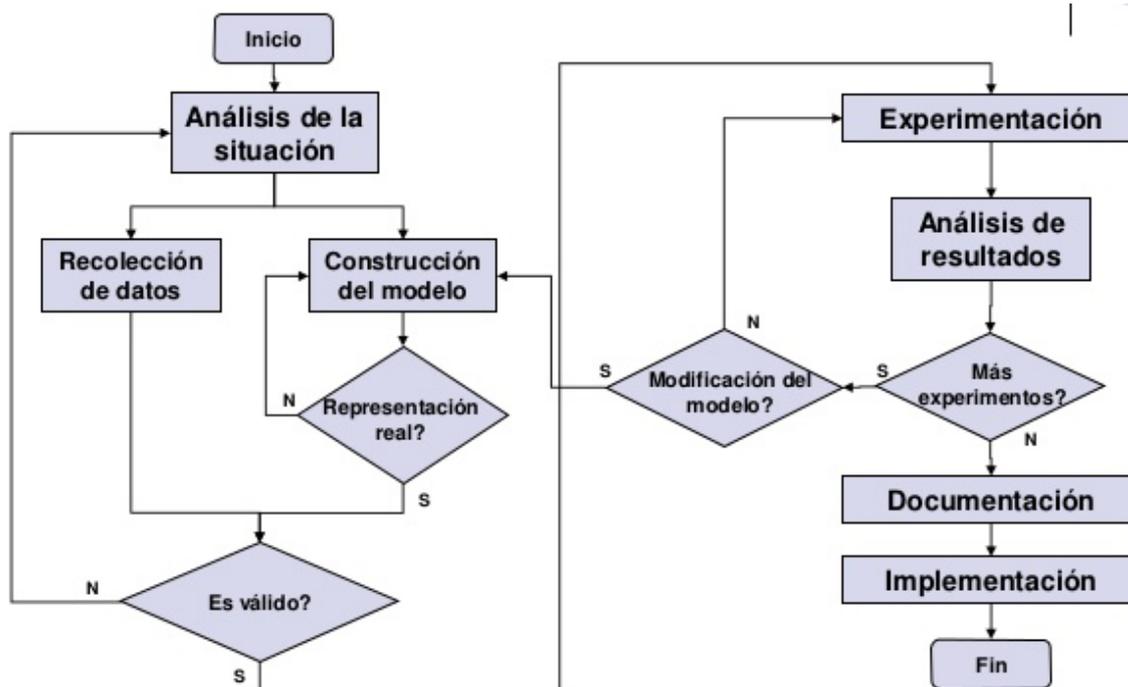


Figura 2.1.1: Estudio de una simulación. (Fuente y adaptado de: [3])

2.2 Simulación de los sistemas eléctricos

2.2.1 Simulador digital

Con el ordenador se puede hacer de forma más rápida, ordenada, segura, práctica y utilizando menos espacio físico, distintas tareas que antes requerían una considerable cantidad de tiempo. La extraordinaria evolución de la tecnología de los ordenadores en las últimas décadas ha supuesto un gran desarrollo en la aplicación de los procesos de cálculo. Paralelamente con esto, los simuladores digitales han tenido también una evolución acentuada, convirtiéndose para los ingenieros eléctricos en uno de sus instrumentos básicos permitiendo un análisis cada vez más realista de la dinámica de los sistemas de energía eléctrica.

Estas herramientas están basadas en diversas aplicaciones software que simulan los distintos elementos del sistema eléctrico resolviendo las ecuaciones matemáticas que describen su comportamiento. Así, se obtienen de forma digital las magnitudes de los parámetros (tensión, intensidad, etc.) que aparecen en el sistema bajo diversas hipótesis de funcionamiento. En general, estos simuladores digitales cuentan con un procesador gráfico que representa la evolución de los parámetros calculados.

- Las ventajas más relevantes de este tipo de herramientas de simulación son:
- Tienen capacidad para modelizar sistemas eléctricos de grandes dimensiones.
- Permiten realizar múltiples variaciones de la topología del sistema analizado de forma más sencilla que los simuladores analógicos.
- Permiten elegir el incremento de tiempo utilizado en la realización de la simulación.
- Se han desarrollado métodos y algoritmos adecuados para reducir los tiempos de cálculo.
- Ocupan menos espacio físico y son más económicos.

Sin embargo, también presentan inconvenientes. Los más relevantes son:

- No operan en tiempo real.
- Pueden presentar problemas de inestabilidades numéricas como consecuencia del algoritmo de resolución o el intervalo de tiempo empleado, lo que puede conducir a errores en los resultados.

Existen en la actualidad numerosas aplicaciones software para realizar cálculos eléctricos. Estas aplicaciones van desde paquetes sencillos creados y distribuidos por los propios fabricantes de equipos, que sirven como herramienta de ayuda para el diseño y la selección de equipos con retos (aparamenta de baja tensión, luminarias, etc.), hasta complejas y completas aplicaciones software, de carácter comercial, que permiten la modelización y simulación de toda una gran variedad de equipos y fenómenos relacionados con el funcionamiento del sistema eléctrico.

2.3 Tipos de estudios de sistemas eléctricos

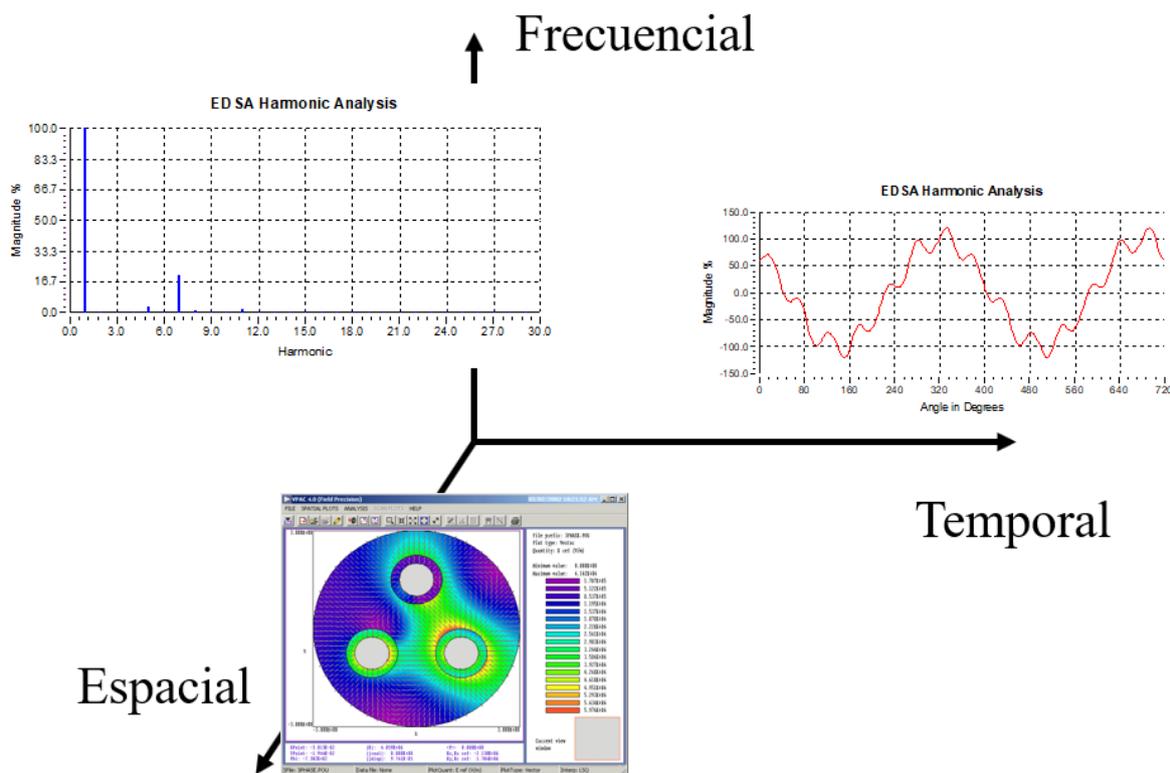


Figura 2.3.1 Regímenes eléctricos clasificados por 3 dominios (Fuente y adaptado de: [25])

2.3.1 Estudios en régimen transitorio electromagnético

Estos estudios analizan fenómenos medibles en término de microsegundos, de forma que las ondas de tensión e intensidad ya no se pueden suponer como sinusoidales, con variación frecuencial de 50Hz. Este tipo de estudios requieren una representación diferente, y su estudio supondría un análisis muy detallado de dinámica de algunos de los elementos del sistema que intervienen en el estudio. Los fenómenos que se incluyen en el ámbito de los transitorios electromagnéticos son: sobre tensiones provocadas por ondas tipos Rayo, sobre tensiones provocadas por maniobras de interruptores y seccionadores, fenómenos de resonancia, resonancia subsíncrona, respuesta de sistemas de protección de alta velocidad, etcétera. Es un régimen temporal que no nos aporta valor para la resolución de consumos medios debido a su corta característica temporal.

2.3.2 Estudios en régimen transitorio electromecánico

Fenómenos transitorios rápidos provocados por las oscilaciones electromecánicas de los rotores de los generadores y las interacciones con los dispositivos de control. El período de tiempo es entre 0.1 segundos y 10 segundos.

Las oscilaciones pueden dar lugar a una pérdida de estabilidad del sistema. Se puede perder el sincronismo de los generadores y llegar a producirse un colapso de la tensión. Estos estudios analizan la estabilidad del ángulo y de la tensión:

- **Estabilidad del ángulo.** analiza la dinámica de los movimientos del rotor alrededor del punto de sincronismo. dependiendo de la amplitud de dichos movimientos (perturbaciones) se distingue entre:
 - Estabilidad transitoria: analiza los movimientos de elevada amplitud durante perturbaciones fortuitas (cortocircuitos, etc.).
 - Estabilidad estática o de pequeña señal: tiene en cuenta las fluctuaciones de pequeña amplitud que se producen durante la operación normal del sistema, por la interacción de los controles ante pequeñas perturbaciones alrededor del punto de equilibrio.

El objetivo de estos estudios consiste en analizar sí, es una perturbación, el sistema se vuelve inestable o vuelve a un estado de operación estable.

- **Estabilidad de tensión:** analiza la capacidad de un sistema eléctrico para alcanzar niveles de tensión fijos (mayores de 0V) en todos sus nudos. Cuando un sistema sufre y la estabilidad de tensión se produce un colapso en las tensiones, qué puede llevar a una pérdida parcial o total del sistema. Suelen venir asociadas la inestabilidad de la tensión con la inestabilidad del ángulo, pero el colapso de tensión se produce antes que la separación de los ángulos.

2.3.3 Estudios en régimen transitorio lento

En estos estudios se analiza la evolución del sistema en periodos de tiempo que van desde minutos hasta varias horas. Su objetivo consiste en estudiar la dinámica del sistema asociada con el control de la tensión y de la frecuencia. El método de trabajo se basa en discretizar el intervalo de estudio y efectuar flujos de carga consecutivos, pero teniendo en cuenta:

- El control primario de la frecuencia
- El control secundario de tensión
- La respuesta de los elementos del sistema (disparó por sobrecarga de elementos, deslastre de cargas o funcionamiento en Isla de los buses, disparo de generadores, etc.).

2.3.4 Estudios en régimen permanente

Es el régimen temporal más largo. Abarca desde minutos (si se tiene en cuenta el régimen transitorio lento dentro de este) hasta el infinito; pero en términos eléctricos normalmente no suelen interesar los valores temporales inmediatamente después de un suceso electromagnético, dentro del régimen humano de sensación temporal, es decir cuando podemos reaccionar ante los cambios sin necesitar de la ayuda de otras máquinas.

Los flujos de carga de un sistema de gran complejidad como es el de nuestro caso suelen estudiarse en este régimen y el programa como más usado en la industria es PSS/E, se explicará porque se escogió este sistema para estas mediciones y no otro en el apartado de [Metodología, software](#).

2.4 Modelos de elementos del sistema eléctrico

2.4.1 Modelos de elementos para el cálculo de flujos de cargas

Se introducen, a continuación, algunos modelos sencillos de componentes utilizados habitualmente en la simulación de los sistemas eléctricos. Se ha optado por introducir únicamente los modelos a la frecuencia fundamental, aunque es aconsejable incluir su dependencia con la frecuencia para obtener resultados más exactos en los flujos de carga armónicos.

Línea de transmisión

Desde un punto de vista sencillo, una línea de transmisión tiene como parámetros básicos por unidad de longitud:

- Resistencia (R).
- Inductancia (L).
- Capacitancia (C).
- Conductancia (cr).

Estos parámetros pueden ser representados mediante un circuito equivalente en π , tal como el que se muestra en la Figura 2.4.1. La reactancia resistiva e inductiva de la línea se incluye en la rama serie, mientras que la capacitancia y la conductancia se dividen en dos ramas conectadas en derivación en los extremos del modelo. Normalmente la conductancia puede ser despreciada, ya que presenta un valor muy pequeño respecto al resto de elementos.

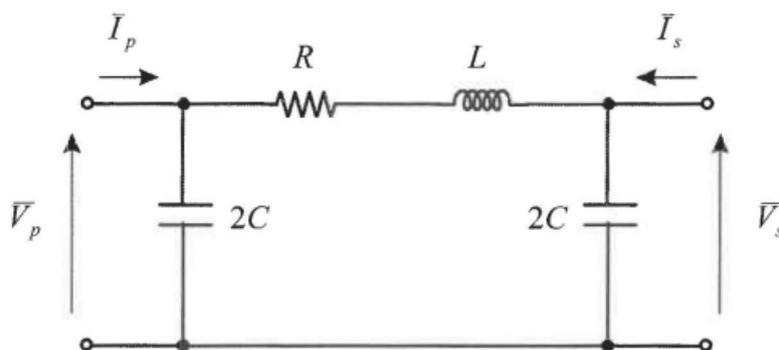


Figura 2.4.1 Circuito eléctrico equivalente en π de una línea de transmisión
(Fuente y adaptado de: [24])

A partir del circuito de la Figura 2.4.1.1 se obtiene la ecuación matricial siguiente,

$$\begin{bmatrix} \bar{I}_p \\ \bar{I}_s \end{bmatrix} = \begin{bmatrix} \frac{1}{R + j\omega L} + \frac{j\omega C}{2} & -\frac{1}{R + j\omega L} \\ -\frac{1}{R + j\omega L} & \frac{1}{R + j\omega L} + \frac{j\omega C}{2} \end{bmatrix} * \begin{bmatrix} \bar{V}_p \\ \bar{V}_s \end{bmatrix}$$

Ecuación 2.4.1.1

Cuando el modelo se utiliza para simular sistemas eléctricos en presencia de armónicos, resulta necesario considerar, además, otros efectos que pueden ser ignorados a la frecuencia de 50 Hz:

- Dependencia con la frecuencia de la resistencia e inductancia de los conductores.
- Reducción de la longitud máxima de la línea a la que puede ser utilizado el modelo en parámetros concentrados.
- Desequilibrios de las líneas.
- Transposiciones de los conductores de las líneas.

Transformador

El circuito equivalente de un transformador monofásico o el equivalente por fase de un transformador trifásico sin considerar el acoplamiento entre fases puede representarse mediante el circuito de la Figura 2.4.1.2. Este modelo sencillo permite realizar la simulación en régimen permanente en condiciones de funcionamiento donde no exista saturación. Como puede observarse, el modelo se construye mediante la conexión en cascada de un transformador ideal y con el circuito equivalente del transformador referido al primario o al secundario según el caso.

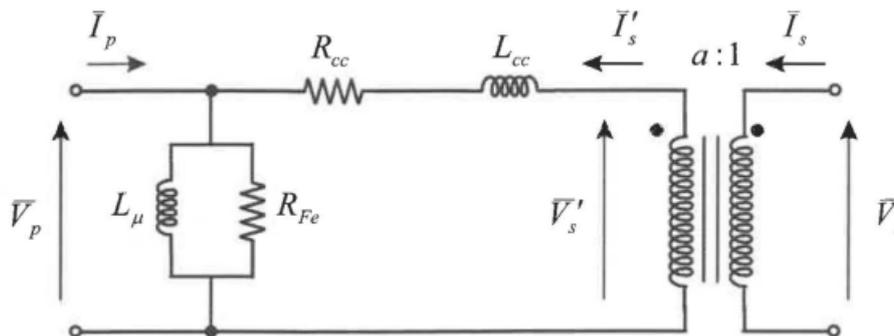


Figura 2.4.2 Circuito eléctrico equivalente del transformador monofásico. (Fuente y adaptado de: [24])

$$\begin{bmatrix} \bar{I}_p \\ \bar{I}_s \end{bmatrix} = \begin{bmatrix} \frac{1}{j\omega L_\mu} + \frac{1}{R_{Fe}} + \frac{1}{R_{cc} + j\omega L_{cc}} & -\frac{a}{R_{cc} + j\omega L_{cc}} \\ -\frac{a}{R_{cc} + j\omega L_{cc}} & -\frac{a^2}{R_{cc} + j\omega L_{cc}} \end{bmatrix} * \begin{bmatrix} \bar{V}_p \\ \bar{V}_s \end{bmatrix}$$

Ecuación 2.4.1.2

Donde,

$\bar{Z}_{cc} = R_{cc} + j\omega L_{cc}$ Es la impedancia de la rama serie que puede obtenerse a partir del ensayo de cortocircuito del transformador.

$\bar{Y}_0 = \frac{1}{R_{Fe}} + \frac{1}{j\omega L_\mu}$ Es la admitancia de la rama derivación que puede obtenerse a partir del ensayo de vacío del transformador.

Máquina síncrona

La máquina síncrona constituye uno de los elementos más complejos de modelar dentro de los sistemas eléctricos de potencia, especialmente cuando se trata de estudiar problemas de estabilidad o de penetración armónica en máquinas que trabajan en la zona de saturación. En este apartado se introduce únicamente un modelo equivalente por fase muy simplificado, válido únicamente para máquinas de rotor simétrico funcionando en la zona lineal. En la Figura 2.4.1.3 se muestra el circuito equivalente del generador síncrono. La ecuación que representa el modelo es,

$$\bar{I}_g = \bar{I}_0 - \frac{1}{jX_s} * \bar{V}_g$$

Ecuación 2.4.1.3

\bar{I}_0 Intensidad interna generada por la máquina. Depende de la velocidad del motor primario y de la intensidad de excitación.

\bar{I}_g Intensidad generada por la máquina.

X_s Impedancia (reactancia) síncrona del generador.

\bar{V}_g Tensión generada por la máquina.

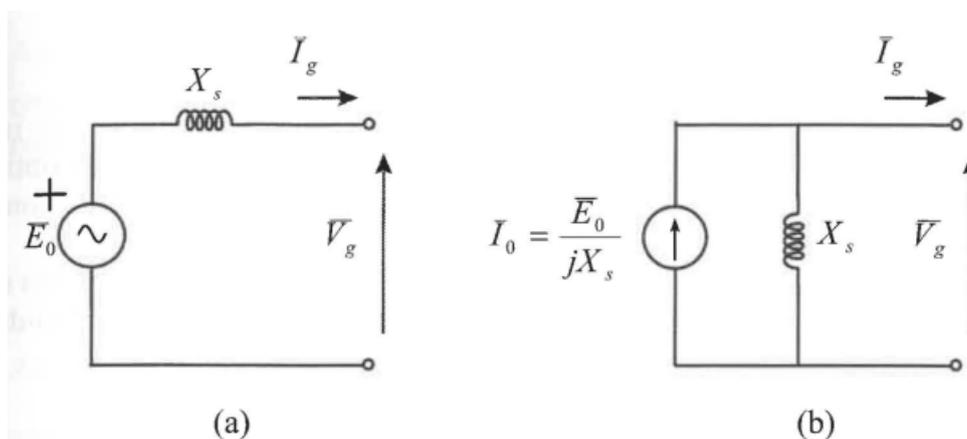


Figura 2.4.3 (a) Circuito eléctrico equivalente de un generador síncrono; (b) Equivalente Norton. (Fuente y adaptado de: [24])

Máquina asíncrona

La máquina asíncrona puede representarse de forma sencilla en régimen permanente mediante el modelo que se muestra en la Figura 2.4.1.4.

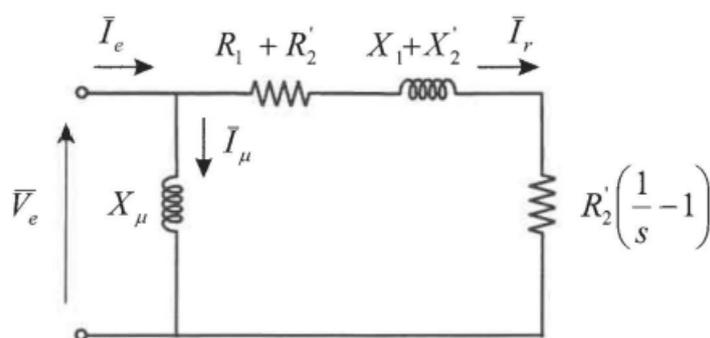


Figura 2.4.4 . Modelo de la máquina asíncrona (Fuente y adaptado de: [24])

$$\bar{Y}_{ma} = \frac{1}{jX_\mu} + \frac{1}{\frac{R'_2}{s} + jX'_2}$$

Ecuación 2.4.1.4

Donde,

\bar{Y}_{ma} Admitancia equivalente de la máquina asíncrona.

X_{μ} Reactancia de magnetización.

R_1 Resistencia equivalente del devanado del estator.

R'_2 Resistencia equivalente del devanado rotórico referida al estator.

X_1 Reactancia de dispersión del estator.

X'_2 Reactancia de dispersión del rotor referida al estator.

s Deslizamiento.

2.4.2 Pérdidas eléctricas en la red.

Las pérdidas de energía y potencia se deben a las condiciones propias de las instalaciones.

En general se destacan dos tipos de pérdidas:

- **Pérdidas Óhmicas o Pérdidas en el Cobre:** son provocadas por la circulación de corriente eléctrica a través de los circuitos. Su magnitud depende entonces de las características de las redes y de la carga a que éstas se ven exigidas. También se las denomina pérdidas de carga.
- **Pérdidas en Vacío:** Estas se originan por la circulación de corrientes de magnetización en los núcleos de hierro de transformadores y otros equipos de la red. También se incluyen en esta categoría las pérdidas por corrientes parásitas en aisladores de líneas y por efecto corona. Estas últimas se manifiestan fundamentalmente en redes de transmisión. En distribución son insignificantes. A diferencia de las pérdidas de carga, las pérdidas de vacío aparecen toda vez que los circuitos están energizados, independientemente del nivel de carga o circulación de flujo de potencia por los mismos.

2.4.3 Tecnologías para reducir las pérdidas y saturación en sistemas de distribución.

Existen una variedad de tecnologías y métodos que pueden implementarse para reducir las pérdidas técnicas en redes de distribución. Algunas pueden implementarse en forma eficiente en los sistemas existentes (ejemplo optimización de la tensión, y rebalanceo de fases), mientras que otras resultan atractivas solo cuando se instalan nuevos circuitos, y se reacondicionan por completo circuitos existentes. Las reducciones de pérdidas que pueden lograrse dependen en gran medida de las características de las redes, de los estándares de

diseño de las empresas, y de la forma de operar los sistemas. Por lo tanto, para determinar si una opción es factible y económicamente viable debe hacerse un análisis específico en el sistema en cuestión. Para los objetivos de este trabajo, se asumen valores generales relativos a las mejoras que pueden lograrse.

Las distintas tecnologías de reducción de pérdidas consideradas en el estudio son las siguientes:

- **Reducción controlada de la tensión:** Este caso se trata de una técnica que hace uso de la sensibilidad de la demanda eléctrica a las variaciones de la tensión. Cuando se disminuye la tensión al usuario final, la demanda neta de potencia activa y reactiva también disminuye. No todas las cargas se comportan de la misma manera, porque en algunos casos la medida es más efectiva que en otros. A esta técnica se la conoce por su nombre en inglés como “Conservation Voltage Reduction (CVR)”. Existen varias técnicas y métodos, que van desde el simple control sobre los “taps” de los transformadores de estación, hasta sistemas de control de lazo cerrado que retroalimentan información desde el usuario final.

- **Cambio de conductor en alimentadores:** El uso de conductores de baja resistencia permiten disminuir las pérdidas técnicas. En este caso aplica un concepto similar, solo que la utilización de conductores trapezoidales es limitada, por lo que refiere mayormente al uso de conductores de mayor sección.

En general es muy difícil justificar el reemplazo de un conductor en un alimentador existente solo por consideraciones de eficiencia, sin embargo, como sucede en muchos otros casos, si el conductor se debe reemplazar por otros motivos (corrosión, rotura, capacidad), una alternativa de menor pérdida óhmica puede resultar viable si se realizan cálculos integrales de eficiencia y economía a lo largo de toda la vida útil del sistema eléctrico.

- **Reconfiguración de mínimas pérdidas:** Los circuitos de las redes de distribución son extensos y presentan una estructura compleja. La forma en que se conectan entre si los distintos circuitos del sistema pueden variarse de acuerdo con las necesidades. Mediante procedimientos de optimización, es posible determinar cuál es la configuración que minimiza las pérdidas manteniendo el nivel de seguridad y confiabilidad requerido.

- **Mejora de la eficiencia de transformadoras de distribución:** Los transformadores de distribución representan una importante fuente de pérdidas en el sistema, principalmente debido a las pérdidas en el hierro, debido a que el factor de uso de estos es bajo (operan demasiado tiempo a baja carga). Se puede lograr una reducción de las pérdidas del orden del 20%-30%.
- **Optimización de la potencia reactiva:** Mediante una optimización del uso de los recursos de compensación de reactivo (bancos de condensadores) y de controladores de tensión (“taps” de transformadores, reguladores de tensión), se pueden minimizar los flujos de potencia reactiva por los circuitos, reduciendo por consiguiente las pérdidas en la red.

Mediante una optimización del uso de los recursos de compensación de reactivo (bancos de capacitores) y de controladores de tensión (“taps” de transformadores, reguladores de tensión), se pueden minimizar los flujos de potencia reactiva por los circuitos, reduciendo por consiguiente las pérdidas en la red.

- **Rebalanceo de fases:** En los circuitos secundarios de distribución, las cargas monofásicas se conectan a las diferentes fases de los circuitos. Si bien se trata de que las cargas en las distintas fases de los circuitos estén balanceadas, no siempre se logra en forma óptima. El desbalanceo de fases provoca corrientes de circulación que originan pérdidas.

Un rebalanceo apropiado de las fases es una medida de relativamente bajo costo que puede ayudar a reducir las pérdidas en distribución. Para ello, conforme a la medida, registro y características de uso de los clientes de cada alimentador, se puede llegar a reubicar a cada uno de los mismos, para llegar a una optimización de la carga y mejorar el factor de uso de cada fase.

- **Incremento de la tensión nominal:** una de las formas más eficaces de reducir las pérdidas en una red es elevar la tensión nominal de los circuitos de subtransmisión y distribución. una de las formas más eficaces de reducir las pérdidas en una red es elevar la tensión nominal de los circuitos de subtransmisión y distribución, como es el caso de las redes de alta tensión de 132kV, el estudiado en este trabajo.

- Derivación de la corriente a otras ramas:** Este sistema lo que hace es disminuir la saturación de alguna línea aumentando su impedancia. así se fuerza a que la corriente eléctrica se distribuya por las líneas paralelas a esta. es un caso muy sencillo en teoría, si se dispone de dos resistencias diferentes en paralelo considera estas líneas de gran longitud que debido a las pérdidas eléctricas se han convertido en impedancia, al aumentar la impedancia de una de ellas la corriente aumentará en la de la otra.

En el caso que nos ocupa para aumentar la impedancia de una línea sin tener más pérdidas eléctricas de las que querríamos nos encargamos de aumentar solamente la reactancia. Esto provocará solo un aumento de las pérdidas en potencias reactivas, y como todos sabemos, la potencia real consumida en cualquier máquina es activa.

2.4.4 Reactancias en la red

La variedad de las dificultades del desarrollo de la red pueden ser difíciles de gestionar. Para resolver estos problemas se suele recurrir a soluciones operativas como predisposición de la red o procedimientos operativos complicados tal como tendido de cable nuevo actualizaciones de la línea etcétera. Sin embargo, la empresa que nos atañe en este caso “Smart Wires”, Permiten aliviar estas dificultades y aumentar la capacidad de transferencia de la red reduciendo la congestión de las líneas más saturadas, y desviando la corriente a líneas paralelas subutilizadas.

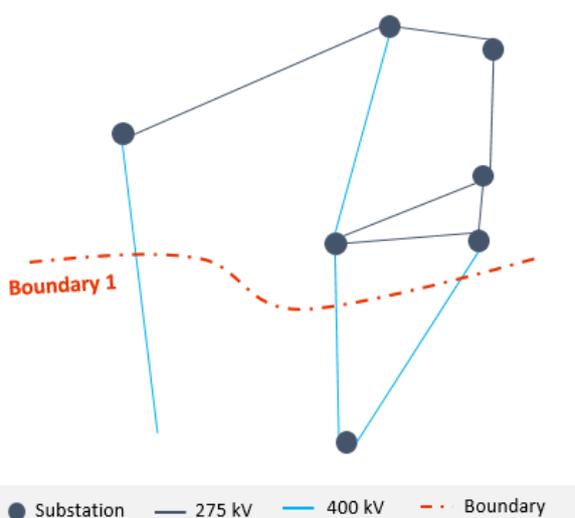


Figura 2.4.6 Circuito saturado pre reactancias (Fuente y adaptado de: [4])

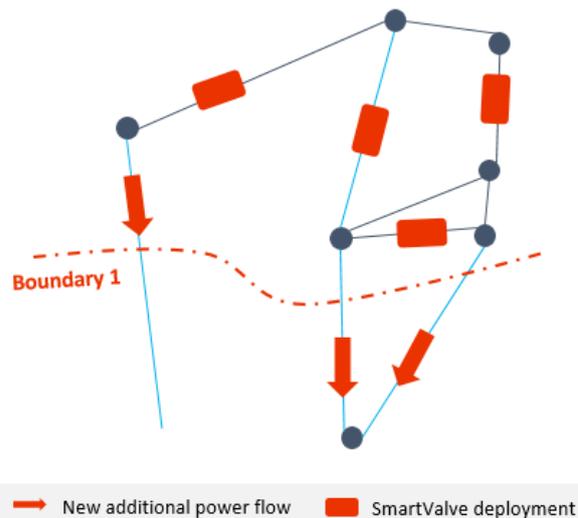


Figura 2.4.5 Circuito post reactancias (Fuente y adaptado de: [4])

El caso ejemplo de red en estado de saturación sería el de la Figura 2.4.4.1, mientras que con el añadido de las reactancias sería la Figura 2.4.4.2.



Figura 2.4.7 Reactancia de línea en una línea de transmisión (Fuente y adaptado de: [4])

El aumento de las energías renovables, así como su penetración en la red obligan a esta a trabajar en circunstancias no contempladas de alta saturación, pero en momentos muy cortos en el tiempo, lo cual daría poco rentable un tendido eléctrico, pero nada seguro mantener el original.

Se trata de una tecnología de fácil escalabilidad con una curva exponencial desarrollo tecnológico que se encuentra ahora mismo el principio de su desarrollo, esto se traduce en que si bien ahora la tecnología puede no ser rentable en el futuro es capaz de promover la gestión dinámica de toda una red.

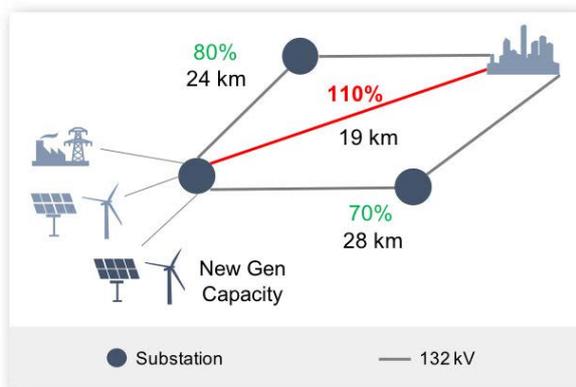


Figura 2.4.8 Circuito saturado prereactancias (Fuente y adaptado de: [4])

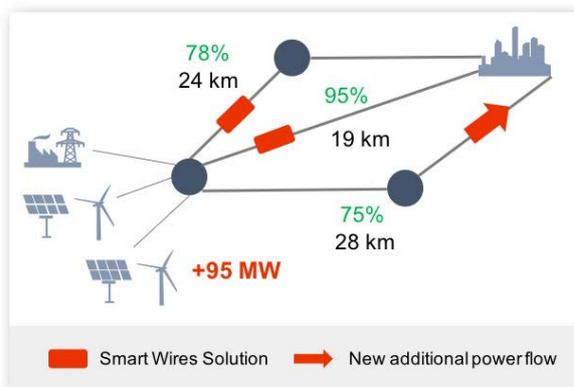


Figura 2.4.9 Circuito postreactancias (Fuente y adaptado de: [4])

Si poseemos una generación eólica, por ejemplo, capaz de alimentar una ciudad, pero no de alimentar debido a la saturación como se puede observar en la Figura 2.4.4.5, nos veríamos obligados de reducir la generación para evitar la rotura de la línea, pero estaríamos desperdiciando una gran cantidad de recursos y capital.

Aplicando una serie de reactancias regulables a la línea podemos sacar el máximo rendimiento de nuestra inversión sin destruir la línea, hacerla correr cierto peligro, o sustituirla por un conductor de mayor sección. Los frutos de esta tecnología pueden ser observados en la Figura 2.4.4.4.

En términos físicos, cada reactancia es capaz de generar $42\mu\Omega$. Esto se traduce en $X = L * w = L * 2\pi f = 42 * 10^{-6} * 2\pi * 50 = 0.0132\Omega$ por unidad.

En términos económicos el precio a pagar por cada unidad es de 16.500 €, haciendo una equivalencia en valores eléctricos obtenemos que cada ohmio sale a: $\frac{16500\text{€/unidad}}{0.0132\Omega/\text{unidad}} =$

1.250.000 €

3 MODELO DEL SISTEMA ELÉCTRICO

3.1 Introducción y sistema unifilar

El proyecto dispone de un modelo general de toda la red eléctrica española actualizada al último diseño que se posee del circuito. Al estar insertada y catalogada en un programa informático se pueden hacer todo tipo de pruebas y mediciones sobre cualquier modelo; y si se obtiene alguna mejora aplicarlos más tarde a la realidad. La modelación del sistema incluye todos los generadores eléctricos, subestaciones con sus flujos de potencia tanto activa como reactiva, y todas las ramas de conexión con distribución desde una tensión de generación 10.000V hasta la de transporte de 400kV.

En nuestro caso solo nos interesa analizar la red de distribución que trabaja a 132KV. Esta se encuentra conectada a la red general de transporte de 400KV gracias a subestaciones elevadoras de tensión.

El diagrama se guardó con un estado de generación de las centrales, y de consumo de las cargas específico; el estado normal. Y las simulaciones solo se pueden realizar para ese estado a no ser que se modifiquen manualmente tales valores para cada una de las centrales y cargas. Esto se soluciona utilizando programas externos como Python. Pero existe otro problema derivado de la simplificación del sistema que no nos permite trabajar en una situación real. Si suponemos una nueva situación de generación y demanda en la zona escogida, éstas en realidad se verían afectadas por cambios en otras zonas de la red española. Esto es lo que se conoce como condiciones de contorno.

Debido a la complejidad que supondría experimentar en esta situación tanto matemáticamente como computacionalmente, utilizaremos al resto de la red española como fuente o sumidero de nuestra demanda o generación de potencia equilibrando así la potencia total generada. Es decir, supondremos las condiciones de contorno constantes en cualquier situación de generación y demanda en nuestra zona de estudio.

Considerar la red como aislada no presenta grandes incongruencias en los resultados finales ya que, al trabajar a 132 Kv, el flujo de potencia estudiado no presenta problemas ni en evacuación ni en alimentación para la red de 400 kV.

La zona estudiada presenta una gran potencia de generación eólica con una baja demanda eléctrica. Esto hace que los problemas de saturación se presenten en casos de máxima generación eólica; esta al rebasar en gran cantidad la demanda de los consumidores locales debe ser evacuada a través de la red de 132 kV y de ahí a la de transporte de 400 kV.

El problema del Python han actúa sobre el modelo de PSS/E, cambiando los valores y por los parámetros de los elementos de la red 132kV en la zona de interés.

A continuación, se muestra el diagrama unifilar de la red al completo dividido en tres segmentos debido al tamaño de esta.

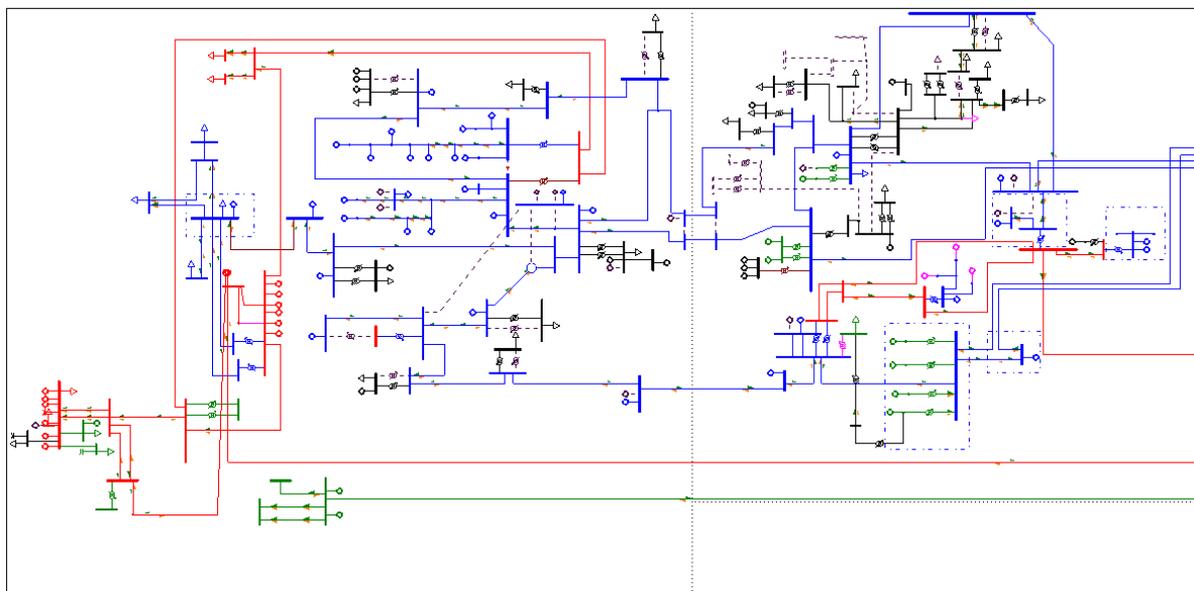


Figura 3.1.1 Zona A del circuito Norte.

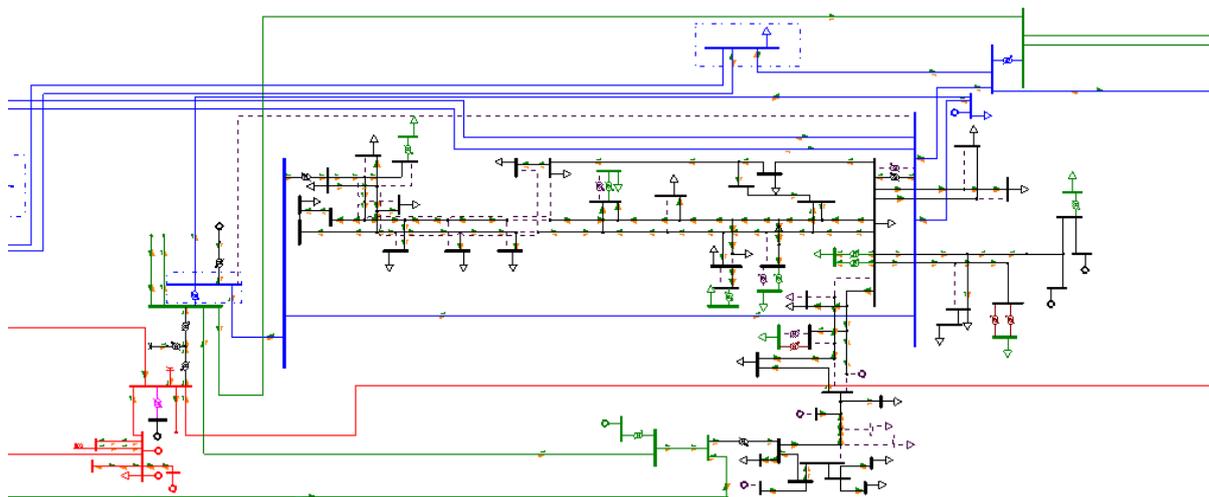


Figura 3.1.2 Zona B del circuito Norte.

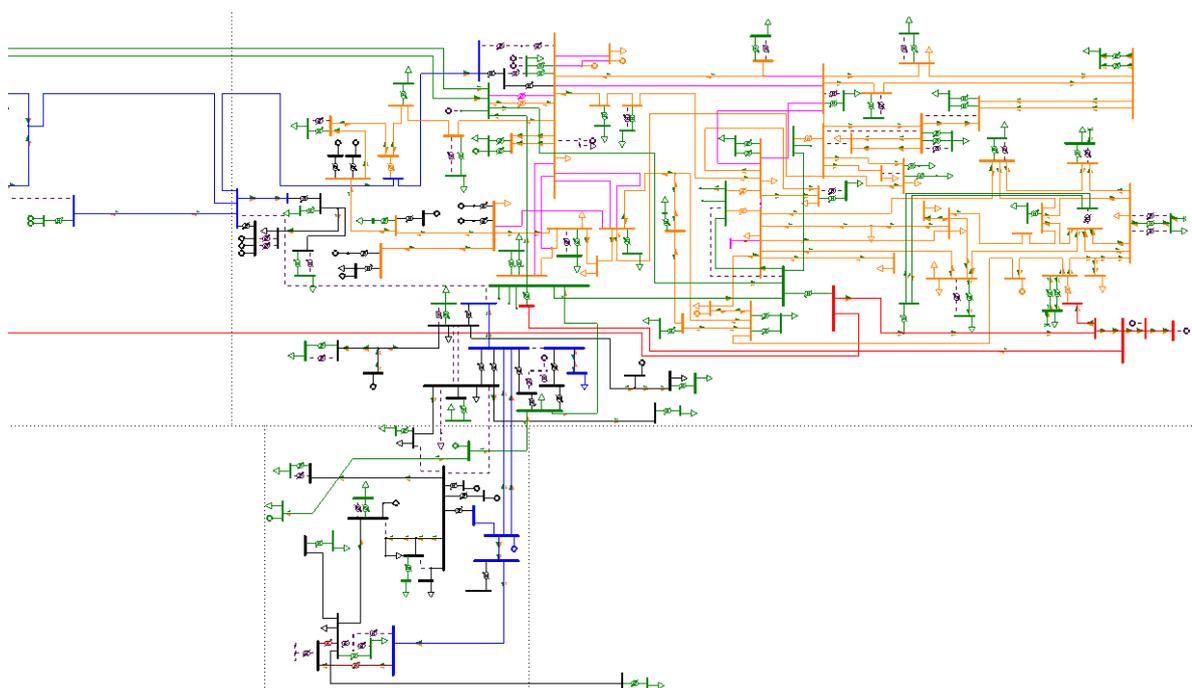


Figura 3.1.3 Zona c del circuito Norte.

Si juntamos las 3 imágenes en una sola obtenemos todas las redes eléctricas española representada en buses generadores y cargas.

En nuestro caso solo nos interesan las líneas de color azul, que son las correspondientes a la red de 132 kV. La situación dentro del esquema se corresponde con la situación geográfica de la red del norte de España.

En cuanto a la simbología, la identificación de los elementos es sencilla. Los buses están representados como barras o líneas gruesas de corta longitud sin quiebras. Perpendiculares a estas se sitúan unas rectas más finas que conectan dos buses entre sí, estas son las líneas de alta tensión de red. Los generadores vienen representados por un círculo y una línea que lo conecta un bus. Las cargas están dibujadas como un triángulo en línea que lo conecta a un bus de igual manera. Por último, los transformadores vienen representados por dos líneas paralelas entre sí en zig zag situadas perpendiculares a la línea en la que se encuentran. Siempre que una línea se encuentra con un transformador, este conectada dos buses pertenecientes a dos tensiones diferentes de red.

Por otro lado, de las líneas están preparadas para aumentar el grosor en función de la potencia del transporte. Cuanta más potencia, más gruesa es la línea a la hora de simular el circuito. Además, podemos observar el sentido de la corriente que atraviesa la línea ayudándonos de dos flechas; la de color rojo se corresponde con una potencia activa y la de amarillo con una reactiva.

El tratamiento de las potencias de los generadores del circuito se encuentra en el apartado de [tablas, potencias](#).

4 METODOLOGÍA

4.1 Software

Tras introducir brevemente las características especiales de la energía eléctrica, la estructura de los sistemas de energéticos y la regulación de dichos sistemas eléctricos en el ámbito español; se procede a introducir en este apartado las principales herramientas digitales de simulación de los sistemas de energía eléctrica con una breve explicación de estos. La mejor manera de conocer la respuesta de los sistemas de energía eléctrica ante situaciones dadas por condiciones normales o extraordinarias es simulando sobre el propio sistema. Dado que esta opción es poco menos que imposible, ya que se dejaría fuera de servicio dichas instalaciones, es necesario recurrir a la simulación. Se pueden utilizar tanto simuladores analógicas como simuladores digitales e híbridos. Dejando a un lado los simuladores analógicos e híbridos, los simuladores digitales van ganando terreno con el paso del tiempo, convirtiéndose para los ingenieros de hoy en día en una de sus herramientas básicas para analizar los sistemas de energía eléctrica. Dichas herramientas están basadas en aplicaciones software que modelan el comportamiento de los distintos elementos que forman los sistemas eléctricos, a través de la resolución de modelos matemáticos que caracterizan dicho comportamiento. Las ventajas que otorga la utilización de este tipo de simuladores son evidentes:

- Simulación de sistemas eléctricos de grandes dimensiones.
- Cambios de topología del sistema de manera mucho más cómoda que los simuladores analógicos.
- Capacidad de elegir el incremento de tiempo utilizado en la simulación.
- Ocupan un menor espacio físico y son más económicos que los simuladores analógicos.

Pero como todo, también presentan algunos inconvenientes:

- No pueden operar en tiempo real.
- Pueden presentar inestabilidades numéricas, bien por el algoritmo empleado o bien por el intervalo de tiempo utilizado, lo que puede llevar a errores en los resultados.

Debido a que en la actualidad existen multitud de herramientas informáticas para analizar los sistemas de energía eléctrica, se describirán algunas de ellas en este apartado:

4.1.1 SimPowerSystem (Simulink)

“SimPowerSystem” es una “*Blokset*”, término usado para referirse a una librería de bloques, de Matlab/Simulink. Utiliza una interfaz gráfica que permite construir el circuito eléctrico. Dentro de sus usos destacan: análisis transitorios, análisis de flujo de cargas y cálculo de tensiones e intensidades en régimen permanente. Al ser una “*Blokset*” de Matlab/Simulink es posible realizar simulaciones desde la ventana de comandos de Matlab, y viceversa.

4.1.2 PowerWorld Simulator

Al igual que SimPowerSystem, presenta una interfaz gráfica de simulación. El principal módulo del que dispone realiza el flujo de cargas. Además, está complementado por otros 27 módulos adicionales que permiten realizar estudios de contingencias, despacho económico, estudios de cortocircuitos, etc.

PowerWorld concibe la evolución en el tiempo de un sistema eléctrico como una sucesión de estados estacionarios, por tanto, no es capaz de simular transitorios electromecánicos. Dicho simulador está orientado a resolver enormes sistemas de energía eléctrica en estado estacionario.

4.1.3 ATP-EMTP

El desarrollo de esta herramienta se realizó con fines de diseño, definición de parámetros eléctricos y especificaciones de equipos, para ello permite realizar cálculos electromagnéticos y electromecánicos.

Dentro de sus múltiples estudios destacan: análisis de protecciones y sistemas de control, transporte en continua, compensadores estáticos de reactiva, análisis de transitorios electromecánicos y de maniobra, coordinación de aislamiento, sobretensiones atmosféricas, etc.

Una de las características más importantes de esta herramienta de simulación es su habilidad para realizar análisis en régimen transitorio, analizando el periodo subtransitorio del mismo (en los primeros instantes).

4.1.4 PSCAD/EMTDC

Esta herramienta está formada por dos módulos, el PSCAD (interfaz gráfica para el usuario) y el EMTDC (simulador de los sistemas de energía eléctrica). En sus inicios, este software fue concebido para realizar análisis electromagnéticos en corriente continua, sin embargo, en la actualidad puede realizar multitud de estudios: análisis de generación distribuida, modelado de líneas y cables, análisis de transformadores saturables, máquinas rotativas, sistemas integrados, dispositivos electrónicos de potencia, etc.

Este software resuelve, tanto desde una perspectiva electromecánica como electromagnética, las ecuaciones diferenciales que modelan los distintos elementos del sistema. Otra característica por señalar es que el usuario puede definir sus propios modelos en otro lenguaje de programación (Fortran, C, o Matlab), encargándose la interfaz gráfica PSCAD de definir el icono y la ventana de entrada de datos.

4.1.5 OpenDSS

OpenDSS es una herramienta de simulación de carácter general, que opera en el dominio de la frecuencia. Posee múltiples modos de análisis, tales como: diario, anual, instantáneo, estudios de faltas, dinámico, flujo de armónicos, estudios Montecarlo, etc. Ha sido usado para multitud de aplicaciones, algunas de ellas son: análisis de generación distribuida, simulaciones de plantas eólicas, simulaciones de sistemas de protección, estimaciones de estado, análisis de armónicos, análisis de corrientes inducidas geomagnéticamente (GIC), estudios de planificación probabilísticos, y muchos más.

La interacción con el usuario es por medio de códigos y líneas de comandos en un lenguaje orientado a objetos, por tanto, debe aprenderse antes de poder usar dicho software.

Además, es necesario señalar que este software, en sus inicios, fue una herramienta de análisis de flujo de armónicos, y con el paso del tiempo fue evolucionando para resolver el problema del flujo de potencia. Dicha manera de abordar el diseño dotó al programa de una gran flexibilidad.

4.1.6 PSS/E

Esta herramienta está formada por dos módulos, el PSCAD (interfaz gráfica para el usuario) y el EMTDC (simulador de los sistemas de energía eléctrica). En sus inicios, este software fue concebido para realizar análisis electromagnéticos en corriente continua, sin embargo, en

la actualidad puede realizar multitud de estudios: análisis de generación distribuida, modelado de líneas y cables, análisis de transformadores saturables, máquinas rotativas, sistemas integrados, dispositivos electrónicos de potencia, etc.

Este software resuelve, tanto desde una perspectiva electromecánica como electromagnética, las ecuaciones diferenciales que modelan los distintos elementos del sistema. Otra característica por señalar es que el usuario puede definir sus propios modelos en otro lenguaje de programación (Fortran, C, o Matlab), encargándose la interfaz gráfica PSCAD de definir el icono y la ventana de entrada de datos.

Entre los súper de simulación específicos para sistemas de energía eléctrica, existen algunas diferencias que conviene tener en cuenta a la hora de realizar un determinado estudio.

En principio todas las aplicaciones descritas realizan el análisis de flujo de carga con gran eficiencia. Sin embargo, en el caso de SimPowerSystem, ATP-EMTP, PSCAD/EMTDC y OpenDSS; el cálculo del flujo de carga está orientado a trabajar en régimen transitorio. Para conseguir un análisis de carga de cientos de nudos y hacer un estudio del sistema en regímenes más distantes en el tiempo de sistemas más complejos Power World o PSS/E se convierten en las herramientas más apropiadas.

	<i>SimPower System</i>	<i>Power World</i>	<i>ATP-EMTP</i>	<i>PSCAD/EMTDC</i>	<i>OpenDSS</i>	<i>PSS/E</i>
Flujo de cargas	Sí	Sí	Sí	Sí	Sí	Sí
Cortocircuitos	Sí	Sí	Sí	Sí	No	Sí
Distorsión armónica	Sí	No	Sí	Sí	Sí	No
Estabilidad de ángulo	Sí	No	Sí	Sí	No	Sí
Regímenes transitorios electromagnéticos	Sí	No	Sí	Sí	sí	No
Régimen permanente	No	Sí	No	No	No	Sí

Tabla 4.1.6.1 Comparativa de los programas de simulación más visuales

4.2 PSS/E

4.2.1 Introducción

Power System Simulation for Engineering (PSS/E), se compone de un conjunto completo de programas para estudios de red de transmisión (en sistemas de energía y generación), analizando su rendimiento tanto en condiciones estables como dinámicas.

Actualmente, son utilizados dos métodos de simulación, uno para el análisis de estado estacionario y otro para simulaciones dinámicas. PSS/E se puede utilizar para facilitar los cálculos de una variedad de análisis, incluyendo:

- Flujos de carga y funciones de red relacionadas
- Flujos de carga óptimo
- Fallas equilibradas y desequilibradas
- Construcción equivalente de la red
- Simulación dinámica

PSS/E utiliza una interfaz gráfica de usuario comprometidos con la funcionalidad de análisis estático, incluye: flujo de carga, análisis de faltas, flujo óptimo de carga, equivalencia y estudios de desconexiones y conexiones.

Además del estudio estático y dinámico, PSS/E aporta al usuario una gran cantidad de programas auxiliares para la inserción y extracción, así como manipulación de datos durante una simulación. Es más, una de las premisas más básicas de PSS/E es que un ingeniero pueda obtener el mayor rendimiento de las herramientas computacionales al mantener un control estricto sobre su aplicación.

Flujos de carga (Power Flow)

Un estudio de los flujos de carga (también conocido como flujos de potencia), es una herramienta importante que envuelve un análisis numérico aplicado a un sistema de potencia. A diferencia del análisis tradicional de un circuito, el estudio del flujo de carga simplifica en gran medida la notación requerida para los diagramas de líneas; pasando a utilizar valores por unidad, dando así paso a un enfoque mejor en valores asociados a la corriente alterna (real, aparente, reactivo).

La importancia de los estudios de flujo de carga reside en que permiten la planificación y la expansión futura de sistemas de energía existentes e inexistentes. Además, se puede utilizar el estudio de un flujo de carga a determinar el diseño más eficiente de un sistema eléctrico.

La interfaz PSS/E admite una variedad de instalaciones interactivas, entre las que se incluyen:

- Introducción, modificación y eliminación de datos de red utilizando una hoja de cálculo.
- Creación de redes y diagramas de una línea.
- Análisis de estado estacionario (flujo de carga, análisis de fallas, flujo de energía óptimo, etc.).
- Presentación de resultados de análisis de estado estacionario.

Dinámica (Dynamics)

El programa de simulación dinámica incluye todas las funcionalidades del análisis transitivo, dinámico y de régimen permanente que se puedan aplicar a un circuito eléctrico. La interfaz de simulación dinámica está operada en un programa independiente de la interfaz visual de PSS/E. El propósito de análisis dinámico es facilitar el análisis de la estabilidad dinámica del sistema. Sin embargo, como trabajamos en régimen permanente no utilizaremos este apéndice del programa principal.

4.2.2 Creación de un caso

Vamos a explicar cómo se creó el archivo original identificar cuáles son las partes importantes para tener en cuenta luego ver cómo va a ser el análisis de este.

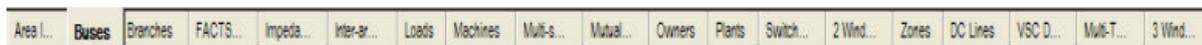
Para crear un nuevo archivo, hacemos clic en – – File → New. Se nos abrirá un pequeño cuadro de diálogo con varias opciones seleccionamos el “Network case”, y clicamos OK.

Esto abrirá un cuadro de diálogo *Build New Case*. Seleccionamos el MVA base como "100" en el cuadro "Base MVA". Luego habrá dos espacios en blanco correspondientes a Línea de rumbo 1 y línea de rumbo 2. Podemos dejarlos en blanco y hacer clic en OK.

Esto le llevará a la interfaz de hoja de cálculo, donde construiremos un nuevo caso. Guardamos el fichero como un archivo “.sav”, o “.raw” si queremos que no esté encriptado permitiéndonos analizar el fichero desde un editor de textos, por ejemplo.

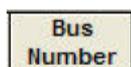
Ahora veríamos cómo sería acceder a un fichero cualquiera y analizar un modificar sus datos correspondientes para un posterior análisis de su simulación.

Una vez abierto el archivo “.sav”, hay 19 pestañas para elegir en la parte inferior de los datos (que se muestra a continuación). Se puede acceder a cada pestaña haciendo clic en esta. Hay seis pestañas que se centrarán en esta sección:

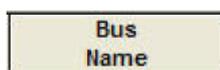


Buses

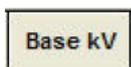
Parámetros importantes:



Es un número asignado al bus originalmente, puede estar entre 1 Y 999997.



Identificador alfanumérico del bus normalmente se pone la tensión del bus en el nombre identificarlo con mayor facilidad.



La base de la tensión del bus en Kv.

El código del tipo de bus qué va a ser.



1. Load bus (Bush de carga un punto de consumo).
2. Generator or plant bus (bus generación asociado a una central o varias).
3. Swing bus (solo hay un bus como este en todo el documento y es el que se utiliza para el análisis de cambio de variable del método de Newton o de Gauss-eidel).
4. Disconnected (isolated) bus.

Esto nos servirá después a la hora de clasificar las cargas y las centrales de generación para saber cuáles modificar y cuáles no tocar.

El Área se refiere a la zona o área de trabajo, nos ha servido en este caso estudiar coger solo los buses de la “ZONA V”.

Ramas



Todas las ramas de una red representada en PSS/E como “Branch”, es introducida leyendo una lista de las ramas de una red, es decir, las ramas no se generan automáticamente leyendo el bus de origen el bus de final, sino que se crean directamente. Sin embargo, Se produce una integración mixta, de tal forma que se introducen datos de la línea cómo resistencia y reactancia; o si no tipo de cable y temperatura hará que se encuentra; pero se asocia automáticamente a los buses de ida y llegada:

From Bus Number	From Bus Name
------------------------	----------------------

El número de bus y nombre del bus de ida.

To Bus Number	To Bus Name
----------------------	--------------------

El número de bus y nombre de bus de llegada.

Line R (pu)

Valor de la resistencia de la línea en valores por unidad.

Line X (pu)

Valor de la reactancia de la línea en valores por unidad.

Rate A (MVA)

Máximo de saturación de la línea en mega voltamperios (su potencia aparente).

Length

Longitud de la línea.

Cargas (Loads)



Estos son los consumidores de potencia de la red, no tienen por qué ir asociados los consumidores directamente con el bus, es decir, puede haber más de una carga por un bus década al igual que ocurrirá en generación:

Bus Number	Bus Name
-------------------	-----------------

Número del bus de la carga y el nombre.

Id

Identificador de cada carga dentro del mismo bus.

In-Service

Si el bus está en servicio o no, este es un valor binario.

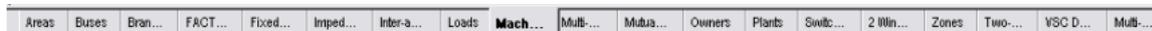
Pload (MW)

Potencia activa de consumo de la carga en megavatios.

Qload (MVAR)

Potencia reactiva de consumo de la carga en mega voltamperios reactivos.

Generadores (Machines)



Estos son los generadores de potencia de la red, el estudio supondremos que cambien mientras que las cargas se mantendrán constantes durante todo el tiempo:

Bus Number	Bus Name
------------	----------

Número del bus de generación y el nombre.

Id

Identificador de cada generador dentro del mismo bus.

In-Service

Si el bus está en servicio o no, este es un valor binario.

Pgen (MW)

Potencia activa generación de la carga en megavatios, recordemos que un generador está pensado solo para generar potencia usable, es decir, activa.

Pmin (MW)

Potencia activa mínima que debe proporcionar el generador.

Pmax (MW)

Potencia activa máxima que no debe sobrepasar el generador.

Qgen (MVAR)

Potencia reactiva que proporcionará el generador de forma involuntaria

Qmin (MVAR)

Potencia reactiva que llegará a proporcionar el generador.

Qmax (MVAR)

Potencia reactiva máxima que no debe sobrepasar el generador.

Una vez introducidos los datos de los parámetros, se guarda el caso con la opción *Save*. En PSS/E la información del caso puede guardarse en dos tipos de ficheros. Mediante la pestaña *Case data*, el fichero se puede guardar bajo extensión “.sav”, y mediante la pestaña *Power Flow Raw Data* permite guardarlo bajo extensión “.raw”.

En el caso de que se genere el caso a través de la ventana gráfica, lo primero es marcar la opción *Diagram*. A continuación, se selecciona la opción *Bind Items*, que permite que a medida que se añaden nuevos elementos en el diagrama unifilar se genere, en la ventana de datos *Spreadsheet View*, el registro de datos correspondiente a los nuevos elementos incorporados, con los valores por defecto de los distintos parámetros, que se pueden modificar directamente desde esa ventana.

Para añadir elementos no hay más que acudir a la Barra de Herramientas donde se encontrarán los distintos elementos y otras opciones relativas a diseño y estética del dibujo del esquema unifilar. Para introducir sus valores se pincha dos veces con el ratón sobre el elemento dibujado y aparece una ventana con las distintas opciones numéricas que ese elemento te permite introducir y modificar.

Si lo que se pretende es la construcción del diagrama unifilar a partir de un caso ya generado por introducción de datos en la ventana de comandos, existe un botón en la Barra de Herramientas denominado *Auto Draw*, en el que únicamente introduciendo el número de bus que se desea representar, te genera dicho bus y todos aquellos elementos que a él están conectados.

4.2.3 Simulación del flujo de cargas

El flujo de cargas persigue obtener las condiciones de funcionamiento en régimen permanente de un sistema eléctrico. De una forma más concreta, el flujo de cargas obtiene las tensiones e intensidades complejas en cada nudo de la red a partir del conocimiento de la potencia generada por los alternadores y de la consumida por cada carga.

Desde un punto de vista práctico, el flujo de cargas constituye una herramienta de gran utilidad para los técnicos responsables de la operación de los sistemas eléctricos, debido a que permite:

- Obtener un conocimiento previo del nivel de carga de las infraestructuras de transporte y distribución.
- Estudiar posibles situaciones de contingencia.
- Planificar.

Los flujos de carga posibilitan, además, la construcción de simuladores para el entrenamiento de los técnicos encargados de la operación del sistema eléctrico.

Desde una perspectiva metodológica su aplicación puede dividirse en dos etapas:

- Una primera dedicada a obtener las tensiones (módulo y fase) en cada uno de los nudos del sistema. Este primer paso no resulta en absoluto trivial, debido a que las cargas no se especifican en forma de impedancias complejas, sino en forma de potencias, lo que da lugar a sistemas de ecuaciones no lineales.

- Una segunda que permite calcular el resto de las magnitudes derivadas (P, Q, pérdidas, etc.) Este cálculo resulta inmediato una vez que se conocen las tensiones e intensidades en todos los nudos de la red.

Formulación del problema

El flujo de cargas se formula a partir de la matriz de admitancias del sistema, definida mediante los modelos de los diferentes elementos del sistema, sus interconexiones y el primer lema de Kirchhoff. Considérese el circuito de la Figura 4.2.3.1,

La ecuación del nudo 1 es: $\bar{I}_1 = \bar{I}_{1,0} + \bar{I}_{1,2} + \bar{I}_{1,3} + \bar{I}_{1,4}$

Para una red genérica de N nudos es posible establecer una relación matricial tal como la que se muestra en la Ecuación 4.2.3.1:

$$\begin{bmatrix} \bar{I}_1 \\ \bar{I}_2 \\ \dots \\ \bar{I}_N \end{bmatrix} = \begin{bmatrix} \bar{Y}_{1,1} & \bar{Y}_{1,2} & \dots & \bar{Y}_{1,N} \\ \bar{Y}_{2,1} & \bar{Y}_{2,2} & \dots & \bar{Y}_{2,N} \\ \dots & \dots & \dots & \dots \\ \bar{Y}_{N,1} & \bar{Y}_{N,2} & \dots & \bar{Y}_{N,N} \end{bmatrix} * \begin{bmatrix} \bar{V}_1 \\ V_2 \\ \dots \\ \bar{V}_N \end{bmatrix}$$

Ecuación 4.2.3.1

Donde,

$\bar{Y}_{k,k}$ Admitancia mutua del nudo k-ésimo. Se obtiene sumando las admitancias individuales conectadas al nudo k-ésimo, incluyendo también la admitancia que conecta dicho nudo con el de referencia.

$$\bar{Y}_{k,k} = \sum_{l=0 \neq k}^N \bar{y}_{k,l}$$

Ecuación 4.2.3.2

$\bar{Y}_{k,l}$ Admitancia entre los nudos k y l.

$$\bar{Y}_{k,l} = -\bar{y}_{k,l}$$

Ecuación 4.2.3.3

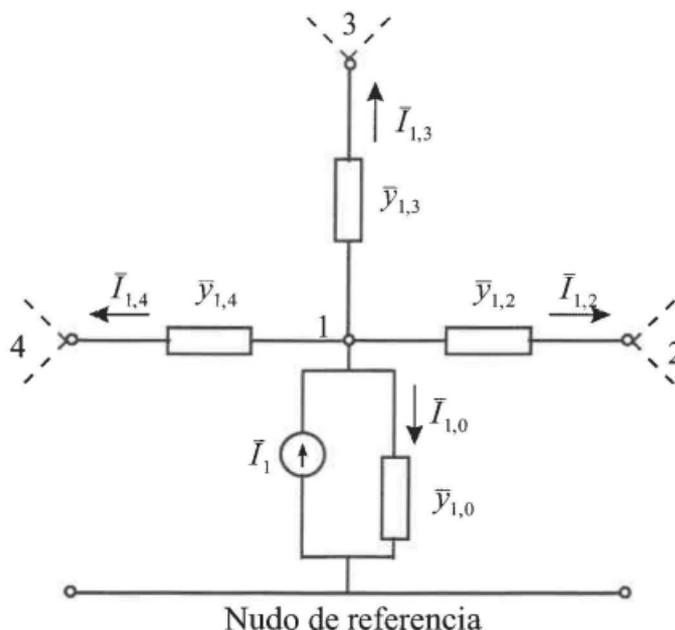


Figura 4.2.1 Figura 5. Nudo de una red genérica. (Fuente y adaptado de: [24])

En cada nudo k debe cumplirse que,

$$\bar{S}_k = \bar{S}_{Gk} - \bar{S}_{Ck} = \bar{V}_k * \bar{I}_k^*$$

Ecuación 4.2.3.4

Siendo \bar{S}_k la potencia aparente compleja neta inyectada en el nudo k, obtenida como diferencia entre la potencia generada y la consumida por la carga en dicho nudo. La expresión anterior puede generalizarse a toda la red, de forma que,

$$[\bar{S}] = diag\{[\bar{V}]\}[\bar{I}]^* = diag\{[\bar{V}]\}\{[\bar{Y}][\bar{V}]\}^*$$

$$[P] + j[Q] = diag\{[\bar{V}]\}([G] - j[B]) * [\bar{V}]^*$$

Ecuación 4.2.3.5

Los métodos utilizados habitualmente para resolver la ecuación anterior se basan en técnicas que convergen hacia la solución a partir de la derivada, lo que obliga a utilizar magnitudes reales. La ecuación (11) puede separarse su parte real e imaginaria. Así, para el nudo k-ésimo,

$$P_k = V_k \sum_{l=1}^N V_l * (G_{k,l} * \cos \theta_{k,l} + B_{k,l} * \sin \theta_{k,l})$$

Ecuación 4.2.3.6

$$Q_k = V_k \sum_{l=1}^N V_l * (G_{k,l} * \sin \theta_{k,l} - B_{k,l} * \cos \theta_{k,l})$$

Ecuación 4.2.3.7

Cada nudo aporta dos ecuaciones y cuatro incógnitas, por lo que es necesario definir dos magnitudes por nudo para poder resolver las ecuaciones anteriores. En función de la información disponible para cada nudo es posible establecer dos grandes grupos de nudos:

- **Nudos de consumo o nudos PQ.** Se trata de nudos donde se conocen los valores de las potencias P y Q consumidas, siendo cero las potencias generadas, tanto P como Q. Este tipo de nudos se corresponden con los nudos donde se conectan cargas.
- **Nudos de generación o nudos PV.** Se trata de nudos donde se conoce la potencia activa P y la tensión V del nudo. Este tipo de nudos se corresponden con los nudos donde están conectados los generadores.

La clasificación anterior presenta el inconveniente de conocer de antemano la potencia generada por los nudos generadores. Dado que no suele ser un dato conocido, lo que se hace habitualmente es establecer una potencia para todos los generadores excepto uno que se utiliza como referencia, para el que se establece el origen de fases. La solución se obtiene mediante un proceso iterativo, en el que se van corrigiendo los valores individuales fijados como punto de partida para la solución. El nudo de referencia que se fija como origen de fases se conoce en la literatura anglosajona como nudo “slack” y suele tratarse de un nudo de generación importante, con capacidad para regular frecuencia. En algunos casos este nudo se ubica en la frontera del sistema, sin que exista físicamente en dicho punto un generador.

Suponiendo que la red tiene N nudos, es posible escribir que,

$$N = N_g + N_c + 1$$

Ecuación 4.2.3.8

Donde,

N_g Número de nudos generadores (PV).

N_c Número de nudos consumidores (PQ).

En el nudo de carga o nudo PQ, se conocen la potencia activa y reactiva inyectada en el nudo, puesto que en el caso de que hubiera generación se mantendría constante y, en consecuencia, se trabajaría con la potencia inyectada resultante.

En el nudo de generación o nudo PV, se conocen la potencia activa inyectada y el módulo de tensión de dicho nudo.

El nudo de referencia PR, es un nudo que, como su nombre indica, se toma como referencia, dejando las potencias activa y reactiva inyectadas como incógnitas, dado que si todos los nudos fueran o bien de carga PQ o bien de generación PV, desde el principio ya se habría establecido una situación de equilibrio entre las potencias activas del sistema, cuestión que establecería una única situación y además imposibilitaría una adaptación o cambio de las demás variables.

Por tanto, dado que la búsqueda del equilibrio depende de la situación de partida, no pudiendo figurar todos los valores como dato, se deja algún nudo generador sin definir su potencia activa inyectada, esta será una incógnita iterativa sobre la que calcular, y el nudo o Bus será un “Bus swing” (de balanceo).

De esta forma, el problema se formula de forma específica según el sistema de ecuaciones siguiente,

$$P_k = V_k \sum_{l=1}^N V_l * (G_{k,l} * \cos \theta_{k,l} + B_{k,l} * \sin \theta_{k,l}) \quad k = 1, 2, \dots, N_g + N_c$$

Ecuación 4.2.3.9

$$Q_k = V_k \sum_{l=1}^N V_l * (G_{k,l} * \sin \theta_{k,l} - B_{k,l} * \cos \theta_{k,l}) \quad k = 1, 2, \dots, N_g + N_c$$

Ecuación 4.2.3.10

En la tabla 1 se resumen las condiciones del problema en términos de ecuaciones e incógnita.

<i>Ecuaciones</i>	<i>No. ecuaciones</i>	<i>Incógnitas</i>	<i>No. Incógnitas</i>
	$2N_c + N_g$	$\theta_k \quad k = 1, 2, \dots, N_c + N_g$ $\ V_k\ \quad k = 1, 2, \dots, N_c$	$2N_c + N_g$

Tabla 4.2.3.1 condiciones del problema en términos de ecuaciones e incógnita

Nótese que al fijar la tensión del nudo “slack” y dejar indeterminada su potencia no se modifica el conjunto de ecuaciones anteriores, ya que las ecuaciones de dicho nudo no intervienen en la resolución del conjunto. Estas ecuaciones podrán ser calculadas una vez resuelto el sistema de ecuaciones anteriores.

Dado que el conjunto de ecuaciones anteriores es no lineal, será necesario utilizar métodos iterativos que se evalúan a partir de una condición inicial. Normalmente se utiliza como condición inicial un perfil plano, que consiste en asumir $\theta_k^0 = 0$ para todos los nudos $N_c + N_g$ y $\|V_k^0\| = 1 pu$ para los nudos de consumo N_c .

4.2.4 Método de Gauss-Seidel

La resolución de los problemas de flujos de potencia mediante programación sigue un proceso iterativo.

El proceso iterativo consiste en calcular la magnitud de las variables desconocidas de los sucesivos nudos mediante la asignación de valores estimados a aquellos datos desconocidos y utilizando el resto de los valores especificados. Las pautas por desarrollar son las siguientes:

- A partir de la clasificación de los nudos eléctricos según su tipología en PR, PQ y PV, se establece un orden para los respectivos nudos y se asignan las correspondientes estimaciones a los datos desconocidos.
- Sin incluir el nudo de referencia (PR), se realiza la iteración por cada nudo donde se desconozca un valor determinado con las estimaciones consideradas inicialmente. De exista forma, en cada nudo se obtienen una nueva magnitud del citado valor.
- Se efectúa la interacción de los sucesivos nudos sustituyendo progresivamente los valores anteriormente obtenidos.

De esta forma al pasar por todos los nudos, excepto por el de referencia (PR), se ha obtenido un nuevo conjunto de valores en un primer ciclo iterativo o iteración del sistema. Cabe destacar en el caso de que se incumpla algún tipo de restricción se actuará en consecuencia, cambiando la tipología de estudio y fijando los valores del límite superado.

Con este nuevo conjunto de valores se procede a realizar una iteración del sistema repitiéndose hasta que los cambios experimentados en los datos obtenidos de los sucesivos ciclos iterativos para cada nudo sean insignificantes o menores que un valor mínimo especificado, en este caso el número de interacciones deseadas.

Si se tiene un sistema eléctrico de n nudos donde existen nudos de carga, PQ, nudos de generación, PV y un nudo de referencia, PR. Dado que para el nudo PR no se realiza la interacción, para obtenerlas las $(n-1)$ ecuaciones de las tensiones de los nudos se procede de la siguiente forma:

- Se ópera de la misma forma que para la situación anterior hasta que en la operativa de cálculo de las tensiones de los nudos correspondiente a la primera iteración si llegue la tensión incógnita de un nudo PV.
- Se calcula la potencia reactiva inyectada perteneciente al nudo PV, para mantener el valor del módulo de la tensión propuesto, mediante los valores más actuales de las pensiones obtenidas con anterioridad utilizando la expresión (Ecuación 4.2.4.1) Qué representa la parte imaginaria de la potencia aparente del nudo k . Se sustituyen los valores más actuales obtenidos de la 1ª iteración (Ecuación 4.2.4.2) y aquellos aún estimados otra vez.

$$Q_k = -Im \left\{ \bar{V}_k^* * \left(\sum_{l=1}^N \bar{Y}_{kl} * \bar{V}_l \right) \right\} \quad 7.11$$

Ecuación 4.2.4.1

$$Q_k^1 = -Im \left\{ \bar{V}_k^* * \left(\sum_{l=1}^{i-1} \bar{Y}_{kl} * \bar{V}_l^1 + \sum_{l=1}^N \bar{Y}_{kl} * \bar{V}_l \right) \right\} \quad 7.12$$

Ecuación 4.2.4.2

De esta forma se obtiene un valor de potencia reactiva inyectada correspondiente al nudo k , (Q_k^1) cuya magnitud deberá compararse con las restricciones establecidas en el sistema eléctrico a estudio pudiendo suceder los siguientes casos:

En el caso de que se incumplan dichas restricciones el nudo pasaría a comportarse como un nudo de carga, PQ, pero esta integración del sistema, (con la potencia reactiva límite establecida) obligaría a que se operara en consecuencia a la nueva naturalidad del nudo, calculando el valor de la tensión correspondiente a esta nueva situación. En el caso de que existieran márgenes superior e inferior del valor de la potencia reactiva, el valor seleccionado para el nuevo nudo PQ sería el correspondiente al del límite incumplido.

La sucesiva iteración del sistema, el nudo volvería a tratarse cómo PV como el valor originalmente especificado del módulo de la tensión y controlando nuevamente el cumplimiento de los márgenes de potencia reactiva establecidos. Por pudiera darse el caso de que la siguiente iteración con valores más terceros de las variables del resto de los nudos se posibilita así la potencia reactiva del nudo PV dentro de los márgenes establecidos. En caso contrario, se procedería de la misma forma que se hizo anteriormente, convirtiendo el nudo PV en nudo PQ, calculando el nuevo valor de la tensión y utilizándolo durante el correspondiente ciclo iterativo.

Durante el proceso iterativo hasta su finalización, mediante la comprobación del cumplimiento de los límites establecidos coma se decide el mantenimiento del nudo de generación como nudo PV o su cambio definitivo por uno de carga, PQ.

En el caso de que la potencia reactiva inyectadas encontrarse dentro de los límites permitidos se procedería a calcular el argumento de la tensión mediante la expresión (Ecuación 4.2.4.3) con los correspondientes valores obtenidos y estimados (mención especial para el ángulo de la tensión \bar{V}_k en la primera iteración a la hora de obtener su valor conjugado).

$$\delta_k^1 = Arg \left\{ \frac{1}{\bar{Y}_k} \left[\frac{P_k - jQ_k^1}{\bar{V}_k^*} - \left(\sum_{l=1}^{i-1} \bar{Y}_{kl} * \bar{V}_l^1 + \sum_{l=1+1}^N \bar{Y}_{kl} * \bar{V}_l \right) \right] \right\}$$

Ecuación 4.2.4.3

De esta forma ya se dispone del valor de la tensión el nudo K, tanto en módulo como argumento, esto posibilita la continuidad del método de resolución pasando al nudo siguiente hasta llegar a la última iteración.

Este proceder se realizará sucesivamente hasta completar el número de iteraciones o hasta que los valores obtenidos experimenten cambios insignificantes, es decir, que no se pueda aumentar la precisión siendo esta el error de la máquina o una encontrada dentro de los márgenes previamente establecidos.

4.2.5 Herramientas de simulación

Las herramientas de simulación disponibles pueden clasificarse atendiendo a varios criterios. Una clasificación lógica, desde un punto de vista técnico consistiría en agrupar las distintas herramientas en base a su dominio de trabajo. Así, encontraríamos herramientas que trabajan en el dominio del tiempo y de la frecuencia. Otro criterio de clasificación podría basarse en su carácter: dominio público o herramienta comercial. A continuación, se revisan algunas

herramientas comerciales ampliamente aceptadas por la comunidad científica internacional, y se resumen en la tabla II otras herramientas disponibles de forma gratuita en Internet para uso educativo.

Herramientas	Descripción
<i>PowerWorld Simulator</i>	Programa para la simulación de flujos de carga y despacho económico. La versión gratuita permite simular redes de hasta 12 nudos.
<i>PFLOW</i>	Programa para el cálculo de bifurcaciones locales y singularidades en el Jacobiano del sistema. Desarrollado por el profesor Claudia Cañizares de la universidad de Waterloo.
<i>PCFLO y PCFLOH</i>	Programa para la simulación de flujos de carga convencionales y poco comunes, así como estudios de cortocircuito. Desarrollado por el profesor Mack Grady de la universidad de Texas en Austin.
<i>Matlab Power Flow Program</i>	Programa desarrollado en Matlab 4.0 para la simulación de flujos de carga. Muy didáctico y con mensajes en castellano. Desarrollado por Marcelino Madrigal, estudiante de doctorado de la universidad de Waterloo.
<i>NETDRAW y LAKU</i>	Programas para la construcción de esquemas de redes eléctricas y la simulación de flujos de carga y cortocircuitos. Desarrollado por el Dr. Hans-Detlef Pannhorst.
<i>Electrical machines and drives toolbox</i>	Programa desarrollado en Matlab-Simulink para la simulación de máquinas eléctricas. Desarrollado por el profesor Mahmoud Riaz de la universidad de Minnesota.
<i>Power Education Toolbox</i>	Toolbox orientado hacia la simulación de flujos de carga, la estimación de estados y el profesor Ali Abur de la universidad de Texas A&M.

Tabla 4.2.5.1 Comparativa de los programas de simulación más eficientes

Es necesario destacar que ya se habló en 4.1 Software a cerca de los programas más comunes usados para la simulación de circuitos, pero este apartado se refería a aquellos con un entorno visual más cómodo y una curva de aprendizaje menor siendo de uso más intuitivo. Estas son herramientas específicamente diseñadas para la simulación de máquinas eléctricas en sistemas más reales, por lo que se encuentran agrupadas en un entorno de uso más concreto y complejo.

4.3 Python

4.3.1 Introducción

Se trata de un lenguaje de programación con existencia desde 1996 pero con un gran crecimiento de popularidad en los últimos años debido a su fácil e intuitiva forma de estructurar el código.

Su estructuración es a color como otros códigos, pero obligando a tabular cada vez que se habrá un bucle esto obliga y hace a la vez que la visión del programa en lenguaje máquina sean mucho más clara y rápida que en otros sistemas.

Por otro lado, es un lenguaje que integra funciones de toda clase. Es capaz de tratar un gran número de archivos, y leer y guardar en toda clase ficheros (".sav", ".raw", ".xlsx", ".m"...). Si nos quedamos sin herramientas podemos recurrir a un sinfín de librerías con programas ya compilados. Su forma de ejecutarse es eficiente y amplia en lo que se refiere a cantidad de compañías que se aprovechan de Python para facilitar un uso más complejo de sus programas sin poner en riesgo la seguridad del código fuente.

4.3.2 Python aplicado al flujo de cargas con PSS/E

En este proyecto nos aprovechamos de Python para extraer toda la información que nos interesa del archivo original en formato ".sav" de PSS/E; almacenarla en una serie de tuplas (listas) y más tarde trabajar en función de esos datos obtenidos.

Por otro lado, también somos capaces de utilizar la conexión entre Python y PSS/E, para hacer simulaciones del funcionamiento del circuito sin tener que recurrir a un entorno gráfico qué siendo más visual y ameno puede hacerse más lento a la hora de ejecutar una rutina reiterativa.

En nuestro caso, esta rutina será la modificación de la reactancia de una línea con la siguiente simulación del sistema mira repetición del proceso hasta obtener una saturación de la línea menor a la deseada.

En este apartado del proyecto analizamos solamente esa conexión entre Python y PSS/E. Las instrucciones utilizadas en orden de ejecución son las siguientes:

```
import sys                                #Importamos la librería "sys" de sistema
sys.path.append("C:\Program Files (x86)\PTI\PSSE33\PSSBIN")
#Le decimos a Python donde está el path, es decir, el directorio con el programa de PSS/E
```

```
import os                                  #Importamos la librería "os" de operative system
os.environ['PATH']+=";C:\Program Files (x86)\PTI\PSSE33\PSSBIN"
#Le decimos a Python donde están las librerías que necesitamos de PSS/E
```

```
import psspy                              #Importamos la librería "psspy" de pss-python librery
import redirect                            #Importamos la librería "redirect" de redireccionamiento
redirect.psse2py()                         #Usamos la función "psse2py" de la librería "redirect"
psspy.psseinit(100000)
_i=psspy.getdefaultint()
#Obtén los valores por defecto de intgar de una máquina (o sea, los valores enteros)
_f=psspy.getdefaultreal()
#Obtén los valores por defecto de real de una máquina (o sea, los valores reales)
_s=psspy.getdefaultchar()
#Obtén los valores por defecto de char de una máquina (o sea, los valores característicos)
```

```
import xlrd
import xlwt
import string
import pyodbc
import random
from time import clock #Sirve para importar un reloj
```

```
CASOraw= r""" C:\Users\Roy\Desktop\Copia_seguridad_tfgvictor\TFG\Experimentos\MI
TFGTFG.raw """
#Indicamos el directorio en el que se encuentra el archivo ".raw" (código legible por el editor
de block de notas porque no está encriptado).
```

```
CASOsav= r""" C:\Users\Roy\Desktop\Copia_seguridad_tfgvictor\TFG\Experimentos\MI
TFG\TFG.sav"""
#Indicamos el directorio en el que se encuentra el archivo ".sav" (ilegible por cualquier editor
de textos porque está encriptado).
```

```
psspy.case(CASOsav);
```

```
#Crea un nuevo caso de trabajo en la RAM sobre el que trabajar a partir de la lectura anterior.
```

```
psspy.fnsi([0,0,0,1,1,1,99,0]);
```

```
#Hazme una simulación utilizando el método FNSL (por Newton-Rhampson).
```

```
psspy.save(CASOsav);
```

```
#Guárdame el caso de trabajo en la RAM para trabajar con él.
```

```
psspy. bsys(
```

```
  sid=1,          # Número del subsistema a generar (1).
```

```
  usekv=1,       # Base de tensión en kV.
```

```
  basekv=[132,132], # Coge los buses con tensión 132kV.
```

```
  numarea=10,
```

```
  areas=[10],    # Solo los buses del área 10, que es el área de "ZONA V".
```

```
) #Creamos un subsistema, (un vector), de los buses correspondientes al área 10.
```

```
ierr, iarray = psspy.abusint(
```

```
  sid=1,          # Número del subsistema del que extraer datos.
```

```
  flag=1,        # Solo los buses en servicio.
```

```
  string="NUMBER" # Pedimos el número de los buses.
```

```
) # Obtenemos los números de los buses en servicio del subsistema generado nº 1.
```

```
BUSV=iarray[0]
```

```
#El primer valor de iarray es la lista buscada (en python el primer valor de una lista ocupa la posición 0).
```

```
ierr, iMbus = psspy.amachint(
```

```
  sid=1, # Número del subsistema del que extraer datos.
```

```
  flag=1, # Solo los buses en servicio de plantas electricas en servicio.
```

```
  string=['NUMBER'] # Pedimos el número de los buses.
```

```
) #Creamos un vector de los buses generadores.
```

```
GENBUSV=iMbus[0]
```

```
ierr, cMids = psspy.amachchar(
```

```
sid=1,      # Número del subsistema del que extraer datos.
string=['ID','EXNAME'] # Pedimos las IDS y los Nombres.
) #Creamos una matriz de las IDS y Nombres de los buses generadores.

IDGENBUSV=cMids[0]
GENBUSNAMEV=cMids[1]

ierr, branchesfrmtobus = psspy.abrnint(
    sid=1, # Número del subsistema a generar.
    ties=3, # Cogemos todas las ramas del sistema.
    flag=1, # Solo las ramas en servicio.
    string=["FROMNUMBER", "TONUMBER"]
) #Creamos una matriz con el bus de origen y de final de cada rama del circuito.
branchesfrmbus=branchesfrmtobus[0]
branchestobus=branchesfrmtobus[1]

ierr, branchesidfrmtname = psspy.abrnchar(
    sid=1, # Número del subsistema del que extraer datos.
    ties=3, # Cogemos todas las ramas del sistema.
    flag=1, # Solo las ramas en servicio.
    string=["ID","FROMEXNAME","TOEXNAME"]
) #Creamos una matriz con la ID, el nombre del bus de origen y el del final de cada rama del
circuito.

psspy.macdat(GENBUSV[i],IDGENBUSV[i],'P')[1]
#Obtenemos la potencia de un generador proporcionando su nº de bus y su ID dentro del bus
(si hay más de uno).

psspy.macdat(GENBUSV[i],IDGENBUSV[i],'PMAX')[1]
#Obtenemos la potencia máxima de un generador proporcionando su nº de bus y su ID dentro
del bus (si hay más de uno).

psspy.macdat(GENBUSV[i],IDGENBUSV[i],'PMIN')[1]
#Obtenemos la potencia mínima de un generador proporcionando su nº de bus y su ID dentro
del bus (si hay más de uno).

psspy.machine_data_2(GENBUSVTER[p],'1',[_i],[NEWSPOTGENBUSVTER[p]])
```

#Modificación de la potencia de los generadores térmicos de número GENBUSVTER[p] por la nueva potencia NEWPOTGENBUSVTER[p].

```
sspy.brnmsc(ibus=rama[0], jbus=rama[1], ickt=rama[2], string="PCTMVA")[1]
```

Obtención de la PCTMVA de una rama, es decir, su porcentaje de saturación.

```
valores=psspy.brndt2(ibus=frmBusramavip, jbus=toBusramavip, ickt=cktramavip, string="RX")[1]
```

Extraemos resistencia y reactancia de una rama o línea.

```
oldresis=valores.real # Valor de la resistencia de la línea más saturada.
```

```
oldreac=valores.imag # Valor de la reactancia de la línea más saturada.
```

```
psspy.branch_chng(i=frmBusramavip, j=toBusramavip, cktramavip="1", intgar=None, realar=[newresis,newreac])
```

Modificamos la reactancia de la línea.

4.4 SQL

4.4.1 Introducción

SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos. Los programas de bases de datos relacionales, como Microsoft Office Access, usan SQL para trabajar con datos. A diferencia de muchos lenguajes de computación, SQL no es difícil de leer y entender, incluso para un usuario inexperto, y al igual que un selecto grupo de lenguajes de computación, SQL es un estándar internacional reconocido por organismos de estándares como ISO y ANSI.

SQL se usa para describir conjuntos de datos que pueden ayudarle a responder preguntas. Al usar SQL, debe usar la sintaxis correcta. La sintaxis es el conjunto de reglas mediante las que se combinan correctamente los elementos de un idioma. La sintaxis SQL se basa en la sintaxis del idioma inglés y usa muchos de los mismos elementos que la sintaxis de Visual Basic para Aplicaciones (VBA).

Recurrimos al uso de SQL en lugar de Excel, por ejemplo, porque se trata de un programa de gestión de datos de gran capacidad. Nosotros en Excel podemos editar datos generales funciones o hacer cálculos en función de éstos; sin embargo, si lo único que queremos es almacenar tales datos y además almacenar una gran cantidad de datos Excel falla debido a

qué está pensado para ofrecer una gran cantidad de tareas, pero con una mínima cantidad de datos, con mínima hablamos por debajo de los 1000 ingresos.

Por otro lado, SQL no posee tantas funciones integradas ni permite crear nuevas si no se tiene un amplio conocimiento del lenguaje máquina; Pero es esa falta de personalización la que la que garantiza una mayor eficiencia del programa dejándolos así trabajar con cientos de miles de datos sin que este se colapse. Así que en el caso de investigación y análisis SQL es el compañero perfecto de cualquier programa de simulación.

4.4.2 Python relacionado con SQL

Sencillamente cogemos las tablas generadas a partir del proyecto original guardadas en la RAM dentro de la simulación de Python, y las guardamos en un proyecto SQL situado en el mismo ordenador en el que se han hecho las simulaciones. SQL nos permitiría por ejemplo guardar los datos en cualquier otro sitio un servidor por ejemplo qué podríamos tener funcionando todo el día al que podemos acceder desde cualquier momento convirtiéndose en una base de datos para páginas web o análisis de otros investigadores.

En orden de uso Estas son las líneas de código referidas a SQL dentro del programa escrito en Python:

```
cnxn = pyodbc.connect(driver='{SQL Server}', server='DIEEPC7\PQS',
database='TFGVICTOR', uid='usuario', pwd='****')
```

#Dentro de la librería pyodbc ya importada recurrimos a la función “connect”, esta conectará Python con un SQL Server, el “DIEEPC7\PQS”. Ahí nos situamos en el fichero “TFGVICTOR” y accedemos con el usuario “usuario” y contraseña “****”.

```
com=cnxn.cursor() #Situamos esa conexión creada como base de SQL dentro de Python en esta ejecución.
```

```
com.execute("DELETE FROM [TFGVICTOR].[dbo].[Potencias]")
```

```
com.commit()
```

```
com.execute("DELETE FROM [TFGVICTOR].[dbo].[Casos]")
```

```
com.commit()
```

```
com.execute("DELETE FROM [TFGVICTOR].[dbo].[Resultados]")
```

```
com.commit()
```

```
# Reiniciamos las tablas con los datos antiguos
```

p=0

for p in range(len(GENBUSV)):

```
st="INSERT INTO [TFGVICTOR].[dbo].[Potencias] (GENBUSV,GENBUSNAMEV,  
POT_Normal,POTMIN,POTMAX) VALUES (" +str(GENBUSV[p])+",  
"+str(GENBUSNAMEV[p])+", "+str(POTGENBUSV[p])+", "+str(POTMINGENBUSV[  
p])+", "+str(POTMAXGENBUSV[p])+")"
```

com.execute(st) # Insertame el stament, es decir, ejecuta la función creada en la línea anterior y asociada a "st".

com.commit() # Guardamos el cambio hecho, este comando no se usa normalmente, pero en Python es necesario porque daría un error por deja un proceso abierto.

Este bucle sirve para recorrer todos los componentes de una lista GENBUSV e ir introduciéndolos en la tabla SQL.

El resto de los comandos consisten en introducir valores igualmente, pero en tablas con columnas y nombres distintos, este procesó explicará con mayor profundidad en el [apartado 6.1.2](#).

5 Resultados

5.1 Análisis y representación gráfica del índice de carga de las líneas ante variaciones de generación

Este apartado se centra en la parte principal del estudio de trabajo; vamos a estudiar la zona V. esta Contiene una serie de grupos de generación (máquinas generadoras) y consumidores.

La clave del proyecto radica en obtener para cualquier tipo de caso de generación las ramas más saturada en cada uno de ellos; y posteriormente, saber que reactancia habría sido necesaria en cada caso para que esa rama alcanzada una saturación máxima del 80 por ciento

La mejor descripción del funcionamiento del programa para llevar a cabo esta simulación de todo el proceso pensado se apoyará en el siguiente flujo grama de trabajo:

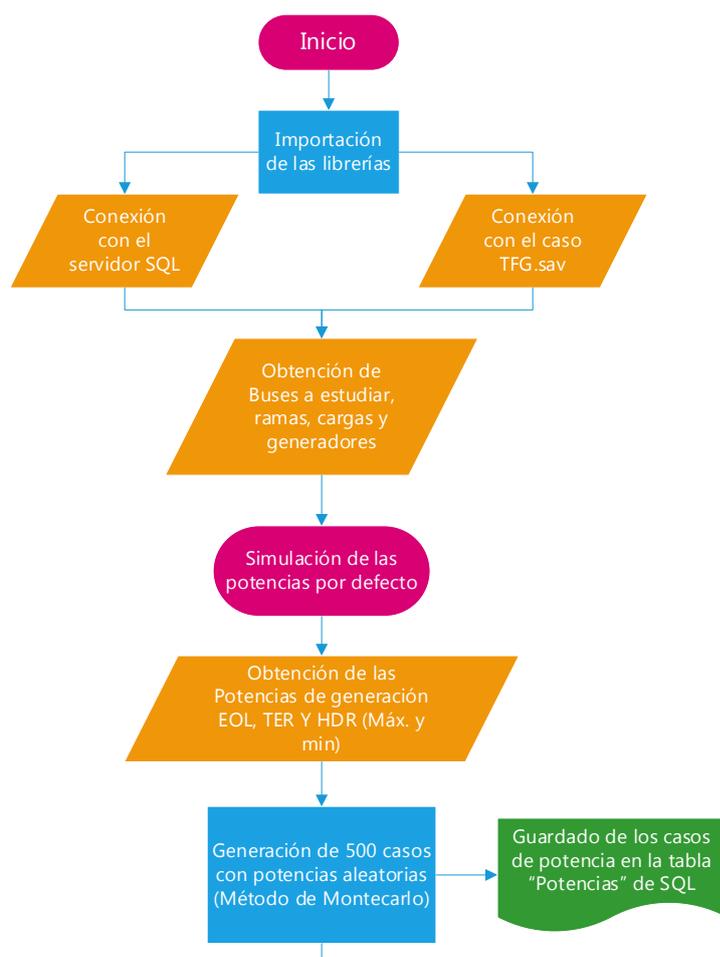


Figura 5.1.1 Flujo grama de funcionamiento del programa de estudio 1.

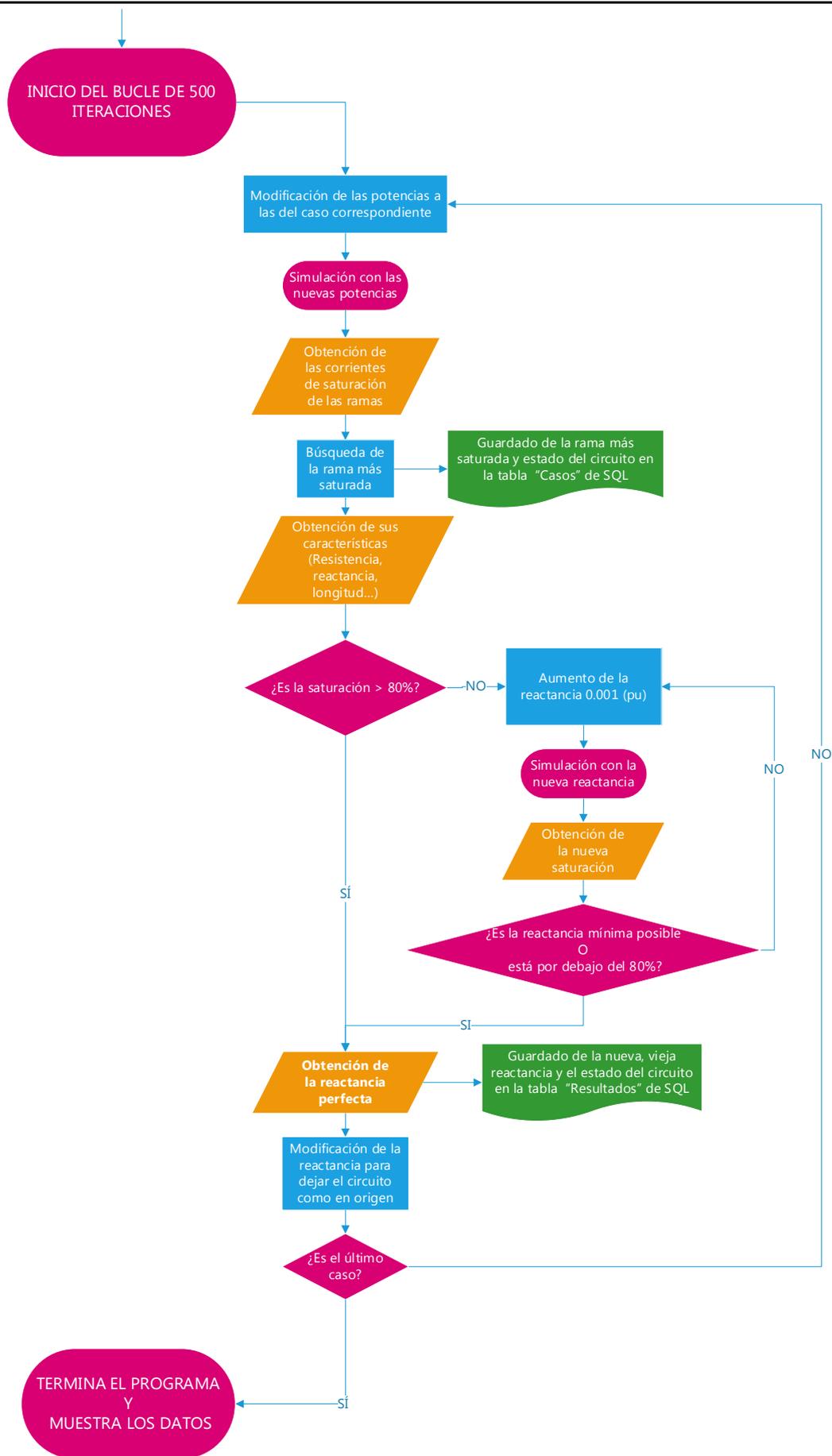


Figura 5.1.2 Flujoograma de funcionamiento del programa de estudio 2.

5.2 Tablas

5.2.1 Potencias

Esta tabla es la única que se generó una sola vez, osase, es independiente de los casos ya que simplemente reflejan las potencias a las que pueden llegar a funcionar las centrales de generación de la red estudiada.

- **GENBUSV:** Número de bus correspondiente a los buses de generación de la ZONA V que serán analizados en este caso. cabe destacar que para una mejor simplificación del estudio debido a que hacía exponencialmente complejo la obtención de potencias hemos asociado todas las máquinas de generación entre forma que corresponde a un bus de generación a una planta de generación teniendo así todas un ID de “1”.
- **GENBUSNAMEV:** Nombre completó (18 caracteres) de los buses correspondientes a la zona V.
- **POT_Normal:** potencia en megavatios ahora que está funcionando, pero central en el caso por defecto (TFG.sav), es decir, un caso de un día de generación normal.
- **POTMIN:** potencia mínima a la que puede o debe funcionar la máquina de generación.
- **POTMAX:** potencia máxima a la que puede o debe funcionar la máquina de generación.

<i>GENBUSV</i>	<i>GENBUSNAMEV</i>	<i>POT_Normal</i>	<i>POTMIN</i>	<i>POTMAX</i>
946	*****	8.000011444	0	38
905	*****	10	0	76.98000336
940	*****	20.15419769	0	58.39999771
219	*****	30.48718262	0	93.59999847
918	*****	20.44179916	0	33
949	*****	33.08995438	0	35
928	*****	43.66582489	0	44.20000076
948	*****	23.63568115	0	25
412	*****	43.22558212	0	49.5
950	*****	81.30674744	0	86
1050	*****	96.2519989	0	258.4299927
1055	*****	65.20237732	0	88

945	*****	66.16159821	0	75.90000153
944	*****	26.47196007	0	77.30000305
1501	*****	17.6430912	0	21.12000084
1201	*****	29.99326324	0	34.5
416	*****	22.38902283	0	22.5
1080	*****	320	0	327
909	*****	73.88140106	0	85.43000031
1101	*****	13.23232079	0	15.00000095
903	*****	56.72594452	0	60.00000381
910	*****	64.70839691	0	72.80000305
911	*****	70.33951569	0	45.75
418	*****	29.11110497	0	33.75
931	*****	41.08729935	0	42.41999817
1401	*****	59.98659897	0	70.41000366
109	*****	142.2597046	0	160
417	*****	18.16351128	0	21
951	*****	14.18141174	0	15.00000095
942	*****	118.7472916	0	120.2000046
411	*****	25.31776428	0	30.36000061
1301	*****	79.3939209	0	104.4599991
220	*****	87.12134552	0	36.54999924
1514	*****	103.7577057	0	134
414	*****	54.69358063	0	62.5

Tabla 5.2.1.1 Potencias de las centrales generadoras estudiadas

Modificamos las potencias de 35 centrales de generación agrupándolas en tres grupos por tema de organización y gasto computacional. Normalmente se suelen agrupar las centrales por sus características; si existe una gran cantidad de viento en toda la zona norte de la península la mayoría de las centrales eólicas estarán funcionando a pleno rendimiento, las térmicas son centrales base difíciles de hacer que dejen de producir y que saldría poco rentable desconectarlas, y las hidroeléctricas sería en el comodín para elegir cuándo usarlas o no. sí han estado saturadas de agua entonces sí que se tendrían que abrir, pero son las calibradoras de un sistema energético muy complejo.

En tema computacional se escogieron tres grupos por la incrementación exponencial del número de casos posibles a simular, si hubiéramos hecho una variación aleatoria para cada una de las 35 centrales de generación, el incremento de número de casos a simular para

tener un reflejo real de todas las simulaciones posibles (sin utilizar número aleatorios ni el método de Montecarlo), habría ascendido a $35^{n^{\text{casos}}} = 35^{100}$. Y utilizando el método de Montecarlo tendríamos que haber hecho por equivalencia $35^{20} = 7.60 * 10^{30}$ simulaciones.

Esto habría supuesto que en vez de tardar 2 horas en hacer la simulación de todos los casos por el método de Montecarlo (que ya nos había ahorrado 4 días de uso computacional), habríamos pasado a gastar 1 año de simulaciones para obtener un resultado muy parecido.

5.2.2 Casos

Esta es una tabla complementaria a la de resultados para saber exactamente cómo están funcionando los 3 conjuntos de centrales de generación en el caso deseado de estudio:

- **CASO:** Pone el número del caso que estamos simulando.
- **Rama_saturada_NAME:** el nombre que recibe la rama está compuesto por el nombre del bus de ida eliminándole la tensión que se incluye con el nombre de éste el programa (132.0 Kv), el bus de llegada y entre paréntesis le ID de la rama para saber si tiene alguna otra en paralelo.
- **Rama_saturada_BUS:** El número de bus de ida sido el número de bus de llegada y el ID de la rama.
- **Sat_rama(%):** Saturación de la rama más saturado en el caso estudiado, debe coincidir con **Sat_rama_pre reac(%):** de la tabla de “Resultados”.
- **Térmica(%):** Porcentaje sobre la diferencia entre la potencia máxima la potencia mínima al que están funcionando las centrales de generación térmicas.
- **Eólica(%):** Porcentaje sobre la diferencia entre la potencia máxima la potencia mínima al que están funcionando las centrales de generación eólicas.
- **Hidroeléctrica(%):** Porcentaje sobre la diferencia entre la potencia máxima la potencia mínima al que están funcionando las centrales de generación hidroeléctricas.

5.2.3 Resultados

En esta tabla tenemos reflejado por columnas los siguientes datos:

- **CASO:** Pone el número del caso que estamos simulando.

-
- **Rama_saturada_NAME:** el nombre que recibe la rama está compuesto por el nombre del bus de ida eliminándole la tensión que se incluye con el nombre de éste el programa (132.0 Kv), el bus de llegada y entre paréntesis le ID de la rama para saber si tiene alguna otra en paralelo.
 - **Rama_saturada_BUS:** El número de bus de ida sido el número de bus de llegada y el ID de la rama.
 - **Reac_(80%)(pu):** Reactancia de la rama cuando se ha obtenido una saturación del 80% en Ésta para uno de los casos, en valores por unidad.
 - **Inc_Reac_(sat-80%)(pu):** incremento de la reactancia de la rama (valores por unidad) qué ha sido necesario para que ésta pasara de su punto de saturación máxima a una del 80%.
 - **Reac_(80%)(ohm):** Reactancia de la rama cuando se ha obtenido una saturación del 80% en Ésta para uno de los casos, en Ohmios.
 - **Inc_Reac_(sat-80%)(ohm):** incremento de la reactancia de la rama (valores por unidad) qué ha sido necesario para que ésta pasara de su punto de saturación máxima a una del 80%.
 - **Sat_rama_posreac(%):** Saturación final que posee la rama habiéndole aplicado la reactancia. esta columna está para comprobar que en caso de que fuera la saturación mayor del 80% hemos llegado a situarla es un valor por debajo, normalmente 79 y unos decimales. en el caso de que la saturación ya estuviera por debajo del 80 por ciento no se modifica la rama hipo será la misma relación que al principio.
 - **Sat_rama_prereac(%):** saturación de la rama antes de haberle modificado la reactancia.

Finalmente, después de haber aplicado 500 casos según el método de Montecarlo obtenemos que en la mayoría de ellos (/500), la rama más saturada es la que va del bus 911 a 1052 (ID:1). Antiguamente en la red de la zona norte de la península la rama más saturada siempre solía ser otra mediante un estudio de gestión dinámica se determinó que podía llegar a funcionar por encima de sus capacidades teóricas, como así, por seguridad se le añadió otra rama en paralelo para disminuir su saturación.

Estos son los datos correspondientes a la rama más saturada en la mayor mayoría de los casos, en el peor caso de todos (con la máxima saturación posible):

CASO	Rama_saturada_BUS	Sat_rama_posreac(%)	Sat_rama_prereac(%)
197	911--1052_id:1	79.8197403	141.0471802

Tabla 5.2.3.1 Caso de mayor saturación eléctrica

Sat_rama_(%)	Térmica_(%)	Eólica_(%)	Hidroeléctrica_(%)
141.0471802	5	95	60

Tabla 5.2.3.2 Caso de mayor saturación eléctrica, estado de las centrales.

Reac_(80%)(pu)	Inc_Reac_ (sat-80%)(pu)	Reac_(80%)(ohm)	Inc_Reac_ (sat-80%)(ohm)
0.13849	0.119	24.09725998	20.706

Tabla 5.2.3.3 Caso de mayor saturación eléctrica, reactancias necesarias bajar saturación al 80%

Estos resultados reflejan que a reducir la saturación de la rama de un 141% a un 79.82% son necesarios 20 ohmios a una tensión de 132 kilovoltios. La resolución de lo que quiere decir esto se verá en el apartado de [Conclusiones](#).

5.3 Futuras investigaciones

Se ha contemplado la idea de que en un futuro utópico en el que hubiera reactancias dinámicas instaladas en todas las líneas de la red siendo estas más baratas que ampliar la línea o repotenciarla; podríamos disponer de una gestión dinámica para cada caso posible de la red.

El estudio sería hacer exactamente el mismo proceso hasta el caso de modificación de la reactancia de claro rama más saturada; en ese momento volveríamos a simular el caso estudiado (uno de los 500), pero esta vez volveríamos a buscar la rama más saturada. Seguiríamos reduciendo la reactancia de esta hasta que la rama más saturada fuera otra distinta y así infinitamente hasta que la rama más saturada que tuviéramos que modificar fuera la menos saturada en un origen. A partir de este punto lo único que haríamos sería ir añadiendo cargas innecesarias al circuito.

De esta forma, sabríamos cuál es la gestión de las reactancias que tenemos que hacer para cada caso posible del circuito; y estaríamos hablando de una gestión de la red dinámica total. Pero como ya hemos comentado, este es un futuro utópico porque es altamente improbable que se pueda dar el caso en el que sea más barato construir una reactancia regulable de 20 ohmios a 132 kilovoltios de tensión que mejorar la sección con capacidad de una línea.

6 Anexos

6.1 Instrucciones en Python

En este apartado mostramos el método utilizado en código Python para dar las instrucciones de PSS/E y llevar cabo la simulación del caso de estudio. Se mostrará por orden todas las líneas el archivo “.py” y los resultados que generan estas. Una explicación más extendida de lo que hace cada función por separado se encuentra en el [anexo 4.3](#). en [4.4](#).

Finalmente, a una conclusión de lo que supone ejecutar este código en términos de gasto computacional.

```
#####
# Instrucciones que permiten la ejecución de los comandos de Python sin arrancar PSS/E #
#####
```

```
import sys                                #Importamos la librería "sys" de sistema
sys.path.append("C:\Program Files (x86)\PTI\PSSE33\PSSBIN")
#Le decimos a Python donde está el path, es decir, el directorio con el programa de PSS/E
```

```
import os                                  #Importamos la librería "os" de operative system
os.environ['PATH']+=";C:\Program Files (x86)\PTI\PSSE33\PSSBIN"
#Le decimos a Python donde están las librerías que necesitamos de PSS/E
```

```
import psspy                               #Importamos la librería "psspy" de pss-python librerý
import redirect                             #Importamos la librería "redirect" de redireccionamiento
redirect.psse2py()                          #Usamos la función "psse2py" de la librería "redirect"
psspy.psseinit(100000)
_i=psspy.getdefaultint()
#Obtenemos los valores por defecto de intgar de una máquina (o sea, los valores enteros)
_f=psspy.getdefaultreal()
#Obtenemos los valores por defecto de real de una máquina (o sea, los valores reales)
_s=psspy.getdefaultchar()
#Obtenemos los valores por defecto de char de una máquina (o sea, los valores
característicos)
```

```
import xlrd
```

```

import xlwt
import string
import pyodbc
import random
from time import clock #Sirve para importar un reloj

ni=input("¿Número de casos a simular (ni)? ") # NUMERO ITERACIONES O CASOS
it_0=input("¿Caso inicial (it)? ")
minpctmva=input("¿Porcentaje mínimo de PCTMVA en la rama (minpctmva)? ")

#Conexión con SQL Server
cnxn = pyodbc.connect(driver='{SQL Server}', server='DIEEPC7\PQS',
database='TFGVICTOR', uid='usuario', pwd='****')

com=cnxn.cursor() #Situamos esa conexión creada como base de SQL dentro de Python en
esta ejecución.

#####
#### Instrucciones para leer los casos de estudio y para simular ####
#####

CASOraw= r""" C:\Users\Roy\Desktop\Copia_seguridad_tfgvictor\TFG\Experimentos\MI
TFGTFG.raw """
#Indicamos el directorio en el que se encuentra el archivo “.raw” (código legible por el editor
de block de notas porque no está encriptado).

CASOsav= r""" C:\Users\Roy\Desktop\Copia_seguridad_tfgvictor\TFG\Experimentos\MI
TFG\TFG.sav"""
#Indicamos el directorio en el que se encuentra el archivo “.sav” (ilegible por cualquier editor
de textos porque está encriptado).

#####
#### Instrucciones para Guardar el caso, simular y comenzar el cronómetro del reloj ####
#####

t_0=clock() #Tiempo inicial

```

```

psspy.case(CASOsav);          #Crea un nuevo caso de trabajo en la RAM sobre el que
trabajar a partir de la lectura anterior.
psspy.fnsi([0,0,0,1,1,1,99,0]) #Hadme una simulación utilizando el método FNSL (por
Newton-Rhampson)
psspy.save(CASOsav);        #Guardame el caso de trabajo en la RAM para trabajar
con él.
t_s=clock()                  #Tiempo del final de la simulación
print("Duración de una simulación: "+str(t_s-t_0))

fichero=open("Tiempo.txt","w") #Abrimos "Tiempo.txt" para escribir (write) directamente
sobre lo que haya escrito en él
t_fichero=clock()           #Tiempo de la ejecución
fichero.write("Inicio de la ejecución: "+str(t_fichero-t_0)+"\n"+"Iteración del caso:
"+str(it_0)+"\n")
fichero.close()

#####
                        Instrucciones para recorrer todos los buses
                        y almacenar aquellos pertenecientes a la zona deseada
#####

stdoutsys=sys.stdout
sys.stdout=os.devnull
sys.stdout = stdoutsys

#Esto evita que se pinte de azul el texto en python

tension=132

#CREACIÓN DE UN SUBSISTEMA DE BUSV

psspy. bsys(
sid=1,          # Número del subsistema a generar (1).
usekv=1,       # Base de tensión en kV.
basekv=[132,132], # Coge los buses con tensión 132kV.
numarea=10,

```

```

areas=[10],      # Solo los buses del área 10, que es el área de "ZONA V".
) #Creamos un subsistema, (un vector), de los buses correspondientes al área 10.

ierr, iarray = psspy.abusint(
    sid=1,          # Número del subsistema del que extraer datos.
    flag=1,        # Solo los buses en servicio.
    string="NUMBER" # Pedimos el número de los buses.
) # Obtenemos los números de los buses en servicio del subsistema generado nº 1.

BUSV=iarray[0]
#El primer valor de iarray es la lista buscada (en python el primer valor de una lista ocupa la
posición 0).

print '\n\n-----\n'

#CREACIÓN DE UN SUBSISTEMA DE LOS BUSES GENERADORES Y DE CARGA DE V
ierr, iMbus = psspy.amachint(
    sid=1,      # Número del subsistema del que extraer datos.
    flag=1,    # Solo los buses en servicio de plantas electricas en servicio.
    string=['NUMBER'] # Pedimos el número de los buses.
) #Creamos un vector de los buses generadores.
GENBUSV=iMbus[0]

#OBTENCIÓN DE LOS IDs y Nombres de TODOS LOS GENERADORES DE BUSV
ierr, cMids = psspy.amachchar(
    sid=1,      # Número del subsistema del que extraer datos.
    string=['ID','EXNAME'] # Pedimos las IDS y los Nombres.
) #Creamos una matriz de las IDS y Nombres de los buses generadores.

IDGENBUSV=cMids[0]
GENBUSNAMEV=cMids[1]

print '\n\n-----\n'

LOADBUSV=[]      #Buses de la "Zona V" CARGAS
LOADBUSNAMEV=[]  #Nombres de los buses de la "Zona V" CARGAS
nb=0

```

```

while nb <len(BUSV):      # N° de buses de Viesgo
    BUSNAMEV.append(psspy.notona(BUSV[nb])[1])
    if 1 == psspy.busint(BUSV[nb], 'TYPE')[1]: #V CARGAS
        LOADBUSV.append(BUSV[nb])
        LOADBUSNAMEV.append(psspy.notona(BUSV[nb])[1])
    nb+=1

print '\n\n----- LOADBUSV= TODOS LOS BUSES "CARGAS" EN EL AREA DE V DE
'+str(tension)+' KV -----\n'
print LOADBUSV
print '\n\n-----\n'
print '\n\n----- LOADBUSNAMEV= LOS NOMBRES DE LOS BUSES "CARGAS" EN EL
AREA DE V DE '+str(tension)+' KV -----\n'
print LOADBUSNAMEV
print '\n\n-----\n'

#####
    Vectores con los datos correspondientes a las ramas que se van monitorizar (Bus Inicial,
        Bus Final, Identificador de Rama=ickt)
#####

ierr, branchesfrmtobus = psspy.abrnint(
    sid=1, # Número del subsistema a generar.
    ties=3, # Cogemos todas las ramas del sistema.
    flag=1, # Solo las ramas en servicio.
    string=["FROMNUMBER", "TONUMBER"]
) #Creamos una matriz con el bus de origen y de final de cada rama del circuito.
branchesfrmbus=branchesfrmtobus[0]
branchestobus=branchesfrmtobus[1]

ierr, branchesidfrmtname = psspy.abrnchar(
    sid=1, # Número del subsistema del que extraer datos.
    ties=3, # Cogemos todas las ramas del sistema.
    flag=1, # Solo las ramas en servicio.
    string=["ID", "FROMEXNAME", "TOEXNAME"]
) #Creamos una matriz con la ID, el nombre del bus de origen y el del final de cada rama del
circuito.

```

```
branchesid=branchesidfrmtoname[0]
```

```
branchesfrmname=branchesidfrmtoname[1]
```

```
branchestoname=branchesidfrmtoname[2]
```

```
branches=range(len(branchesid))
```

```
branchesname=range(len(branchesid))
```

```
p=0
```

```
for p in range(len(branchesid)):
```

```
    branches[p]=[branchesfrmtoname[0][p],branchesfrmtoname[1][p],branchesid[p][0]]
```

```
    branchesname[p]=[branchesfrmname[p],branchestoname[p],branchesid[p][0]]
```

```
TOTALRAMAS=branches
```

```
TOTALRAMASNAME=branchesname
```

#VAMOS A ESTIPULAR LAS RAMAS A ESTUDIAR MARCADAS

```
RAMAS=[[920, 903, '1'],[903, 984, '1'],[923, 903, '1'],[920, 903, '2'],[907, 904, '1'],[911, 904, '1'],[900, 901, '1'],[915, 901, '1'],[918, 913, '1'],[929, 919, '1'],[920, 922, '1'],[924, 984, '1'],[925, 917, '1'],[927, 917, '1'],[983, 921, '1'],[983, 917, '2'],[947, 917, '1'],[9050, 906, '1'],[905, 931, '1'],[906, 9010, '1'],[9000, 9010, '2'],[911, 9000, '1'],[906, 914, '1'],[911, 1052, '1'],[940, 1052, '1'],[4, 907, '1'],[908, 301, '1'],[908, 1079, '1'],[4, 909, '1'],[940, 909, '1'],[945, 301, '1'],[9110, 9111, '1'],[1079, 910, '1'],[9110, 942, '1'],[946, 945, '1'],[946, 943, '1'],[912, 943, '1'],[913, 914, '1'],[931, 1161, '1'],[931, 914, '1'],[1284, 915, '1'],[928, 942, '1']]
```

#Buscamos en la lista de ramas original las que coinciden con las ramas que queremos estudiar

```
RAMASNAME=[]
```

```
p=0
```

```
coincidencia=[]
```

```
for p in range(len(RAMAS)):
```

```
    coincidencia=TOTALRAMAS.index(RAMAS[p])
```

```
    RAMASNAME.append(TOTALRAMASNAME[coincidencia])
```

```
p=0
```

```
a=0
```

```
b=0
```

```

RAMAS2=[]
RAMASNAME2=[]
while p<len(RAMASNAME):
    divalpha=RAMASNAME[p][0].replace("132.00","").split()
    divgamma=RAMAS[p][0]
    if len(divalpha)==3:
        a=str(divalpha[0])+str(divalpha[1])
    else:
        a=str(divalpha[0])
    divbeta=RAMASNAME[p][1].replace("132.00","").split()
    divdelta=RAMAS[p][1]
    if len(divbeta)==3:
        b=str(divbeta[0])+str(divbeta[1])
    else:
        b=str(divbeta[0])
    RAMAS2.append(str(RAMAS[p][0])+"--"+str(RAMAS[p][1])+"_id:"+str(RAMAS[p][2]))
    RAMASNAME2.append(a+"-"+b+"_"+str(RAMASNAME[p][2]))
    p+=1

```

```
print ("\n----- RAMAS=TODAS LAS RAMAS DE EL AREA DE V -----\n')
```

```
print ("Ramas, [Bus de inicio, Bus final, tipo línea]: "+str(RAMAS)+"\n")
```

```
#####
```

Instrucción para Obtener los valores de Potencia de todas las centrales y de cada conjunto de centrales clasificados en Eólica, Térmica e Hidroeléctrica

```
#####
```

```
i=0
```

```
POTGENBUSV=[]
```

```
POTMAXGENBUSV=[]
```

```
POTMINGENBUSV=[]
```

```
GENBUSZONEV=[]
```

```
GENBUSVTER=[]
```

```
GENBUSNAMEVTER=[]
```

```
POTGENBUSVTER=[]
POTMAXGENBUSVTER=[]
POTMINGENBUSVTER=[]
```

```
GENBUSVEOL=[]
GENBUSNAMEVEOL=[]
POTGENBUSVEOL=[]
POTMAXGENBUSVEOL=[]
POTMINGENBUSVEOL=[]
```

```
GENBUSVHDR=[]
GENBUSNAMEVHDR=[]
POTGENBUSVHDR=[]
POTMAXGENBUSVHDR=[]
POTMINGENBUSVHDR=[]
```

```
while i<len(GENBUSV):
```

```
    Pg= psspy.macdat(GENBUSV[i],IDGENBUSV[i],'P')[1]
```

```
    #Obtenemos la potencia de un generador proporcionando su nº de bus y su ID dentro del bus (si hay más de uno)
```

```
    POTGENBUSV.append(Pg)
```

```
    Pgmax=psspy.macdat(GENBUSV[i],IDGENBUSV[i],'P MAX')[1] #Obtenemos la potencia máxima de un generador proporcionando su nº de bus y su ID dentro del bus (si hay más de uno)
```

```
    POTMAXGENBUSV.append(Pgmax)
```

```
    Pgmin=psspy.macdat(GENBUSV[i],IDGENBUSV[i],'P MIN')[1]
```

```
    #Obtenemos la potencia mínima de un generador proporcionando su nº de bus y su ID dentro del bus (si hay más de uno)
```

```
    if Pgmin== -9999.0:
```

```
        Pgmin=0.0
```

```
    POTMINGENBUSV.append(Pgmin)
```

```
    mbase=psspy.macdat(GENBUSV[i],IDGENBUSV[i], 'M BASE')[1]
```

```
    print("Mbase: "+str(mbase))
```

```
    zn=psspy.busint(GENBUSV[i],'ZONE')[1]
```

```
    GENBUSZONEV.append(zn)
```

```
    if zn==7:
```

```
        GENBUSVTER.append(GENBUSV[i])
```

```

GENBUSNAMEVTER.append(GENBUSNAMEV[i])
POTGENBUSVTER.append(POTGENBUSV[i])
POTMAXGENBUSVTER.append(POTMAXGENBUSV[i])
POTMINGENBUSVTER.append(POTMINGENBUSV[i])

```

elif zn==80 or zn==82 or zn==83:

```

GENBUSVEOL.append(GENBUSV[i])
GENBUSNAMEVEOL.append(GENBUSNAMEV[i])
POTGENBUSVEOL.append(POTGENBUSV[i])
POTMAXGENBUSVEOL.append(POTMAXGENBUSV[i])
POTMINGENBUSVEOL.append(POTMINGENBUSV[i])

```

elif zn==87 or zn==86:

```

GENBUSVHDR.append(GENBUSV[i])
GENBUSNAMEVHDR.append(GENBUSNAMEV[i])
POTGENBUSVHDR.append(POTGENBUSV[i])
POTMAXGENBUSVHDR.append(POTMAXGENBUSV[i])
POTMINGENBUSVHDR.append(POTMINGENBUSV[i])

```

i=i+1

```

print '\n\n----- BUSV= TODOS LOS BUSES EN EL AREA DE V DE '+str(tension)+' KV ----
-----\n'
print BUSV
print '\n\n-----\n'
print '\n\n----- BUSNAMEV= LOS NOMBRES DE LOS BUSES EN EL AREA DE V DE
'+str(tension)+' KV -----\n'
print BUSNAMEV
print '\n\n----- GENBUSV= TODOS LOS BUSES "GENERADORES" EN EL AREA DE V
DE '+str(tension)+' KV -----\n'
print GENBUSV
print '\n\n----- GENBUSNAMEV= NOMBRE DE LOS BUSES "GENERADORES" DE V ----
-----\n'
print GENBUSNAMEV
print '\n\n----- POTGENBUSV= POTENCIA DE LOS BUSES "GENERADORES" DE V EN
MW-----\n'
print POTGENBUSV

```

```
print '\n\n----- POTMAXGENBUSV=POTENCIA MAX DE LOS BUSES "GENERADORES"
DE V EN MW-----\n'
print POTMAXGENBUSV
print '\n\n----- POTMINGENBUSV=POTENCIA MIN DE LOS BUSES "GENERADORES"
DE V EN MW-----\n'
print POTMINGENBUSV
print '\n\n----- GENBUSZONEV=NUMERO DE ZONA DE LOS BUSES "GENERADORES"
DE V -----\n'
print GENBUSZONEV

print '\n\n-----\n'
print '\n\n-----\n'

print '\n\n----- GENBUSVTER= TODOS LOS BUSES "GENERADORES" TERMICOS EN
EL AREA DE V DE '+str(tension)+' KV -----\n'
print GENBUSVTER
print '\n\n----- GENBUSNAMEVTER=NOMBRE DE LOS BUSES "GENERADORES"
TERMICOS DE V -----\n'
print GENBUSNAMEVTER
print '\n\n----- POTGENBUSVTER=POTENCIA DE LOS BUSES "GENERADORES"
TERMICOS DE V EN MW-----\n'
print POTGENBUSVTER
print '\n\n----- POTGENBUSVTER=POTENCIA DE LOS BUSES "GENERADORES"
TERMICOS DE V EN MW-----\n'
print POTMAXGENBUSVTER

print '\n\n-----\n'

print '\n\n----- GENBUSVEOL= TODOS LOS BUSES "GENERADORES" EOLICOS EN EL
AREA DE V DE '+str(tension)+' KV -----\n'
print GENBUSVEOL
print '\n\n----- GENBUSNAMEVEOL=NOMBRE DE LOS BUSES "GENERADORES"
EOLICOS DE V -----\n'
print GENBUSNAMEVEOL
print '\n\n----- POTGENBUSVEOL=POTENCIA DE LOS BUSES "GENERADORES"
EOLICOS DE V EN MW-----\n'
print POTGENBUSVEOL
```

```

print '\n\n-----\n'

print '\n\n----- GENBUSVHDR= TODOS LOS BUSES "GENERADORES"
HIDROELECTRICOS EN EL AREA DE V DE '+str(tension)+' KV -----\n'
print GENBUSVHDR
print '\n\n----- GENBUSNAMEVHDR= NOMBRE DE LOS BUSES "GENERADORES"
HIDROELECTRICOS DE V -----\n'
print GENBUSNAMEVHDR
print '\n\n----- POTGENBUSVHDR= POTENCIA DE LOS BUSES "GENERADORES"
HIDROELECTRICOS DE V EN MW -----\n'
print POTGENBUSVHDR

print '\n\n-----\n'
print '\n\n-----\n'

```

#####

Se crean los vectores con los números aleatorios utilizando la expresión random. ni = número de iteraciones o casos.

Se crean 3 vectores de longitud ni para cada uno de los 3 tipos distintos de generación.

#####

```

pctter=[] #Porcentaje asociado a la potencia de las centrales térmicas
pcteol=[] #Porcentaje asociado a la potencia de las centrales eólicas
pcthdr=[] #Porcentaje asociado a la potencia de las centrales hidroeléctricas

p=0
while p < ni:
    pctter.append(random.randrange(0,100,5))
    pcteol.append(random.randrange(0,100,5))
    pcthdr.append(random.randrange(0,100,5))
    p+=1

#####

```

```
#### Generación de la tabla de potencias en SQL a partir de los valores anteriores ####
#####
```

```
# Reiniciamos las tablas con los datos antiguos
```

```
com.execute("DELETE FROM [TFGVICTOR].[dbo].[Potencias]")
com.commit()
com.execute("DELETE FROM [TFGVICTOR].[dbo].[Casos]")
com.commit()
com.execute("DELETE FROM [TFGVICTOR].[dbo].[Resultados]")
com.commit()
p=0
```

```
# Este bucle sirve para recorrer todos los componentes de una lista GENBUSV e ir
introduciendolos en la tabla SQL.
```

```
for p in range(len(GENBUSV)):
```

```
    st="INSERT INTO [TFGVICTOR].[dbo].[Potencias] (GENBUSV,GENBUSNAMEV,
    POT_Normal,POTMIN,POTMAX)          VALUES          ("+str(GENBUSV[p])+",
    "+str(GENBUSNAMEV[p])+"," "+str(POTGENBUSV[p])+"," "+str(POTMINGENBUSV[
    p])+"," "+str(POTMAXGENBUSV[p])+")"
```

```
com.execute(st) # Insertame el stament, es decir, ejecuta la función creada en la
línea anterior y asociada a "st".
```

```
com.commit() # Guardamos el cambio hecho, este comando no se usa normalmente,
pero en Python es necesario porque daría un error por deja un proceso abierto.
```

```
#####
#### COMIENZO DE LAS ITERACIONES EN FUNCION DE NI (Nº de iteraciones) ####
#####
```

```

PCTMVAMAXRAMAS=[]          # Lista de la saturación de cada rama
PCTMVALUDRIOMEIRA=[]
RAMAVIP=[]                 # Buses de la rama más saturada
RAMAVIPNAME=[]            # Nombre de la rama más saturada
REAC=[]                    # Nuevas reactancias de las líneas
INCREAC=[]                 # Reactancias añadidas a las líneas

for it in range(0,ni,1):
    t_it0=clock()
    NEWPOTGENBUSVTER=[]
    NEWPOTGENBUSVEOL=[]
    NEWPOTGENBUSVHDR=[]

    #####
    ## TÉRMICAS ##
    #####

    #Cálculo de las potencias en función de máximos y mínimos para las centrales Térmicas
    p=0
    while p<len(POTGENBUSVTER):
        NEWPOTGENBUSVTER.append((POTMAXGENBUSVTER[p]-
        POTMINGENBUSVTER[p])*(pctter[it]/100.0))
        p+=1
    #Modificación de tales potencias en el fichero.sav para su posterior simulación de las
    potencias en función de máximos y mínimos para las centrales Térmicas
    #ierr = machine_data_2(i=ibus, id, intgar, realar)
    p=0
    while p<len(POTGENBUSVTER):
        psspy.machine_data_2(GENBUSVTER[p], '1', [_i],[NEWPOTGENBUSVTER[p]])
        #Modificación de la potencia de los generadores térmicos
        p+=1

    #####
    ## EÓLICAS ##
    #####

```

#Cálculo de las potencias en función de máximos y mínimos para las centrales Eólicas

p=0

while p<len(POTGENBUSVEOL):

NEWPOTGENBUSVEOL.append((POTMAXGENBUSVEOL[p]-

POTMINGENBUSVEOL[p])*(pcteol[it]/100.0))

p+=1

p=0

while p<len(POTGENBUSVEOL):

psspy.machine_data_2(GENBUSVEOL[p], '1', [_i], [NEWPOTGENBUSVEOL[p]])

p+=1

#####

HIDROELÉCTRICAS

#####

#Cálculo de las potencias en función de máximos y mínimos para las centrales
Hidroeléctricas

p=0

while p<len(POTGENBUSVHDR):

NEWPOTGENBUSVHDR.append((POTMAXGENBUSVHDR[p]-

POTMINGENBUSVHDR[p])*(pcthdr[it]/100.0))

p+=1

p=0

while p<len(POTGENBUSVHDR):

psspy.machine_data_2(GENBUSVHDR[p], '1', [_i], [NEWPOTGENBUSVHDR[p]])

p+=1

Simulación ### ### Simulación ### ### Simulación ### ### Simulación

Simulación ### ### Simulación ### ### Simulación ### ### Simulación

Simulación ### ### Simulación ### ### Simulación

psspy.fdns([0,0,0,1,1,1,99,0])

U=psspy.solv() #Para saber si se ha resuelto el sistema o no

```
#####
## Obtención de la saturación de las ramas del circuito ##
#####
```

p=0

PCTMVARAMAS=[]

while p<len(RAMAS):

rama=RAMAS[p]

ramaname=RAMASNAME2[p]

frmBus = rama[0]

toBus = rama[1]

ckt = rama[2]

```
#####
## Obtención de la saturación de cada rama del circuito ##
#####
```

```
PCTMVARAMAS.append(psspy.brnmsc(ibus=rama[0], jbus=rama[1], ickt=rama[2],
string="PCTMVA")[1])
```

#Me interesa solo la $PCTMVA = S/S_{max} * 100 = I_{total}/I_{max} * 100 = \text{Saturación}$

p+=1

PCTMVAMAXRAMAS.append(max(PCTMVARAMAS)) #La rama con mayor saturación

RAMAVIP.append(RAMAS2[PCTMVARAMAS.index(max(PCTMVARAMAS))])

RAMAVIPNAME.append(RAMASNAME2[PCTMVARAMAS.index(max(PCTMVARAMAS))])

```
#####
## Insertamos en la tabla SQL "Casos" los datos correspondientes ##
#####
```

```
st="INSERT INTO [TFGVICTOR].[dbo].[Casos] (CASO, Rama_saturada_NAME,
Rama_saturada_BUS, [Sat_rama_(%)], [Termica_(%)], [Eolica_(%)], [Hidroelectrica_(%)])
VALUES (" +str(it)+",
```

```
" +str(RAMAVIPNAME[it])+" ", "+str(RAMAVIP[it])+" ", "+str(PCTMVAMAXRAMAS[it])+" ", "+str(
pctter[it])+" ", "+str(pcteol[it])+" ", "+str(pcthdr[it])+" )"
```

com.execute(st) # Insertame el stament, es decir, ejecuta la función creada en la línea anterior y asociada a "st".

com.commit() # Guardamos el cambio hecho, este comando no se usa normalmente, pero en Python es necesario porque daría un error por deja un proceso abierto.

```
posramavip=PCTMVARAMAS.index(max(PCTMVARAMAS)) #Posición en la lista de la
rama con mayor PCTMVA
```

```
frmBusramavip=RAMAS[posramavip][0] #Bus de ida de la rama con mayor PCTMVA
```

```
toBusramavip=RAMAS[posramavip][1] #Bus de llegada de la rama con mayor PCTMVA
```

```
cktramavip=RAMAS[posramavip][2]
```

```
pctmvamaxramas=PCTMVAMAXRAMAS[it]
```

```
valores=psspy.brndt2(ibus=frmBusramavip, jbus=toBusramavip, ickt=cktramavip,
string="RX")[1]
```

```
oldresis=valores.real # Valor de la resistencia de la línea más saturada.
```

```
oldreac=valores.imag # Valor de la reactancia de la línea más saturada.
```

```
newreac=oldreac
```

```
p=0
```

```
m=0
```

```
MINPCTMVAMAXRAMAS=[]
```

```
PCTMVAMAXRAMAS2=[]
```

```
NEWREACMIN=[]
```

```
t_s0=clock()
```

```
while pctmvamaxramas>minpctmva: #Si pctmvamaxramas es un mayor que el minpctmva
asignado.
```

```
#####
## Modificación de la reactancia de una rama conservando igual los otros datos ##
#####
```

```

##psspy.branch_chng(i=946, j=945, ckt="1", intgar=[1,1,1,0,0,0], realar=None, **kwds)
##realar=[resis, reac, charge, ratea, rateb, ratec, 0, 0, 0, 0, length, 1, 1, 1, 1]

newresis=oldresis
newreac+=0.001 #Nueva reactancia de la línea en (pu), la incrementamos en 0.001
(pu)==0.2(ohm)
error_branch_chng=psspy.branch_chng(i=frmBusramavip, j=toBusramavip,
cktramavip="1", intgar=None, realar=[newresis, newreac])

### Simulación ###
psspy.fdns([0,0,0,1,1,1,99,0])

## Obtención de la PCTMVA de una rama
PCTMVAMAXRAMAS2.append((psspy.brnmsc(ibus=frmBusramavip,
jbus=toBusramavip, ickt=cktramavip, string="PCTMVA")[1]))
pctmvamaxramas=PCTMVAMAXRAMAS2[p]
if p>=2:
    if PCTMVAMAXRAMAS2[p]>PCTMVAMAXRAMAS2[p-1]
    and PCTMVAMAXRAMAS2[p-1]<PCTMVAMAXRAMAS2[p-2]: #Si encontramos un
    mínimo en la función de pctmvamaxramas
        MINPCTMVAMAXRAMAS.append(PCTMVAMAXRAMAS2[p-1]) #Almacenamos
        el primer mínimo
        NEWREACMIN.append(newreac)

    if MINPCTMVAMAXRAMAS[m]>MINPCTMVAMAXRAMAS[m-1]:
        minpctmvamaxramasfinal=MINPCTMVAMAXRAMAS[m-1]
        pctmvamaxramas=minpctmvamaxramasfinal
        newreac=NEWREACMIN[m-1]
        break
    m+=1
if m>=1:
    if PCTMVAMAXRAMAS2[p]>(MINPCTMVAMAXRAMAS[m-1]+20):
        minpctmvamaxramasfinal=MINPCTMVAMAXRAMAS[m-1]
        pctmvamaxramas=minpctmvamaxramasfinal
        newreac=NEWREACMIN[m-1]
        break

```

```

    p+=1
    t_send=clock()
    if p>0:
        tsimulaciones=(t_send-t_s0)/p

    REAC.append(newreac)          #LISTA CON LA Nueva Reactancia de la linea (pu)
    increac=newreac-oldreac      #Diferencia entre la reactancia antigua de la línea y la
    nueva, es decir, LA REACTANCIA AÑADIDA (pu)
    INCREAC.append(increac)
    zbase=132*132/100
    newreacohm=newreac*zbase     #La Nueva Reactancia de la línea (ohm)
    increacohm=newreacohm-oldreac*zbase #Diferencia entre la reactancia antigua de la
    línea y la nueva, es decir, LA REACTANCIA AÑADIDA (ohm)

#####
    ## Insertamos en la tabla SQL "Resultados" los datos correspondientes ##
#####

st="INSERT INTO [TFGVICTOR].[dbo].[Resultados] (CASO, Rama_saturada_NAME,
Rama_saturada_BUS, [Reac_(80%)(pu)], [Inc_Reac_(sat-80%)(pu)], [Reac_(80%)(ohm)],
[Inc_Reac_(sat-80%)(ohm)], [Sat_rama_posreac(%)], [Sat_rama_prereac(%)]) VALUES
("+str(it+it_0)+",
"+str(RAMAVIPNAME[it])+",""+str(RAMAVIP[it])+",""+str(REAC[it])+",""+str(increac)+",""+str(
newreacohm)+",""+str(increacohm)+",""+str(pctmvamaxramas)+",""+str(PCTMVAMAXRAMA
S[it])+")"
com.execute(st) # Insertame el stament, es decir, ejecuta la función creada en la línea
anterior y asociada a "st".
com.commit() # Guardamos el cambio hecho, este comando no se usa normalmente, pero
en Python es necesario porque daría un error por deja un proceso abierto.

#Ahora hay que dejar el circuito como estaba antes del cambio de la reactancia:

#####
## Modificación de la reactancia de una rama conservando igual los otros datos ##
#####

```

```
error_branch_chng=psspy.branch_chng(i=frmBusramavip, j=toBusramavip,  
cktramavip="1", intgar=None, realar=[oldresis,oldreac])
```

```
fichero = open('Tiempo.txt','a') # Abrimos 'Tiempo.txt' para añadirle (append) líneas  
t_itend=clock() # Tiempo de la ejecución  
titeraciones=t_itend-t_it0  
fichero.write("Inicio de la ejecución: "+str(titeraciones)+"\n"+"Iteración del caso:  
"+str(it+it_0)+"\n")  
fichero.close()
```

```
print(RAMAVIP)  
print(RAMAVIPNAME)  
print(PCTMVAMAXRAMAS)
```

```
t_end=clock()  
duracion=t_end-t_0
```

```
print("Reactancias añadidas a las líneas (pu): "+str(INCREAC))  
print("Máxima reactancia añadida a la línea: "+str(max(INCREAC))+ " correspondiente al caso  
"+str(INCREAC.index(max(INCREAC))))  
print("Duracion de cada simulacion(modificacion reac) en bucle: "+str(tsimulaciones))  
print("Duracion de cada iteración en bucle: "+str(titeraciones))  
print("Duracion del proceso: "+str(duracion))
```

```
fichero = open('Tiempo.txt','a')  
t_fichero=clock()  
fichero.write("Inicio de la ejecución: "+str(t_fichero-t_0)+"\n"+"Iteración del caso: "+str(it)+"\n")  
fichero.close()
```

7 Conclusiones

Como primera conclusión, podemos confirmar que el análisis del flujo de cargas mediante sistemas de simulación ha sido satisfactorio, siendo este una continuación de los estudios de otros compañeros, pero con un sistema de red actualizado. Los datos obtenidos son fáciles de interpretar presentando estos una adecuada correspondencia con el comportamiento real de la red; Esto demuestra utilidad del estudio.

Aunque el trabajo se ha hecho en colaboración con una empresa distribuidora, todos los resultados se han anonimizado por confidencialidad. Esto quiere decir que, aunque los resultados no pueden aplicarse a la realidad ni son de dominio público, el trabajo y conclusiones obtenidas en el proyecto sí que pueden interesarle a la sociedad.

En el proyecto no solamente se ha desarrollado un sistema automatizado para simulación del flujo de cargas (PSS/E) así como la obtención de todos los datos importantes de la red, como tensión de línea saturación de las ramas etc. sino que se ha ido más allá que se ha calculado para cualquier caso posible cuál es la rama más dura del sistema y que reactancia sería necesaria añadirla para que volviera a márgenes seguros de operación.

Por otro lado, también se ha hecho una labor de investigación en esta nueva tecnología, Y desarrollado posibles ideas para un futuro de una red completamente integrada y controlada. Una de esas ideas de propuesta de futuro, fáciles de gestionar computacionalmente, pero muy costosas de aplicar con la tecnología actual; sería desarrollar en cada caso posible de generación y consumo de la red (estando ésta controlada gracias a una reactancia con L variable), una forma de conocer el valor de la reactancia a añadir en cada línea de tal forma que estuvieran todas al mismo nivel de saturación. Esto se traduce en que podríamos gestionar en todo momento la saturación todas las ramas de nuestro sistema gracias a poseer reactancias de línea variables electrónicamente en cada una de ellas.

En lo que se refiere al caso estudiado llegamos a la conclusión de que con un estado de la red de Generación térmica al 5%, Eólica al 95% e Hidroeléctrica al 60%; la rama más saturada es aquella que conecta el bus 911 con el 1052 (los nombres de estos han sido ocultados por confidencialidad).

Hemos sido capaces de reducir la saturación de un 141.05 % a un 79.82% sirviéndonos de una reactancia de línea acoplada en serie de 20.71 ohm.

La impedancia común de una línea de esas características es de 3 ohm, esto se traduce en que con la tecnología actual (donde el costo de cada ohmio obtenido utilizando tecnología de vanguardia como las reactancias en línea de "Smart wires" es de 1.250.000€) no es posible sacar más rendimiento de las instalaciones bajo condiciones de seguridad sin vernos obligados a repotenciar la línea.

8 Bibliografía

- [1] F. J. B. García, *Sistemas de energía eléctrica en alta tensión*, Santander: Universidad Cantabria, 2017.
- [2] A. G. Berrocal, «<https://es.slideshare.net>,» [En línea]. Available: <https://es.slideshare.net/GilJGH/simulacin-apli> . [Último acceso: 16 Agosto 2019].
- [3] H. R. E. González, «<http://acacia.org.mx>,» [En línea]. Available: http://acacia.org.mx/busqueda/pdf/11_06_simulacion.pdf. [Último acceso: 15 Agosto 2019].
- [4] Smartwires, «<https://www.smartwires.com>,» [En línea]. Available: <https://www.smartwires.com/increase-power-transfer-capacity/>. [Último acceso: 14 Septiembre 2019].
- [5] D. H. Begines, «Modelado de Redes de Distribución Usando,» [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/5576/fichero/Proyecto+fin+de+carrera.pdf>. [Último acceso: 20 Agosto 2019].
- [6] «Power System Simulation for Engineers (PSS/E): Fault Analysis,» [En línea]. Available: http://home.engineering.iastate.edu/~jdm/ee457/PSSE_Fault_Intro_Instructions.pdf. [Último acceso: 22 Agosto 2019].
- [7] Microsoft, «Access SQL: conceptos básicos, vocabulario y sintaxis,» Microsoft, [En línea]. Available: <https://support.office.com/es-es/article/access-sql-conceptos-básicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671>. [Último acceso: 26 Agosto 2019].
- [8] Siemens, «Getting started with PSS®E: Tutorial 1 - The GUI,» Siemens, [En línea]. Available: <https://www.youtube.com/watch?v=EX1T9ISNBHo>. [Último acceso: 5 Febrero 2019].
- [9] P. M. Anderson, *Power System Protection*, Verlag: IEEE Publications,U.S., 1998.
- [10] J. G. Castillejos, *Cálculo de sistemas eléctricos con PSS/E, Régimen permanente,*, Edicions UPC.
- [11] E.ON Distribución, «Proyecto Dynelec,» *INDUSTRIAMBIENTE*, p. 35 a 41, 2013.
- [12] Colorado State University, « Introduction to PSS/E (Power System Simulation for Engineering),» Colorado State University , Colorado, 2015.
- [13] Z. A. Shaw, *Learn More Python 3 the hard way*, Addison-Wesley, 2017.

- [14] P. P. Anjan K. Deb, Power Line Ampacity System Theory, Modeling, and Applications, Boca Raton London New York Washington, D.C.: CRC Press, 2000.
- [15] I. A. G. & D. A. d. Rosso, «REDUCCIÓN DE PÉRDIDAS EN SISTEMAS DE TRANSMISIÓN Y DISTRIBUCIÓN,» CÁMARA ARGENTINA DE LA CONSTRUCCIÓN, Buenos Aires, 2013.
- [16] «Tutorial PSS/E v. 32,» [En línea]. Available: <https://es.scribd.com/document/245650437/Tutorial-PSS-E-parte1-Flujo-de-Potencia>. [Último acceso: 10 Marzo 2019].
- [17] «Psspy.org,» Psspy.org, [En línea]. Available: <https://psspy.org/psse-help-forum/questions/>. [Último acceso: 17 Septiembre 2019].
- [18] A. S. Anwer, «PSS/E Tutorial,» [En línea]. Available: <https://www.youtube.com/channel/UCmGtyISVaUY1bcNNOI9H5vA/videos>. [Último acceso: 17 Septiembre 2019].
- [19] Simulación de sistemas eléctricos, Pearson Prentice Hall, 2005.
- [20] K. R. Padiyar, Power system dynamics : Stability and control.
- [21] L. M. Checa, Líneas de transporte de energía, Barcelona: L.M.Checa, 1979.
- [22] J. D. M. S. Glover, Power system analysis and design : [with personal computer applications], PWS Publishing Company.
- [23] T. Gönen, Modern power system analysis.
- [24] V. autores, «CALIDAD DEL SUMINISTRO,» UNIVERSIDAD DE CANTABRIA , Laredo, 2002.
- [25] M. M. Canteli, «Simulación de flujos de carga,» Presentación powerpoint, 2002.
- [26] Dev.mysql, «Dev.mysql.com,» dev.mysql.com, [En línea]. Available: <https://dev.mysql.com/doc/connector-python/en/connector-python-api-mysqlicursor-fetchall.html>. [Último acceso: 27 Agosto 2019].
- [27] Khanacademy, «khanacademy.org,» khanacademy.org, [En línea]. Available: <https://es.khanacademy.org/computing/computer-programming/sql>. [Último acceso: 25 Agosto 2019].
- [28] W3schools, «SQL Tutorial,» [En línea]. Available: <https://www.w3schools.com/sql/>. [Último acceso: 28 Agosto 2019].