

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

INFORME DE PASANTÍA

ANÁLISIS DE UNA ESTRATEGIA DE IDENTIFICACIÓN DE FALLAS DE BAJA IMPEDANCIA UTILIZANDO OPENDSS Y MATLAB.

AUTORES

Julio Ernesto Castro Rico
Código:20192372005
Diego Alberto Anzola Bustos
Código:20192372002

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD TECNOLÓGICA

INGENIERÍA ELÉCTRICA POR CICLOS PROPEDÉUTICOS

BOGOTÁ D.C

2023

Índice

1. Introducción **2**
1.1. Planteamiento . . . 2
1.2. Objetivos . . . 3
1.3. Marco teórico . . . 3
1.4. Estado del arte . . . 5

2. Metodología **7**
2.1. Fase 1: Busqueda de documentos . . . 7
2.2. Fase 2: Aprendizaje del software OpenDSS . . . 7
2.3. Fase 3: Desarrollo del módulo de identificación de fallas para el sistema IEEE de 13 y 34 nodos en Appdesigner . . . 7
2.4. Fase 4: Material académico . . . 7

3. Resultados **8**
3.1. Módulos de identificación de fallas de baja impedancia en el sistema IEEE de 13 y 34 nodos desarrollado en Appdesigner . . . 8
3.2. Comprobación de resultados para la identificación de fallas . . . 11
3.3. Manual sobre el uso de los módulos de identificación de Fallas de baja impedancia . . . 11
3.4. Elaboración de un manual sobre el uso de OpenDSS y su enlace con Matlab. . . . 11
3.5. Elaboración de videos sobre el uso de OpenDSS, modelamiento de elementos asociados a un sistema de Potencia y modelamiento de Fallas de baja impedancia. . . . 53

4. Conclusiones **54**

5. Trabajos futuros **54**

Referencias **54**

1. Introducción

Los Sistemas Eléctricos de Potencia (SEP) comprenden todos los equipos utilizados en la generación, transmisión, distribución de energía eléctrica, iniciando desde las plantas de generación como son las térmicas e hidráulicas hasta todos los dispositivos de consumo [1]. El aumento de la demanda de energía en muchos sectores de la sociedad ha conllevado a expandir y mejorar toda la composición del SEP, de tal manera que se pueda atender la demanda requerida y que se garantice un suministro continuo. A medida que se aumenta el tamaño de todos los componentes que conforman el sistema, se crean nuevas redes para expandirlo y se anexen sistemas de generación basados en las energías renovables, pueden ocurrir sucesos como son fallas de baja impedancia que son eventos no controlados que afectan directamente la operación de los SEP.

Las fallas de baja impedancia ocurridas en algún punto de la zona pueden generan desestabilización en la red, ocasionando que se suspenda el servicio en un determinado tiempo o que ocurran daños menores o mayores en el propio sistema. Por lo tanto, existe la necesidad de desarrollar métodos como son algoritmos y dispositivos, que además de proteger los sistemas de eléctricos de potencia, puedan tener la capacidad de detectar e identificar las fallas de baja impedancia de forma inmediata, para así poder realizar un diagnóstico oportuno e implementar de forma automática acciones correctivas para el pronto restablecimiento del servicio y determinen los daños ocasionados [2].

El presente documento está compuesto en 4 secciones. La primera sección corresponde a la introducción donde se incluye el planteamiento del proyecto, los objetivos propuestos, el marco teórico y el estado del arte. La segunda sección se explica la metodología desarrollada en 4 fases para la ejecución del presente proyecto de pasantía. La tercera sección son los resultados donde se explica la interfaz de los módulos de identificación de fallas implementados en la aplicación APP LD-UD y la elaboración de los manuales. La cuarta sección corresponde a las conclusiones del proyecto. Complementariamente, se agregó trabajos futuros, donde se indica la posibilidad de agregar nuevos métodos en la presente aplicación y se anexó una tabla con una lista de vídeos aplicados al modelamiento del sistema IEEE de 13 nodos en software de OpenDSS.

1.1. Planteamiento

El grupo de investigación en Sistemas de Potencia de la Universidad Distrital Francisco José de Caldas (GIS-PUD), es un grupo que tiene diversas líneas de investigación en el campo de los Sistemas eléctricos de Potencia. Dentro de su campo de investigación se encuentra el desarrollo de sistemas inteligentes, optimización, automatización, estrategias para reducir la tasa de abandono, repitencia y permanencia de los estudiantes en el proyecto curricular de tecnología en electricidad de media y baja tensión e ingeniería eléctrica por ciclos propedéuticos y el desarrollo de herramientas de aprendizaje en diversas áreas que conforman al SEP como son las aulas virtuales, desarrollo de aplicaciones y videotutoriales publicados en plataformas digitales.

Es por esto que para el presente proyecto de modalidad pasantía se propuso complementar una aplicación previamente desarrollada por el grupo de investigación de Sistemas y Redes Cognitivas (SIREC) relacionada con la localización de fallas de baja impedancia llamada APP LF-UD. Los algoritmos se desarrollaron en Matlab y la aplicación se realizó con el entorno de desarrollo de aplicaciones Appdesigner y la implementación de los sistemas IEEE de 13 y 34 nodos se elaboraron en el software libre OpenDSS. En la aplicación APP LF-UD, se anexó dos nuevos módulos correspondientes a la identificación de fallas de baja impedancia, usando los dos sistemas IEEE ya contenidos en el archivo base. Este nuevo módulo tiene la capacidad de procesar y analizar los datos de la falla programada, verificar el tipo de falla que ocurre en cada uno de los sistemas, mostrar corrientes de falla en magnitud, ángulo, mostrar un diagrama vectorial las corrientes de falla. Por último, se agregó un tercer modulo aplicado a la variación de parámetros de la línea de transmisión.

1.2. Objetivos

Objetivo General

- Análisis de una estrategia de identificación de fallas de baja impedancia utilizando OpenDSS y Matlab.

Objetivos específicos

- Realizar una investigación y documentar sobre la identificación de fallas de baja impedancia usando el software OpenDSS.
- Establecer los parámetros para el modelamiento de sistemas eléctricos de potencia a través del software OpenDSS.
- Seleccionar e implementar un caso de estudio para la identificación de fallas de baja impedancia utilizando OpenDSS y Matlab.
- Desarrollar material académico sobre el software OpenDSS y su conexión a través de Matlab.

1.3. Marco teórico

Todo sistema trifásico senoidal desequilibrado, representado por tres fasores desequilibrados, puede sustituirse por la suma de tres sistemas de fasores simétricos: un sistema directo o de secuencia positiva; un sistema inverso o de secuencia negativa y un sistema homopolar o de secuencia cero, que constituyen las componentes simétricas del sistema desequilibrado [3].

Secuencia positiva o sistema directo

En las componentes de secuencia positiva está representada por tres fasores de igual magnitud con un desplazamiento de 120° y mantienen la secuencia de a,b,c. En la Figura 1 se muestra el sistema en secuencia positiva.

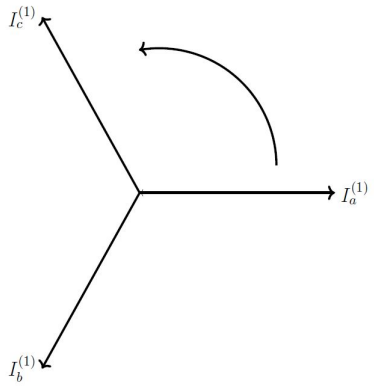


Figura 1: Secuencia positiva

Secuencia negativa o inversa

De igual forma para las componentes de secuencia negativa se representará con tres fasores de igual magnitud o modulo, con un desplazamiento de 120° y mantienen la secuencia a,c,b. En la Figura 2 se muestra el sistema en componentes de secuencia negativa.

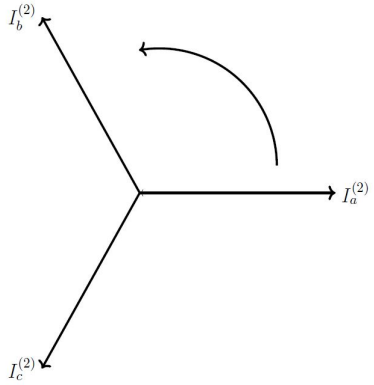


Figura 2: Secuencia negativa

Secuencia cero o homopolar

Para las componentes de secuencia cero se representa por tres fasores con la misma magnitud y con el mismo ángulo, tal como se puede observar en la figura 3.

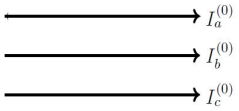


Figura 3: Secuencia cero

Representación de las tensiones y corrientes en componentes de secuencia y fase

Para la obtención de las corrientes de secuencia y fase, se usan operaciones entre matrices. En la Ecuación 1 se obtiene las corrientes en componentes de fase y para obtener las corrientes en componentes de secuencia se usa la Ecuación 2. De igual forma aplica con las tensiones, tal como se observa en las Ecuaciones 3 y 4.

$$I_{abc} = [A][I_{012}] \tag{1}$$

$$I_{012} = [A^{-1}][I_{abc}] \tag{2}$$

$$V_{abc} = [V][I_{012}] \tag{3}$$

$$V_{012} = [A^{-1}][V_{abc}] \tag{4}$$

Fallas en los Sistemas Eléctricos de Potencia debido a cortocircuitos de baja impedancia

Existen varios tipos de fallas, las cuales pueden ser analizadas con el método de las componentes simétricas.

Falla simétrica

Esta falla ocurre cuando se ponen en cortocircuito todas las fases del sistema, aunque tiene un porcentaje de probabilidad muy bajo. En la Figura 4 se puede observar la falla trifásica.

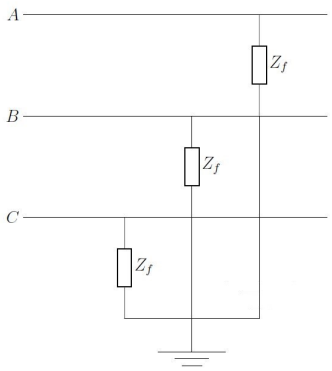


Figura 4: Falla trifásica

Fallas asimétricas

Las fallas asimétricas son las más comunes en un sistema eléctrico de potencia, donde podemos encontrar en la Figura 5 la falla monofásica a tierra(izquierda), bifásica línea-línea(centro) y bifásica a tierra(derecha).

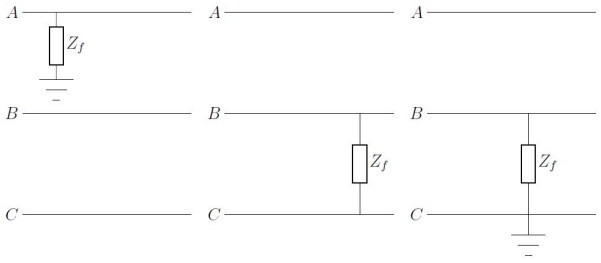


Figura 5: Falla monofásica, bifásica y bifásica a tierra

1.4. Estado del arte

Las fallas eléctricas son el principal desafío en los sistemas eléctricos de potencia, dado que son difíciles de predecir o evitar. Por lo tanto, la implementación de métodos de identificación, permiten detectarlas y clasificarlas de forma rápida, de tal manera que se puedan tomar medidas para proteger el sistema[4]. Algunos de estos métodos se presentan a continuación.

En [5] los autores implementan lógica difusa y redes neuronales para la detección, clasificación, caracterización y localización de fallas basadas en datos de sensores y medidores inteligentes instalados en la red. Este artículo hace uso de Matlab y OpenDSS de forma simultánea en el sistema IEEE de 37 nodos, donde propone una estrategia, obteniendo una precisión del 99 %, con un alto rango de efectividad, para la detección y clasificación de la falla.

Los siguientes artículos e investigaciones proponen métodos para la identificación de fallas, haciendo uso de diferentes herramientas computacionales, los cuales son:

En [6] el autor implementa una técnica la cual extrae las componentes simétricas de falla de las señales de tensión y corriente, luego clasifica la falla haciendo uso de la secuencia cero y negativa y para aumentar la precisión de las fallas a tierra, establece un criterio basado en las componentes de secuencia. Este artículo se hace uso del software EMTP/ATP para simular el sistema y evaluar el algoritmo propuesto, el cual identifica todos los tipos de fallas en líneas de transmisión de forma efectiva.

En [7] los autores usan la combinación de la transformada discreta de wavelet y lógica difusa para la identificación de varios tipos de fallas en sistemas eléctricos no balanceados. Este artículo hace uso de Matlab en un sistema IEEE de 13 nodos y de algunos datos de fallas del sistema eléctrico de distribución de la ciudad de Janpur en India, obtenidos por la compañía MIPOWER, alcanzando una precisión del 95 % para la clasificación de las fallas.

En [8] los autores implementan una metodología que combina una red neural auto asociativa y sistemas de inferencia neurodifusos adaptativos para la detectar, identificar y determine el instante en que ocurre la falla. Este articulo hace uso de herramientas computacionales y se aplicada en el sistema IEEE de 24 nodos, obteniendo una precisión superior al 86 %.

En [9] los autores usan el algoritmo k-NN o conocido como el método de los k vecinos más cercanos, para la detección y clasificación de fallas, simulando variaciones de la resistencia de falla, ángulos de inicio de fallas, relaciones X/R, niveles de cortocircuito, ángulos de carga de la fuente y ubicaciones de las fallas. Este artículo hace uso de Matlab y Simulink en un sistema de 5 nodos, donde se prueban diferentes tipos de fallas y se calcula el error con datos previamente definidos de 6 métodos de clasificación existentes.

En [10] los autores proponen la ejecución conjunta del análisis discriminante lineal y el algoritmo de bosque aleatorio basado en el proceso de aprendizaje del optimizador de sepia que permite detectar y clasificar fallas en la línea de transmisión en un sistema de potencia. Este artículo hace uso de Matlab y Simulink en un sistema de prueba donde se extraen los parámetros del flujo de potencia, se evalúan con los algoritmos anteriormente mencionados, y se determina la falla en el sistema. Además, analiza y compara los resultados con otras técnicas existentes como son algoritmo de bosque aleatorio, red neuronal artificial, entre otras.

2. Metodología

En la presente pasantía se desarrollaron una serie de actividades para dar cumplimiento a los objetivos propuestos, las cuales son:

2.1. Fase 1: Busqueda de documentos

Esta fase abarca la investigación documental sobre el software OpenDSS, donde se prioriza sobre las principales características, los posibles estudios que puede realizar, como es su interfaz, la interpretación de los códigos para generar algoritmos aplicados a la simulación de sistemas, visualización de resultados a través de archivos con extensión .TXT y .CSV, exportar datos para posteriores análisis y el modo de realizar el enlace con Matlab para trabajar simultáneamente ambos software. Además se hizo la búsqueda de artículos, trabajos de grado y revistas científicas relacionadas con el desarrollo de técnicas aplicadas para la identificación de fallas en sistemas eléctricos de potencia.

2.2. Fase 2: Aprendizaje del software OpenDSS

En la presente fase se hace la revisión de los documentos obtenidos en la investigación de la fase 1 y se selecciona aquellos que traten sobre el uso del software OpenDSS. Posteriormente se inicia la interacción con el software OpenDSS para hacer un reconocimiento de su interfaz, las opciones que dispone para el usuario, establecer los parámetros acorde a la información recolectada para modelar un sistema eléctrico de potencia, realizar flujos de cargas y modelar fallas de baja impedancia mediante algoritmos. Por último se analiza los datos de salida, según el estudio realizado, para evidenciar los datos de corrientes y tensiones en componentes de fase,secuencia y potencia.

2.3. Fase 3: Desarrollo del módulo de identificación de fallas para el sistema IEEE de 13 y 34 nodos en Appdesigner

En la fase 3 inicialmente se implementa el enlace de OpenDSS con Matlab para ejecutar los códigos de los sistemas IEEE de 13 y 34 nodos desde Matlab. Estos códigos son extraídos de la aplicación APP LF-UD correspondiente a la localización de fallas. Escrito el código en Matlab, se hace una serie de pruebas para cargar los sistemas escritos en OpenDSS desde diferentes rutas, verificando el correcto enlace entre ambos software. Posteriormente se desarrolla un algoritmo para clasificar e identificar la falla propuesta, donde se valida los resultados a partir de los resultados de las corrientes de falla y se hacen los ajustes si son necesarios. Posteriormente se implementa en Appdesigner los módulos aplicados a la identificación de fallas en la aplicación APP LF-UD. Se anexa un nuevo módulo para la selección de la línea y se implementa la opción de modificar sus parámetros como son resistencia y reactancia en componentes de secuencia y la longitud. Por último, se hará un archivo ejecutable la aplicación y se harán las pruebas respectivas validando su correcto funcionamiento.

2.4. Fase 4: Material académico

La última etapa consiste en realizar una serie de vídeos sobre el modelamiento del sistema IEEE de 13 nodos en el software OpenDSS. Estos vídeos serán de corta duración, los cuales abarcarán desde qué es OpenDSS, que tipo de simulaciones son posible hacer en el software, como descargarlo y su modo de instalación, reconocimiento de la interfaz, modelamiento de los principales elementos, flujo de carga y modelamiento de fallas de baja impedancia en el sistema eléctrico de potencia. Se desarrollará un manual relacionado con el uso del software de OpenDSS y su respectivo enlace con Matlab, describiendo elementos y mostrando ejemplos de 5 sistemas diferentes. Por último se complementarán los manuales de la aplicación APP LF-UD, anexando las características de los módulos de identificación de fallas para los sistemas IEEE de 13 y 34 nodos.

3. Resultados

A través de la serie de actividades desarrolladas, se tienen los siguientes resultados:

3.1. Módulos de identificación de fallas de baja impedancia en el sistema IEEE de 13 y 34 nodos desarrollado en Appdesigner

Se complemento la aplicación APP LF-UD elaborado por el grupo de investigación SIREC de la Universidad Distrital Francisco José de Caldas desarrolla en Matlab y Appdesigner enlazada con el software OpenDSS, donde inicialmente este programa tiene un modulo de localización de fallas de baja impedancia implementando 2 métodos para un sistema IEEE de 13 nodos y un sistema IEEE de 34 nodos. Este programa calcula un valor aproximado de localización de la falla, ajustando algunos parámetros del sistema, con la posibilidad de exportar gráficos, acorde a los resultados obtenidos.

El módulo implementado en el presente proyecto se desarrolló para el grupo de investigación GISPUD y posee el nombre de "Identificación de Fallas". En este modulo se implementó un algoritmo para clasificar la falla a partir de ejecutar una simulación, donde inicialmente se debe establecer unas condiciones iniciales para obtener unos resultados. Se implementó el enlace desde la aplicación con OpenDSS, ingresar generación distribuida, la opción de seleccionar la línea en falla, variar el porcentaje de longitud donde ocurre la falla, la opción de agregar un valor de resistencia de falla, seleccionar el tipo de falla si es simétrica o asimétrica, seleccionar la ruta del archivo base del sistema IEEE tanto para el de 13 nodos como para el 34 nodos creado previamente en OpenDSS y por último una sección para mostrar los resultados, como son corrientes en componentes de fases con su respectivo ángulo y un diagrama fasorial, que varía acorde a la falla propuesta. En las Figuras 6 y 7 se observa los módulos de identificación de fallas aplicados para los los sistemas de 13 y 34 nodos.

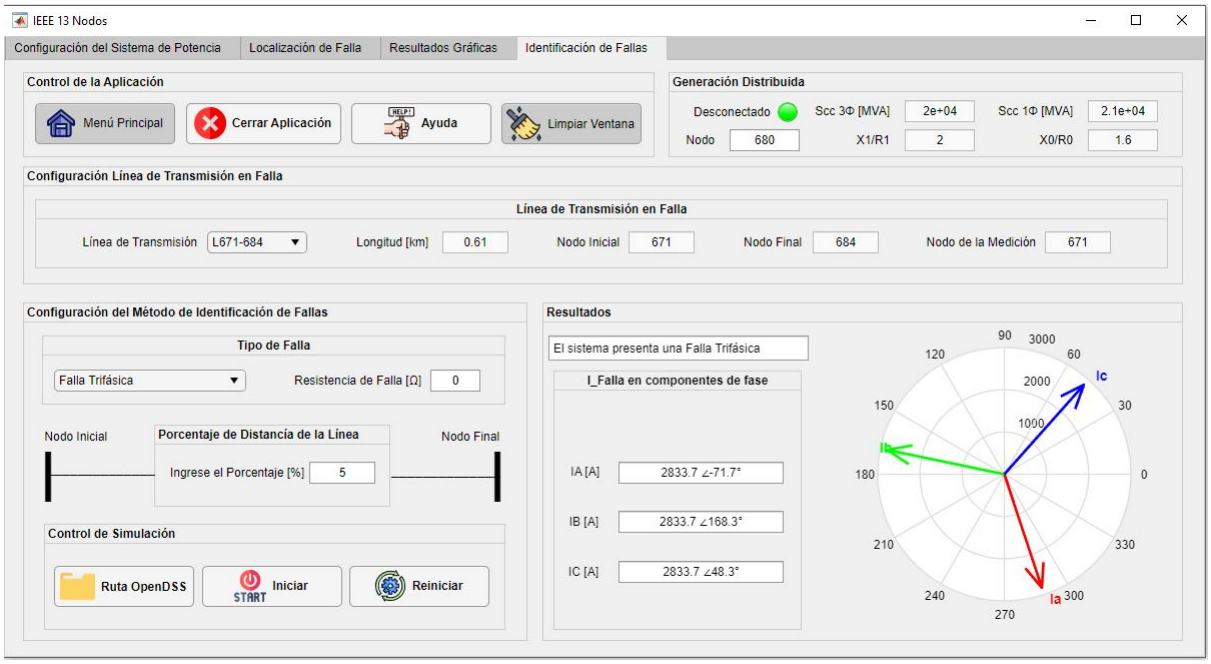


Figura 6: Modulo de identificación de Fallas para el Sistema IEEE 13 Nodos

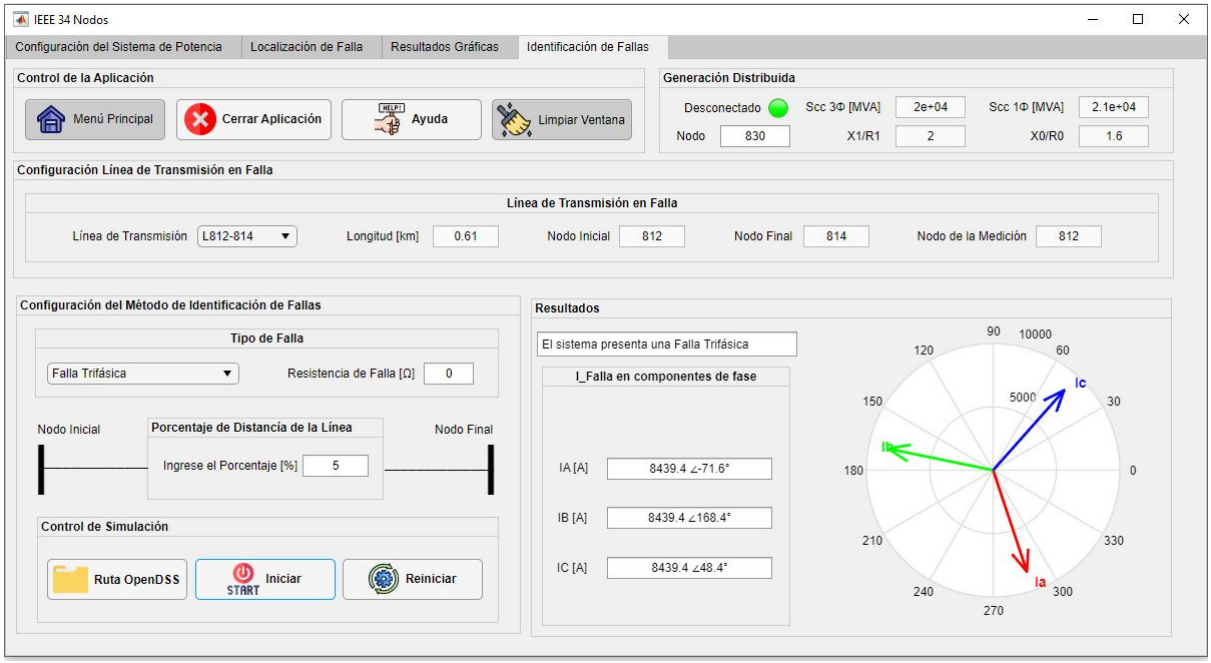


Figura 7: Modulo de identificación de Fallas para el Sistema IEEE 34 Nodos

En la sección de Configuración del Sistema de Potencia se agregó la opción de modificar los parámetros iniciales como son resistencia y reactancia para la longitud y la secuencia positiva y cero en el modelo serie de la línea, tal como se puede ver en la Figura 10. Estos valores pueden ser cargados por línea y el sistema pueden ser nuevamente ejecutado, para evidenciar las variaciones en las corrientes de falla que puede haber en el sistema. Esta opción se configuró para los sistema IEEE de 13 y 34 nodos como se puede ver en las Figuras 8 y 9.

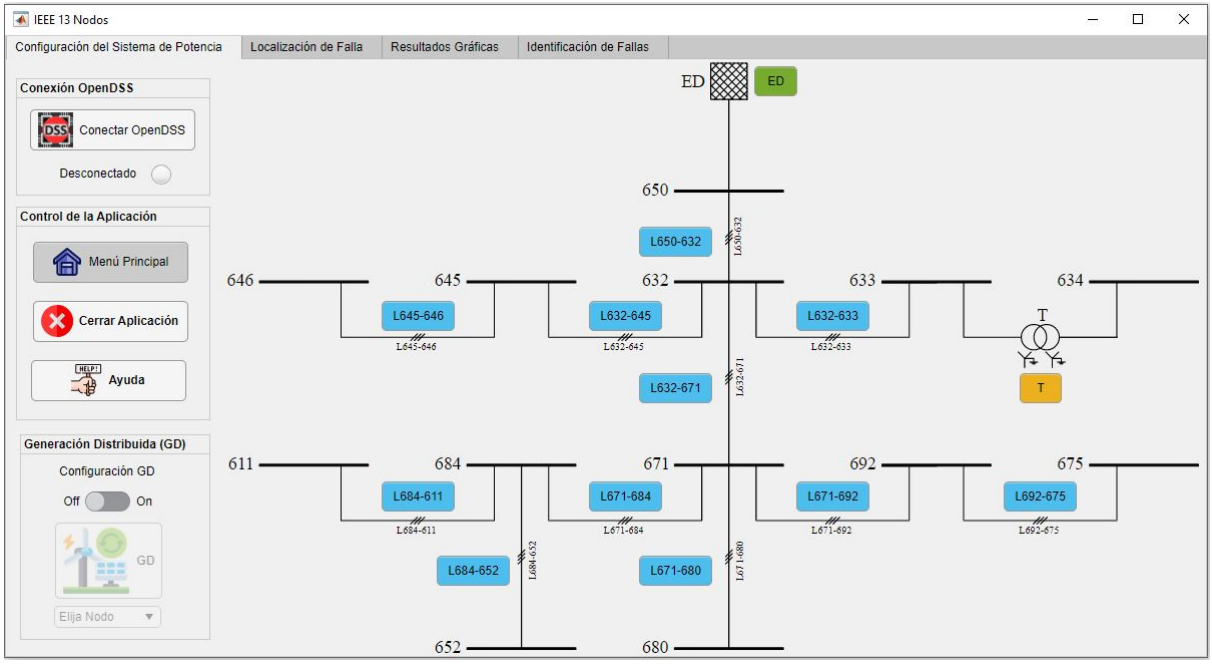


Figura 8: Modulo Configuración de Fallas para el Sistema IEEE 13 Nodos

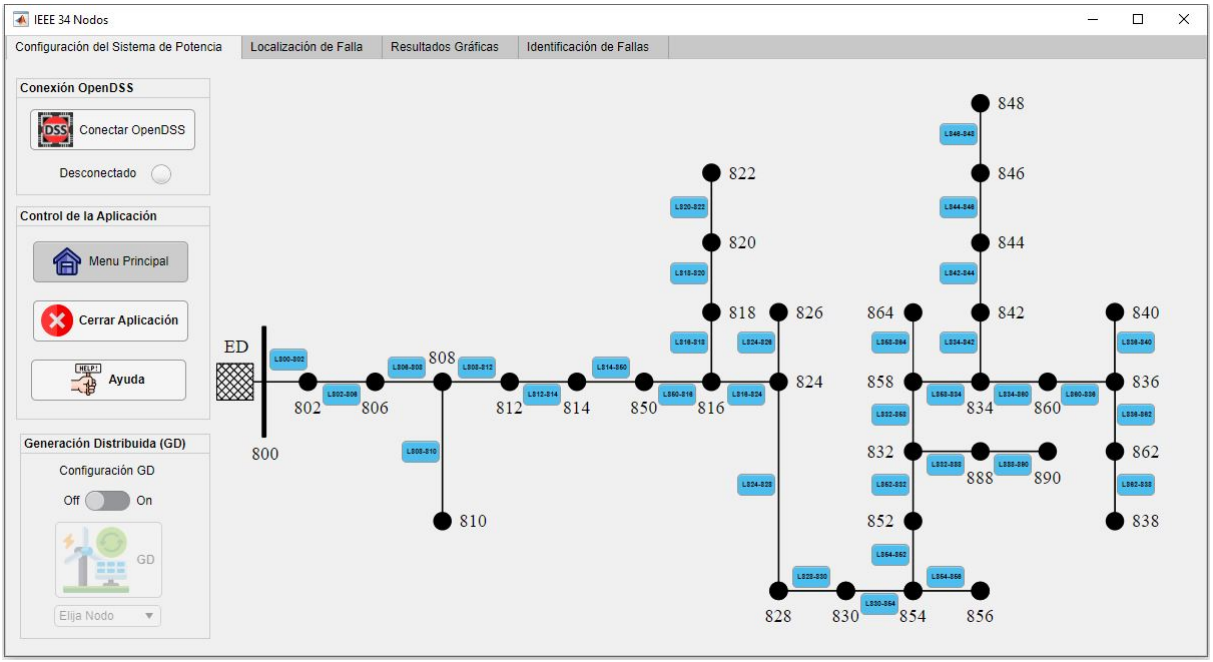


Figura 9: Modulo Configuración de Fallas para el Sistema IEEE 34 Nodos

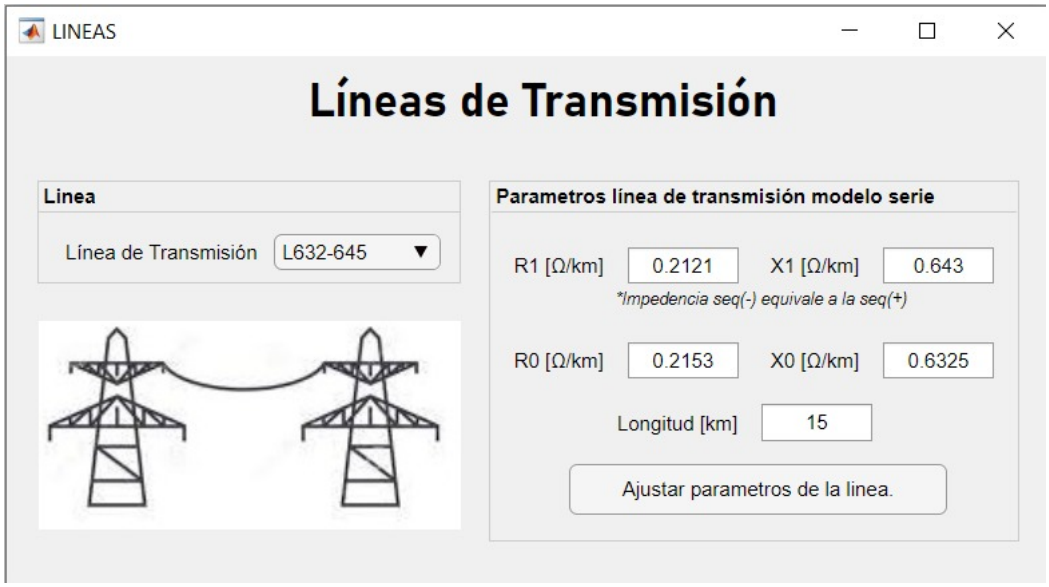


Figura 10: Módulo de configuración de parámetros de la línea

3.2. Comprobación de resultados para la identificación de fallas

El módulo de identificación de fallas en la aplicación APP LF-UD está en la capacidad de determinar si la falla ocurrida en algún punto de la red es monofásica, bifásica, bifásica a tierra y trifásica para el sistema IEEE de 13 y 34 nodos. Para la comprobación de resultados, en la Tabla 1 se muestran los datos de la aplicación APP LF-UD usando el sistema IEEE de 13 nodos con las 4 fallas propuestas, en diferentes líneas de transmisión con una variación porcentual en dónde ocurre la falla y se programa de la misma forma las diferentes fallas directamente en OpenDSS, comprobando que la aplicación APP LF-UD determina de manera correcta las corrientes de falla e indica el tipo de la falla que ha ocurrido.

Línea en falla	Tipo de falla	Corrientes de falla en APP LF-UD en [A]	Corrientes de falla en OpenDSS en [A]
L632-633 con localización del 20 % del nodo de medición	Falla monofásica	I[A]=4859.2$\angle -71{,}5°$ I[B]=0.0016644$\angle -20{,}1°$ I[C]=0.00016713$\angle -159{,}9°$	I[A]=4859.2$\angle -71{,}5°$ I[B]=0.0016644$\angle -20{,}1°$ I[C]=0.00016713$\angle -159{,}9°$
L671-684 con localización del 30 % del nodo de medición	Falla bifásica	I[A]=0.0056337$\angle 90°$ I[B]=2187.6$\angle -161{,}7°$ I[C]=2187.6$\angle 18{,}3°$	I[A]=0.0056337$\angle 90°$ I[B]=2187.6$\angle -161{,}7°$ I[C]=2187.6$\angle 18{,}3°$
L671-680 con localización del 50 % del nodo de medición	Falla bifásica a tierra	I[A]=0.0015565$\angle 89{,}8°$ I[B]=2335.7$\angle 168{,}1°$ I[C]=2322.9$\angle 48{,}6°$	I[A]=0.0015565$\angle 89{,}8°$ I[B]=2335.7$\angle 168{,}1°$ I[C]=2322.9$\angle 48{,}6°$
L645-646 con localización del 15 % del nodo de medición	Falla trifásica	I[A]=2702$\angle -71{,}7°$ I[B]=2702$\angle 168{,}3°$ I[C]=2702$\angle 48{,}3°$	I[A]=2702$\angle -71{,}7°$ I[B]=2702$\angle 168{,}3°$ I[C]=2702$\angle 48{,}3°$

Tabla 1: Comprobación de resultados de APP LF-UD y OpenDSS para el sistema IEEE de 13 nodos

Para cada tipo de falla, la teoría indica lo siguiente:

- En la falla monofásica para la fase A, la magnitud de corriente en componentes de fase está en el orden de [kA],en cambio las magnitudes de corrientes de las fases B y C no son comparables con la fase A dado que están el el orden de [mA], por lo tanto estas se tomarán como cero.
- En el caso de la falla bifásica, la corriente en componentes de fase en las fases B y C tendrá igual magnitud, pero la fase C, será el negativo de la fase B, haciendo un desplazamiento de 180°. Para la fase A, la magnitud de corriente no será comparable con las otras fases, por lo tanto se tomará como cero.
- Para la falla bifásica a tierra, en el caso de la fase A, la magnitud de corriente en componentes de fase no será comparable respecto a las corrientes de falla para las fases B y C, por lo cual se tomará como cero. En cambio, la magnitud de corriente para las fases B y C tendrán un valor en el orden de [kA] y el desfase angular no estará a 180° como la falla bifásica .
- Por último la falla trifásica, para la fases A,B y C se tendrá un mismo valor en su magnitud de corriente, pero habrá un desplazamiento angular de 120° por cada fase.

En el sistema IEEE de 34 nodos se aplicó el mismo procedimiento, variando las fallas, las líneas de transmisión y la localización, donde se obtuvo unos resultados, comprobando cada una de las fallas y calculando de manera correcta las corrientes en componentes de fase.

3.3. Manual sobre el uso de los módulos de identificación de Fallas de baja impedancia

La aplicación APP LF-UD posee una serie de manuales de usuario para los sistemas IEEE de 13 y 34 nodos donde explican el modulo de localización de fallas de baja impedancia, la instalación de la aplicación APP LF-UD, un manual de instalación de OpenDSS y por último un manual de usuario para el menú principal. Se ha anexado en los manuales, el manejo del nuevo módulo de identificación de fallas, donde se explica la posibilidad de seleccionar la línea en falla, modificar los parámetros de la línea, ingresar la resistencia de falla y la sección donde muestra los resultados como son corrientes en componentes de fase con su ángulo, un diagrama fasorial representando las magnitudes de las corrientes anteriormente calculadas y la confirmación de la falla establecida.

3.4. Elaboración de un manual sobre el uso de OpenDSS y su enlace con Matlab.

Se escribió un manual relacionado con el uso del software OpenDSS para la simulación de sistemas eléctricos de potencia, donde se explica inicialmente que es OpenDSS, dónde se descarga, reconocimiento de la interfaz,

sección de ayudas y los códigos requeridos y la forma de ingresarlos para el modelamiento de elementos básicos, ejecución del código y la forma de mostrar resultados . También se explican 5 ejercicios aplicados al flujo de carga, modelamiento de la línea y cálculo de fallas simétricas y asimétricas. Por último, se anexa una sección en donde se trata la implementación de OpenDSS con Matlab realizando su respectivo enlace. El manual se presenta a continuación:



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

FACULTAD TECNOLÓGICA

INGENIERÍA ELÉCTRICA POR CICLOS PROPEDÉUTICOS

MANUAL DE USUARIO PARA EL SOFTWARE OPENDSS

AUTORES

Julio Ernesto Castro Rico
Diego Alberto Anzola Bustos

BOGOTÁ D.C

Índice

Índice de figuras **3**

Índice de cuadros **3**

1. Conociendo OpenDSS **1**
1.1. ¿Qué es OpenDSS? . . . 1
1.2. Modo de instalación . . . 1
1.3. Conociendo la interfaz . . . 2

2. Sección de ayudas **2**

3. Modelos de los elementos del sistema eléctrico de potencia **3**
3.1. Modelamiento del generador o equivalente de red . . . 3
3.2. Modelamiento del generador de potencia controlada . . . 4
3.3. Modelamiento de la máquina de inducción . . . 5
3.4. Modelamiento del transformador . . . 5
3.5. Modelamiento del transformador regulador . . . 6
3.6. Modelamiento de la línea . . . 6
3.6.1. Forma 1: Conociendo la resistencia, reactancia y capacitancia. . . . 6
3.6.2. Forma 2: Conociendo la matriz de impedancias . . . 7
3.6.3. Forma 3: Geometría de la línea . . . 7
3.6.4. Línea transpuesta y no transpuesta con haz de conductores . . . 8
3.7. Modelamiento del capacitor . . . 9
3.7.1. Control del capacitor . . . 10
3.8. Modelamiento de la carga . . . 10
3.8.1. Modelamiento de cargas estáticas . . . 10
3.8.2. Modelamiento de cargas variables . . . 11
3.9. Modelamiento del generador eólico . . . 11
3.10. Modelamiento del generador fotovoltaico . . . 11
3.11. Monitor . . . 12
3.12. Modelamiento de fallas de baja impedancia . . . 12
3.12.1. Falla monofásica . . . 12
3.12.2. Falla bifásica . . . 13
3.12.3. Falla bifásica a tierra . . . 13
3.12.4. Falla trifásica . . . 14
3.13. Modo de ejecutar el código . . . 15
3.14. Modo de exportar . . . 15
3.15. Mostrar resultados . . . 16
3.16. Comentar texto . . . 17

4. Ejemplos **18**
4.1. Ejemplo 1: Circuito básico de 3 nodos. . . . 18
4.1.1. Modelamiento del generador . . . 18
4.1.2. Modelamiento del transformador . . . 18
4.1.3. Modelamiento de la línea . . . 19
4.1.4. Modelamiento de la carga . . . 19
4.1.5. Mostrar resultados . . . 19
4.1.6. Ejecución del código . . . 19
4.2. Ejemplo 2: Generador Slack y PV. . . . 21
4.2.1. Modelamiento del generador 1 ó nodo de referencia . . . 21
4.2.2. Modelamiento de las líneas de transmisión . . . 21
4.2.3. Modelamiento del generador de potencia controlada . . . 22
4.2.4. Modelamiento de las cargas . . . 22
4.3. Ejemplo 3: Generación distribuida. . . . 23
4.3.1. Generador . . . 23
4.3.2. Algoritmo para las curvas de carga . . . 23
4.3.3. Algoritmo para la impedancia de las líneas de transmisión . . . 24
4.3.4. Conexión de las líneas de transmisión . . . 24

4.3.5. Transformador . . . 24
4.3.6. Transformador regulador . . . 25
4.3.7. Cargas . . . 25
4.3.8. Capacitor . . . 25
4.3.9. Control del capacitor . . . 26
4.3.10. Generador eólico . . . 26
4.3.11. Monitor . . . 26
4.3.12. Ejecución del Código . . . 26
4.4. Ejemplo 4: Línea no transpuesta . . . 27
4.4.1. Definiendo el nuevo circuito . . . 27
4.4.2. Características de los conductores . . . 27
4.4.3. Posición de los conductores de fase y guarda . . . 28
4.4.4. Matriz de impedancia . . . 28
4.4.5. Resultados . . . 29
4.5. Ejemplo 5: Modelamiento de falla monofásica, bifásica, bifásica a tierra y trifásica. . . . 30
4.5.1. Modelamiento del Equivalente de Red . . . 30
4.5.2. Modelamiento de las líneas de transmisión . . . 31
4.5.3. Modelamiento del Transformador . . . 31
4.5.4. Modelamiento de las Fallas de baja impedancia . . . 32
4.5.5. Definición de Tensión base y Solución del Sistema . . . 32
4.5.6. Mostrar Resultados . . . 32

5. Enlace con Matlab **34**
5.1. Conexión de OpenDSS y Matlab . . . 34
5.2. Vinculación del circuito de OpenDSS . . . 34
5.3. Mostrar resultados . . . 34
5.4. Exportar datos . . . 34
5.5. Modificar datos de un circuito de OpenDSS desde Matlab . . . 35

6. Otros recursos **35**

7. Créditos **36**

Referencias **36**

Índice

Índice de figuras

1. Página principal de OpenDSS . . . 1
2. Selección para descargar OpenDSS . . . 1
3. Página para descargar OpenDSS . . . 2
4. Ventana de inicio del software OpenDSS . . . 2
5. Selección de ayudas. . . . 3
6. Ventana de ayuda de OpenDSS . . . 3
7. Tipos de configuraciones del capacitor [4] . . . 9
8. Triangulo de potencia . . . 10
9. Modo de ejecución del código . . . 15
10. Ventana de exportación en OpenDSS . . . 15
11. Ruta de generación de archivos .CSV en OpenDSS . . . 16
12. Explorador de archivos. . . . 16
13. Archivo .CSV generado por OpenDSS . . . 16
14. Ventana para mostrar datos en OpenDSS . . . 17
15. Circuito Básico de 3 nodos . . . 18
16. Modo de ejecución del ejemplo 1 . . . 20
17. Resultados del Ejemplo 1 . . . 20
18. Sistema Eléctrico del ejemplo 3 . . . 21
19. Circuito aplicando generación distribuida . . . 23
20. Configuración del circuito sencillo con haz de conductores . . . 27
21. Matriz de impedancia para la línea no transpuesta . . . 29
22. Esquema del sistema IEEE de 13 Nodos . . . 30
23. Corrientes de falla trifásica en el sistema IEEE de 13 nodos . . . 33

Índice de cuadros

1. Tabla de conductores para líneas aéreas en Aluminio/Acero . . . 27

1. Conociendo OpenDSS

1.1. ¿Qué es OpenDSS?

Según EPRI "Electric Power Research Institute" indica que OpenDSS es un simulador de sistemas de distribución de energía eléctrica diseñado para apoyar la integración de los recursos de energía distribuida y la modernización de la red[1].
Dentro de los estudios que permite realizar OpenDSS encontramos lo siguiente:

- Estudios de armónicos
- Impactos de la generación fotovoltaica
- Estudios de cortocircuito
- Flujos de carga
- Análisis estocásticos y probabilísticos
- Entre otros

1.2. Modo de instalación

Si se quiere instalar OpenDSS, inicialmente se debe dirigir al siguiente enlace `https://www.epri.com/pages/sa/opendss`, el cual lo dirige a la página principal de OpenDSS, tal como se puede ver en la Figura 1.

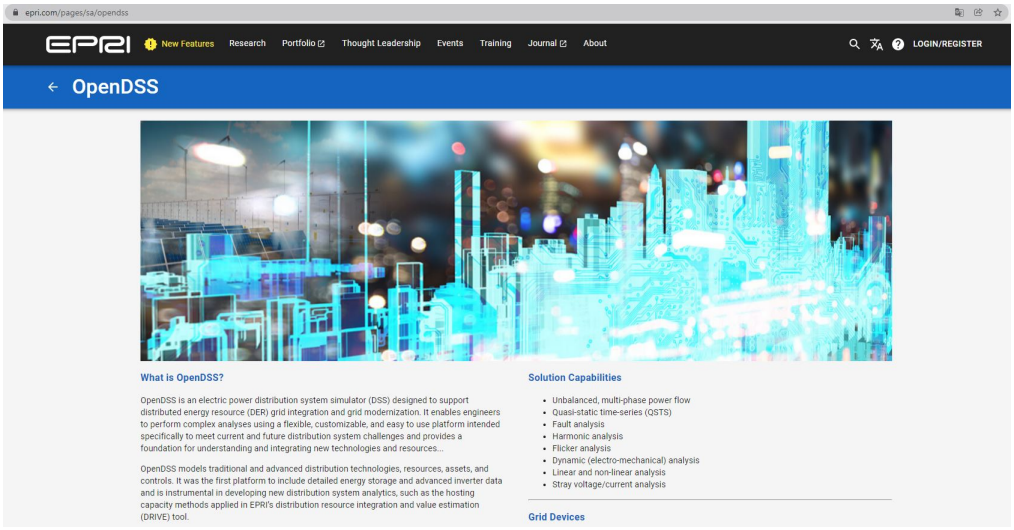


Figura 1: Página principal de OpenDSS

En esta página se busca la opción "Download tool" y se selecciona como se puede observar en la Figura 2.



Figura 2: Selección para descargar OpenDSS

Posteriormente se enlaza a otra página llamada “SOURCEFORGE”, como se puede ver en la Figura 3. En esta página se selecciona la opción “Download” y automáticamente se descargará el software.

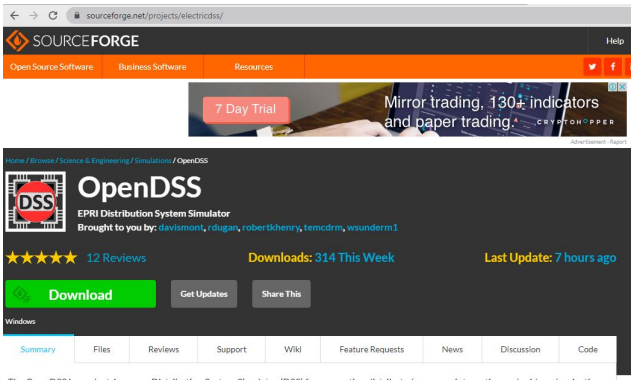


Figura 3: Página para descargar OpenDSS

Descargado el archivo, se selecciona y se procede a instalar siguiendo los pasos indicados.

1.3. Conociendo la interfaz

Al iniciar el software OpenDSS, se encuentra en la parte central un recuadro blanco grande. En este recuadro es donde se escribirá los códigos del sistema. En la parte izquierda hay un recuadro de forma vertical, el cual muestra si al compilar el código fue resuelto de manera satisfactoria o si no fue resuelto por tema de algún error en el código. En la parte superior se encuentra la sección de configuraciones, opción para guardar, mostrar resultados, ayudas, etc. Por último en la parte de abajo en el recuadro que dice “Messages” mostrará donde quedó guardado el proyecto y de igual manera todos los archivos generados en esa misma ruta.

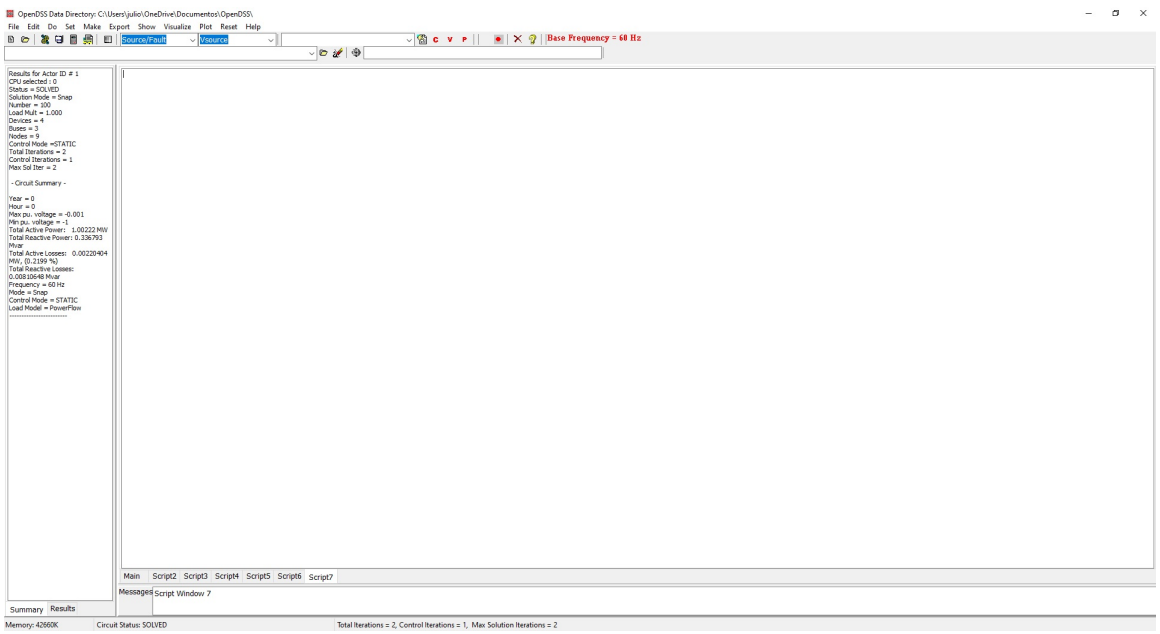


Figura 4: Ventana de inicio del software OpenDSS

2. Sección de ayudas

OpenDSS tiene una sección de ayudas, donde explica los elementos que conforman un sistema de potencia. Para esto en la Figura 5 se busca la opción que dice “Help” y se selecciona la opción “DSS Help” que tiene un signo de interrogación de color amarillo.

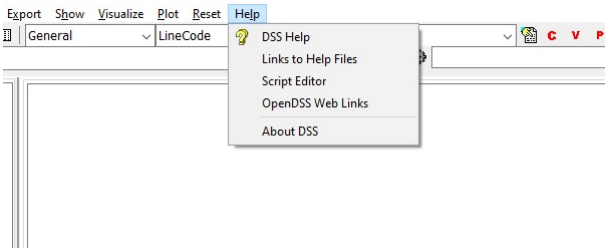


Figura 5: Selección de ayudas.

Seleccionado el signo de interrogación, se desplegará una ventana como se ve en la Figura 6. Aquí se encuentra la explicación de cada uno de los comandos y elementos para modelar el sistema. Este manual no abordará todos y sólo explicará los elementos básicos.

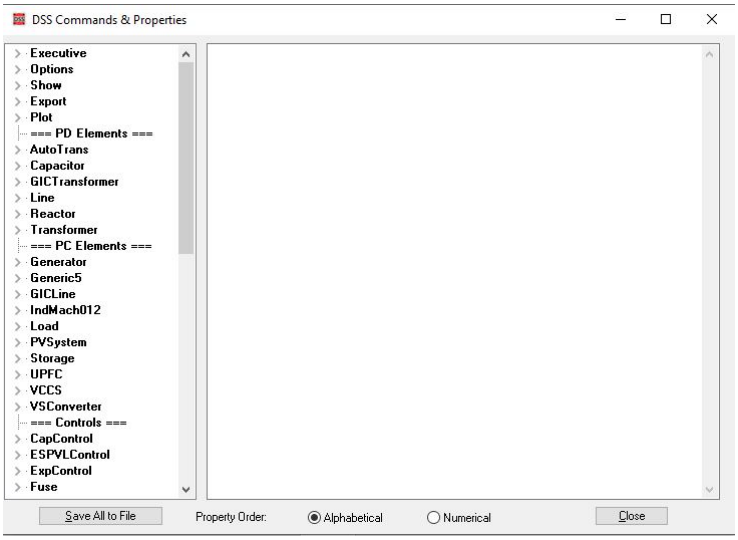


Figura 6: Ventana de ayuda de OpenDSS

En la Figura 5 también sale otras opciones de ayuda de forma online. Por último sale la opción de la versión del OpenDSS que está instalada.

3. Modelos de los elementos del sistema eléctrico de potencia

Para modelar un sistema de potencia básico se debe tener presente algunos datos para cada uno de los elementos que conforman el sistema. El orden de los comandos no se debe cambiar dado que puede presentarse errores al compilar. Además hay datos que se pueden omitir porque ya están establecidos por defecto.

3.1. Modelamiento del generador o equivalente de red

Para modelar el generador o equivalente de red se deben tener algunos datos básicos:

- Bus1 = Es el nodo en el cual va a estar conectado la referencia del sistema.
- basekv = Tensión base línea a línea en kV.
- pu = Valor en por unidad en la cual está operando el generador.
- Angle = Ángulo en grados que tiene la primera fase.
- Frequency = Frecuencia en Hz general del sistema. Por defecto OpenDSS trabaja a 60 Hz.
- Phases = Número de fases que tiene el sistema. Podemos definir un sistema trifásico o un sistema monofásico.
- Mvac3 = Potencia de cortocircuito trifásica. Por defecto su valor se escribe en VA.

- Mvac1 = Potencia de cortocircuito monofásica. Por defecto su valor se escribe en VA.
- x1r1 = Relación que hay entre la reactancia de secuencia positiva y la resistencia en secuencia positiva x1/r1. OpenDSS por defecto tiene el valor de 4.0.
- x0r0 = Relación que hay entre la reactancia de secuencia cero y la resistencia en secuencia cero x0/r0. OpenDSS por defecto tiene el valor de 3.0.
- Isc3 = Corriente de cortocircuito trifásica en amperios. Por defecto su valor se escribe en amperios.
- Isc1 = Corriente de cortocircuito monofásica en amperios. Por defecto su valor se escribe en amperios.
- R1 = Resistencia en secuencia positiva.
- X1 = Reactancia en secuencia positiva.
- R0 = Resistencia en secuencia cero.
- X0 = Reactancia en secuencia cero.

Los datos mínimos parta modelar un equivalente de red o un generador son:

- Bus1
- basekv
- pu
- Angle
- Frequency
- Phases
- Mvac3 o ISC3
- Mvac1 o ISC1
- x1r1
- x0r0

Los demás datos son complementarios y se ingresarán si el estudio a realizar los requiere.

3.2. Modelamiento del generador de potencia controlada

Para ingresar un generador de potencia controlada "PV", se ingresarán los siguientes datos:

- phases = Número de fases del generador.
- bus = Nodo en que está conectado el generador.
- kV = Tensión nominal del generador.
- kW = Potencia activa del generador.
- model = Es un valor entero que va desde el 1 hasta el 7 y se usa para la variación de la generación con el voltaje. Las mas comunes son: 1 es porque sólo se inyecta un kW constante, acorde al factor de potencia especificado, 2 si el generador se modela como una admitancia constante, 3 si kW y kV son constantes (se conoce también como generador P-V), 4 si kW es constante y Q se mantiene fija y 5 si kW es constante y Q como una reactancia constante.
- Vpu = Valor en por unidad del generador.
- maxkvar = Potencia reactiva máxima que puede entregar el generador.
- minkvar = Potencia reactiva mínima que puede entregar el generador.
- PVFactor = Corresponde al factor de desaceleración para el modelo de generador P-V (modelo 3). Su valor por defecto es 0.1.

3.3. Modelamiento de la máquina de inducción

Para modelar una máquina de inducción se deben escribir los siguientes comandos:

- Phases = Número de fases de la máquina de inducción. Por defecto su valor es 3.
- bus1 = Nodo en el cual se encuentra conectada la máquina.
- kv = Tensión nominal de la máquina. Si la conexión es en estrella se indica tensión fase-tierra. Si la tensión es en delta se indica la tensión fase-fase.
- kW = Potencia activa de la máquina.
- Pf = Factor de potencia cuando la máquina está a plena carga (valor nominal).
- Conn = Conexión del estator que puede ser en delta o estrella. Por defecto OpenDSS lo toma en delta y no se debe ingresar el comando. Si se requiere en estrella, se coloca el comando "wye".
- kVA = Potencia aparente de la máquina.

3.4. Modelamiento del transformador

El software permite modelar transformadores de dos o tres devanados, independientemente de si su conexión sea en estrella o delta. Para este caso se explicará el modelamiento de un transformador de dos devanados. Algunos de los comandos requeridos son:

- Phases = Número de fases que tiene el transformador. Por defecto OpenDSS maneja 3 fases pero también se puede trabajar con una fase si el transformador es monofásico.
- Windings = Número de devanados que tiene el transformador. Por defecto este tiene dos.
- Wdg = Corresponde al devanadado del transformador. Si es el devanado primario se escribe "wdg=1" y si es el devanado secundario "wdg=2".
- Bus1 = Nodo en el cual estará conectado el transformador tanto en el primario como en el secundario.
- Conn = Corresponde al tipo de conexión tanto del primario como del secundario. Por defecto estará en Y solidamente puesto a tierra, pero si se quiere establecer el tipo de conexión dentro de una llave, se escribe la conexión. Si por ejemplo es Y-Delta el comando es "Conns=[Wye Delta]" y así sucesivamente para las otras conexiones.
- kv = Nivel de tensión en kV que hay entre el primario y el secundario. Para definir los niveles de tensión escribimos el comando kv=[115 13.2]. En este caso tenemos en el primario 115 kV y en el secundario 13.2 kV.
- kva = Potencia aparente en kva que tiene el transformador. Se debe declarar para el devanado primario y el devanado secundario.
- %R = Porcentaje de pérdidas de la resistencia en los devanados del transformador.
- XHL = Porcentaje de reactancia que hay de un devanado hacia el otro.
- %Noloadloss = Porcentaje de pérdidas para la resistencia en la rama de magnetización.
- %imag = Porcentaje de la corriente de la rama de magnetización.

Algunos de los comandos se les puede agrega una "s" al final. Por ejemplo "Conn" se dejaría como "*Conns*" y así sucesivamente con los demás comandos, exceptuando los que ya terminan con "s". En los ejemplos se verá la forma de utilizar los comandos de esta manera. Se puede omitir datos como son los porcentajes de pérdidas en los devanados y la rama de magnetización, ya que por defecto OpenDSS tiene valores por defecto, aunque no sólo para estos, si no para muchos otros.

3.5. Modelamiento del transformador regulador

El modelamiento de un transformador regulador se usan los siguientes comandos:

- New regcontrol.Tr = Comando usado para especificar un nuevo control de transformador.
- transformer = Nombre del transformador al cual se le realizará el control de tensión.
- winding = Número de devanados del transformador que se están monitoreando.
- vreg = Ajuste que tiene en regulador en voltios. Este valor se multiplica por ptratio y se obtiene la tensión de salida del devanado controlado. Por defecto su valor es de 120.
- band = Ancho de banda en voltios para el nodo controlado. Por defecto su valor es 3.
- ptratio = Relación del PT que hay para convertir la tensión del devanado controlado a la tensión del control del regulador. Por defecto su valor es 60.
- delay = Corresponde a un retardo en el tiempo en segundos.

3.6. Modelamiento de la línea

Para modelar la línea existen tres formas:

- Ingresar la resistencia, reactancia y capacitancia de la línea.
- Ingresar la matriz Zbus del sistema.
- Ingresar la geometría de la línea para que calcule la matriz de impedancia del sistema.

3.6.1. Forma 1: Conociendo la resistencia, reactancia y capacitancia.

La primera forma de ingresar los datos de la línea es asignar algunos datos conocidos propiamente de los conductores. Algunos de los comandos son los siguientes:

- New Line.A = Forma de declarar una nueva línea entre dos nodos. Por cada nueva línea el valor en el comando "*Line*" cambia y se debe declarar posteriormente los demás comandos que están a continuación.
- bus1 = Nodo inicial donde comienza la línea.
- bus2 = Nodo final donde termina la línea.
- Linecode = Nombre que se le asigna a cada línea.
- Lenght = Distancia que tiene la línea.
- Units = Unidad en que están las medidas de la línea. Las unidades que hay son mi,km,kft,m,ft,in,cm.
- Phases = Número de fases de la línea. Por defecto OpenDSS maneja 3 fases pero se puede trabajar con una.
- R1 = Resistencia en secuencia positiva en [Ω] por unidad de longitud.
- X1 = Reactancia en secuencia positiva en [Ω] por unidad de longitud.
- R0 = Resistencia en secuencia cero en [Ω] por unidad de longitud.
- X0 = Reactancia en secuencia cero en [Ω] por unidad de longitud.
- C1 = Capacitancia en secuencia positiva en [nF] por unidad de longitud.
- C0 = Capacitancia en secuencia cero en [nF] por unidad de longitud.
- B1 = Susceptancia en secuencia positiva en [μS] por unidad de longitud.
- B0 = Susceptancia en secuencia cero en [μS] por unidad de longitud.

3.6.2. Forma 2: Conociendo la matriz de impedancias

La segunda forma es ingresando la matriz de impedancias, tal como se puede ver en la Ecuación 1. En esta matriz ya está incluidos todos los datos de las líneas, su geometría, configuración de haz, número de conductores de guarda, longitud, etc. Está matriz es del tamaño de NxN y generalmente se ingresan todos los datos que están por debajo de la diagonal de la matriz. Se debe aclarar que se debe ingresar por separado la resistencia y la reactancia. Además si el modelo es del tipo π y hay capacitancias a tierra, se puede ingresar la matriz de potencial.

$$\left[Z = R + jX\right] = \begin{bmatrix} Z_{11} + Z_g & Z_{12} + Z_g & Z_{13} + Z_g \\ Z_{21} + Z_g & Z_{22} + Z_g & Z_{23} + Z_g \\ Z_{31} + Z_g & Z_{32} + Z_g & Z_{33} + Z_g \end{bmatrix} \tag{1}$$

Ejemplo

Se requiere ingresar un sistema que está conformado por tres nodos con un conductor 336 MCM ACSR en configuración horizontal. Se ingresa las matrices con los valores de resistencia con el comando *"rmatrix"*, reactancia con el comando *"xmatrix"* y capacitancia con el comando *"cmatrix"*. A continuación se declara dos nuevas líneas con el comando con el comando *"New line."*, la primera que va desde el nodo 1 hasta el 2 y la otra que va desde el nodo 2 hasta el nodo 3. Con el comando con el comando *"linecode"* se indica de dónde viene la matriz de impedancia asociada y por último se especifíca la longitud con el comando *"lenght"*.

- New linecode.336matrix nphases=3
- ∼rmatrix = [0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455]
- ∼xmatrix = [0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 -0.2025449]
- ∼cmatrix = [2.74 | -0.70 2.96 | -0.34 -0.71 2.74]
- New line.line1 bus1=Nodo1 bus2=Nodo2 linecode=336matrix lenght=10
- New line.line2 bus1=Nodo2 bus2=Nodo3 linecode=336matrix lenght=20

3.6.3. Forma 3: Geometría de la línea

Cuando no se tiene la matriz Zbarra, se puede declarar la posición de cada conductor, teniendo como referencia los ejes X y Y del plano cartesiano tanto para los conductores de fase como para los de guarda. Los comandos necesarios son:

- set earthmodel = Corresponde al modelo de tierra del sistema. OpenDSS maneja tres tipos los cuales son Carson, FullCarson y Deri. Por defecto en los ejercicios se manejará Carson.
- New Wiredata = Nombre que se le asigna para el tipo de conductor, ya sea de fase o de guarda.
- GMR = Radio medio geométrico del conductor de fase o de guarda.
- DIAM = Diámetro del conductor de fase o de guarda.
- RAC = Resistencia eléctrica en corriente alterna del conductor de fase o de guarda. También se puede anexar un valor y un símbolo, si el número de haz de conductores es mayor o igual a dos.
- RDC = Resistencia eléctrica en corriente directa del conductor de fase o de guarda.
- Runits = Unidades en que está la resistencia eléctrica. Se puede especificar como mi,kft,km,m,Ft,in o cm.
- radunits = Unidades en que está el radio del conductor de fase o de guarda. Se puede especificar como mi,kft,km,m,Ft,in o cm.
- gmrunits = Unidades en que está el radio medio geométrico del conductor de fase o de guarda. Se puede especificar como mi,kft,km,m,Ft,in o cm.
- Nconds = Número de conductores de fase y guardas que tiene el sistema.
- Nphases = Número de fases que tiene el sistema. Por defecto su valor es 3.
- Cond = Comando que sirve apara identificar el conductor de fase.

- Wire = Tipo de conductor.
- X = Posición del conductor de fase en el eje x. Por defecto se encuentra en 0.
- H = Posición del conductor de fase en el eje y.
- Units = Unidades en que está X y H. Por defecto OpenDSS maneja las unidades en pies "ft". Si queremos otro tipo de unidades escribimos mi,kft,km,m,Ft,in o cm.
- Normaps = Corriente nominal del conductor. También se puede anexar un valor y un símbolo, si el número de haz de conductores es mayor o igual a dos.
- reduce = Corresponde si OpenDSS aplica la reducción de Kron a nuestra matriz Zbus, para disminuir su tamaño a una matriz de [3x3].
- resisitivy = Corresponde a la resistividad del terreno. Por defecto para los tres tipos de modelos de tierra se maneja 100 $\Omega - m$. Si se requiere otro valor, sencillamente se escribe el presente comando.

3.6.4. Línea transpuesta y no transpuesta con haz de conductores

Para la configuración de las líneas de transmisión, hay que tener presente si hay o no transposición de sus líneas. Para esto se debe definir como va la conexión por cada conductor de fase desde el nodo en que inicia la línea y hasta donde termina. Si por ejemplo tenemos una línea con un haz de conductores de fase pero no es transpuesta, lo escribiremos de la siguiente forma:

NewLine.circuito Bus1=Businicial.1.2.3 Bus2=Busfinal.1.2.3

Se observa que los conductores tanto para el nodo donde se inicia hasta donde terminar mantienen la misma posición. Ahora si hay dos haz de conductores por fase, se escribira como se muestra a continuación:

NewLine.circuito Bus1=Businicial.1.1.2.2.3.3 Bus2=Busfinal.1.1.2.2.3.3

Para los conductores de fase cuando la línea no es transpuesta, se aumentará los números acorde a los haz de conductores que se presenten.

Cuando hay transposición de líneas, el bus inicial se escriben las fases de forma consecutiva, pero el bus donde termina la línea se debe intercalar de la siguiente forma:

NewLine.circuito Bus1=Businicial.1.2.3 Bus2=Busfinal.2.3.1

De igual forma como en el ejemplo anterior, se escribe de la misma manera, pero se tiene presente que este modo aplica para dos haz de conductores. Si son más haz de conductores, se escribe más valores siguiendo el consecutivo que se forma.

NewLine.circuito Bus1=Businicial.1.1.2.2.3.3 Bus2=Busfinal.2.2.3.3.1.1

El anterior método aplica cuando se declara cada conductor de haz por fase si es mayor o igual a dos haz de conductores. Pero hay una forma que es más corta, declarando un sistema de más de dos haz como si fuera de uno solo haz, pero el número de haz se definirá en el comando de la resistencia eléctrica y en la corriente nominal del conductor, como se puede ver a continuación:

RAC=(0.1456 2 /) NormAmps=(951 2 *)

Se puede observar que dentro del comando RAC se indica la resistencia en corriente alterna pero al lado se coloca un 2 y un símbolo de / y en la corriente nominal NormAmps también se coloca un 2 y al lado un asterisco. Con esto definimos que es un sistema de dos haz de conductores. Si tenemos un sistema con tres haz de conductores por fase, lo escribiremos de la siguiente forma:

RAC=(0.1456 3 /) NormAmps=(951 3 *)

Se aclara que lo anterior sólo aplica para los conductores de fase, para los conductores de guarda se debe declarar sus datos individualmente ya que estos generalmente no van en haz.

3.7. Modelamiento del capacitor

El condensador es un elemento que suministra energía reactiva en el nodo que está conectado, pero si no se indica el nodo el software OpenDSS lo conectará automaticamente al nodo de referencia.

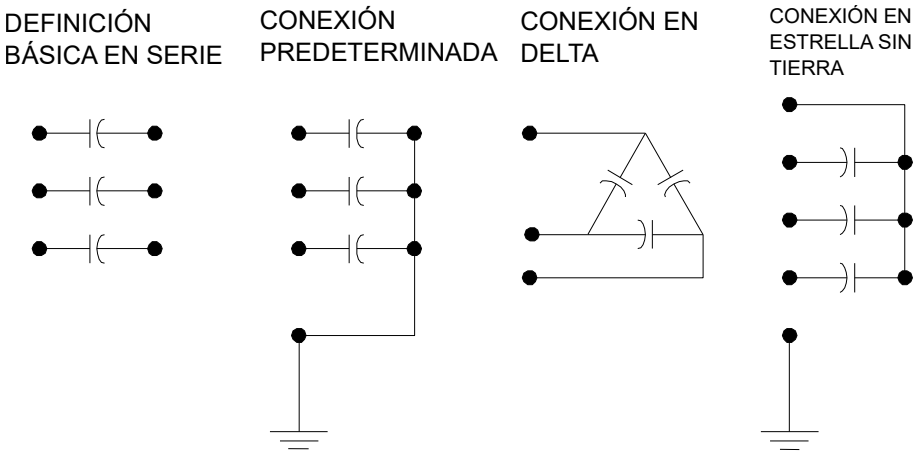


Figura 7: Tipos de configuraciones del capacitor [4]

En la Figura 7 se observa cuatro configuraciones para la conexión del banco de capacitores. En la primera opción se tiene una configuración en serie con tres condensadores, la segunda opción es una conexión en estrella sólidamente puesta a tierra, la tercera opción es una conexión en delta y la cuarta opción es una conexión en estrella sin conexión a tierra. Para la conexión en serie se debe especificar el nodo donde inicia y donde termina, para la conexión en estrella sólidamente puesta a tierra está viene por defecto, a menos que se indique otro tipo de conexión. Sin embargo podemos especificarla en donde dice tipo de conexión con el comando "wye". Para la conexión en delta sólo se especifica un nodo de conexión con el comando "delta" y por último para la conexión en estrella sin conexión a tierra se especifica de la siguiente manera:

Bus1=B1 bus2 = B1.4.4.4

Para ingresar los datos de un banco de capacitores se tiene el siguiente orden:

- new capacitor.cap1 = Comando para generar un nuevo capacitor.
- Bus1 = Nodo inicial donde se conecta el banco de condensadores.
- Bus2 = Nodo final donde se conecta el banco de condensadores. Si la conexión es en delta sólo se declara el bus1.
- Phases = Número de fases del banco de condensadores. Por defecto su valor es 3.
- Kvar = Potencia reactiva en kVAr.
- kv = Tensión nominal del capacitor en kV.
- Conn = Tipo de conexión en que está el capacitor. Si es en estrella se escribe wye o si es en Delta, se escribe la misma palabra Delta.
- Cmatrix = Es una forma alternativa para definir un banco de capacitores mediante una matriz de tamaño de 3x3 si se está manejando en el modelo de la línea capacitancias a tierra. Se declara como una matriz singular inferior pero sin los ceros en la parte superior, ya que OpenDSS completa los valores en las otras posiciones de la matriz de capacitancias.

$$\text{Cmatrix}=[c_{11}|-c_{21}\ c_{22}|-c_{31}\ -c_{32}\ c_{33}]$$

3.7.1. Control del capacitor

Para ingresar los datos de un banco de capacitores cuando es por pasos se escriben los siguientes comandos:

- New capcontrol.cap1 = Se indica con el mismo nombre del capacitor al cual se le va a realizar el control.
- element = Nombre del elemento en el que está conectado el PT. Puede ser una línea o un transformador.
- terminal = Numeral de la terminal en el que está conectado el Capcontrol. Por defecto es 1.
- capacitor = Se indica con el mismo nombre del capacitor al cual se le va a realizar el control.
- type = Tipo de control que hay. Los comandos que hay son Current, voltage, kvar,PF y time.
- ctratio = Relación que tiene el TC de la corriente de línea al ajuste de la corriente de control para los tipos de control de corriente y kVAr.
- OnSetting = Valor que indica el encendido del condensador o el aumento de los pasos.
- OFFsetting = Valor que indica el apagado del condensador.
- delay = Valor que corresponde al retardo en el tiempo. Su valor por defecto es 15.

3.8. Modelamiento de la carga

Para modelar una carga en un sistema se puede de tres maneras: la primera opción es ingresar la potencia activa y el factor de potencia. La segunda opción es ingresar la potencia activa y la potencia reactiva y la tercera opción es ingresar la potencia aparente y el factor de potencia. En resumen las opciones anteriormente mencionadas son las siguientes:

- Opción 1: kW,fp
- Opción 2: kW,kVAr
- Opción 3: kVA,fp

El software automáticamente calcula los otros datos de la carga acorde con el triangulo de potencia, tal como se ve en la Figura 8.

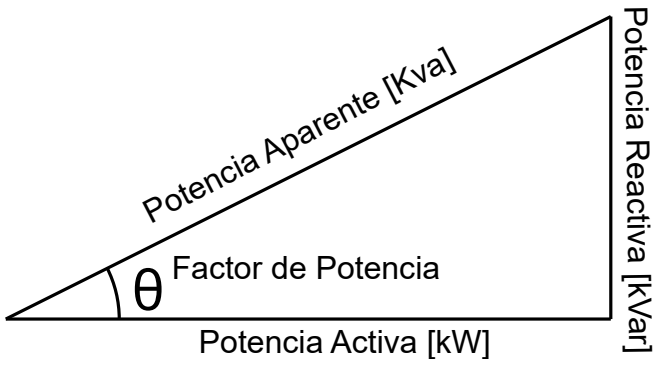


Figura 8: Triangulo de potencia

3.8.1. Modelamiento de cargas estáticas

Acorde con los datos del sistema a modelar, se debe ingresar los siguientes comandos:

- bus1 = Nodo en que se conecta la carga. Esta sólo se debe declarar en un nodo y no en dos nodos, dado que las cargas no se colocan en serie.
- Phases = Número de fases de la carga.
- Kv = Tensión nominal de la carga. Si la conexión es en Y se coloca tensión fase-tierra “wye”, pero si la conexión es fase-fase se pone “Delta”. Por defecto tiene conexión en Y.
- kW = Potencia activa total de la carga.

- Pf = Factor de potencia de la carga.
- Model = Este comando define como varía la carga respecto al voltaje. Por defecto muestra potencia activa y reactiva es constante se le asigna el valor de 1, el cual sirve para los estudios de flujo de carga. Los valores pueden variar desde el 1 hasta el 8, pero los otros valores son para otro tipo de estudios.
- Conn = Conexión de la carga en el nodo. Puede estar en delta o estrella. El comando para delta es "delta" y para estrella es "wye". Si no se declara dicho comando, OpenDSS por defecto lo toma en estrella.
- class = Valor entero que representa la carga. Por defecto se deja en 1.

3.8.2. Modelamiento de cargas variables

Para modelar cargas variables en el tiempo se usa el comando "loadshape". Generalmente las cargas son fijas pero también pueden ser variables, por lo tanto se debe especificar el tiempo en horas. Los comandos necesarios son:

- new loadshape = Comando para indicar una una carga variable.
- Npts = Número de puntos con los cuales se define la curva.
- Interval = Corresponde al intervalo de tiempo de los datos. Su valor predeterminado es 1.
- Mult = Comando que corresponde a un arreglo de datos. Estos valores dependerán del número de puntos establecidos en el comando Npts y su variación oscila entre 0.0 y 1.0.

3.9. Modelamiento del generador eólico

OpenDSS permite ingresar fuentes de generación eólica, para diversos estudios. Como datos mínimos se deben ingresar los siguientes comandos:

- New generator. = Comando requerido para mencionar un nuevo generador eólico en el sistema.
- bus1 = Nodo en el que está conectado el generador eólico.
- kV = Tensión nominal del generador eólico.
- kw = Potencia activa que genera el generador.
- pf = Factor de potencia del generador.
- conn = Tipo de conexión, ya sea en delta o estrella. Para el caso delta se escribe "delta" y para el caso estrella se escribe "wye".
- duty = Por defecto se debe colocar el comando wind, para que el software lo reconozca como un generador eólico.
- Model = Valor que corresponde a la potencia (activa y reactiva) vista desde el monitor.

3.10. Modelamiento del generador fotovoltaico

Para modelar un generador fotovoltaico se deben ingresar los siguientes comandos:

- New PVSystem. = Comando para establecer un nuevo generador fotovoltaico.
- phases = Número de fases que maneja en panel en la red. Por lo general el número es 3.
- bus1 = Nodo de conexión donde está conectado en panel.
- kV = Tensión nominal del panel conectado a la red.
- kVA = Potencia aparente que entrega el panel a la red.
- irrad = Valor de irradiancia en kW/m^2.
- Pmpp = Potencia máxima que puede se puede generar.
- temperature = Temperatura en la que esta operando el sistema fotovoltaico.
- PF = Factor de potencia del sistema fotovoltaico.

3.11. Monitor

El monitor es un elemento que registra los valores de tensión, corriente y potencia, en el dominio del tiempo para todas las fases en tiempo real. Estos valores pueden ser de forma diaria, anual o para los armónicos. Estos valores pueden ser guardados en un archivo .CSV. Los comandos necesarios son:

- New monitor. = Comando establecido para generar un nuevo monitor.
- Element = Es el elemento al cual se va a realizar el registro de las variables eléctricas.
- Terminal = Número del terminal al cual se conectará el monitor. Si no se indica el comando, se tomará automáticamente para las tres fases.
- Mode = Especificación para los datos que debe guardar el monitor mediante un código específico. Los códigos son:

  1. 0 = Tensión y corriente por cada fase de forma compleja.
  2. 1 = Potencia por cada fase en forma compleja (potencia activa y reactiva).
  3. 2 = Taps del transformador y se conecta en uno de los devanados.
  4. 3 = Variables de estado.
  5. 4 = Nivel de flicker e indice de severidad. Además nivel de parpadeo en un periodo de tiempo.
  6. 5 = Variables de solución como por ejemplo iteraciones.
  7. 6 = Conmutación de los condensadores.
  8. 16 = Tensiones y corrientes en componentes de secuencia.
  9. 32 = Muestra unas variables pero solo en magnitud.

3.12. Modelamiento de fallas de baja impedancia

OpenDSS permite simular fallas simétricas y asimetricas en los nodos, dentro de las cuales se encontran:

3.12.1. Falla monofásica

Para calcular la corriente de falla monofásica en componentes de fase, OpenDSS usa la Ecuación 2.

$$I_a = \frac{3V_1}{2Z_1 + Z_0} \tag{2}$$

Para calcular en componentes de secuencia se utiliza la Ecuación 3. El valor que toma E es de 1, dado que es un valor por unidad y está establecido por la ANSI/IEEE.

$$I_0 = I_1 = I_2 = \frac{E}{Z_1 + Z_2 + Z_0 + 3Z_F} \tag{3}$$

Obtenida la corriente de falla en componentes de secuencia, se puede pasar a componentes de fase, usando la Ecuación 4.

$$\left[I_{abc}\right] = \left[A\right]\left[I_{012}\right] \tag{4}$$

Expandiendo de forma general los componentes de la matriz A y de las corrientes en componentes de fase y secuencia se obtiene la Ecuación 5.

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} 1\angle 0° & 1\angle 0° & 1\angle 0° \\ 1\angle 0° & 1\angle -120° & 1\angle 120° \\ 1\angle 0° & 1\angle 120° & 1\angle -120° \end{bmatrix} \begin{bmatrix} I_0 \\ I_1 \\ I_2 \end{bmatrix} \tag{5}$$

OpenDSS cuando calcula la corriente de falla monofásica, lo muestra en valores reales, pero si en los cálculos teóricos se tienen en por unidad, se deben multiplicar por el valor base correspondiente.

Para que OpenDSS calcule la falla monofásica se debe declarar en una misma código el comando *"New Fault.F1"* seguido del comando *"phases=1"* y por último el nodo en el cual se presenta la falla, por ejemplo *"bus1=B.1"*. Si la falla no es en la fase 1, si no en la 2 o 3, simplemente donde esta el número 1 se cambia por la fase en que debe ocurrir la falla. Si hay un valor de resistencia de falla se agrega en la misma línea la letra *"r"* y se indica un valor especifico. Si la resistencia es de 1[Ω] se escribe *"r=1"*.

Posteriormente se debe indicar un valor base, para que el software calcule en valores reales y no solo en valores en por unidad. Por último se coloca los comandos *"calcv"* y *"solve"* para que el software realice los cálculos de las corrientes de falla del sistema.

Algoritmo para calcular la falla monofásica

```
New Fault.F1 phases=1 bus1=B.1 r=0
set voltagebases=[230]
calcv
solve
```

3.12.2. Falla bifásica

Para calcular la falla bifásica, si por ejemplo se presenta en las fases B y C, OpenDSS indicará que la corriente en componentes de secuencia para la secuencia cero será cero. Para la secuencia positiva tomará el valor de 1 sobre la suma de la impedancia en por unidad de la secuencia positiva, negativa y la impedancia de falla. Por último la secuencia negativa será igual a la secuencia positiva multiplicada por -1 ó $1\angle 180°$. Representado esto se puede ver en la ecuación 6.

$$\left[I_{012}\right] = \begin{bmatrix} 0 \\ \frac{E}{Z_1+Z_2+Z_F} \\ -I_1 \end{bmatrix} \tag{6}$$

Obtenidas las corrientes en componentes de secuencia, se puede proceder a calcularlas en componentes de fase usando la ecuación 7.:

$$\left[I_{abc}\right] = \left[A\right]\left[I_{012}\right] \tag{7}$$

Ahora para que OpenDSS calcule una falla bifásica, primero se declara el comando *"New Fault.F2"*, seguido el comando phases, donde le indicaremos que va el número 2 ya que nuestra falla es línea a línea. Posteriormente se indica el bus donde va la falla y las fases donde ocurre la falla *"bus1=B.2.3"* . Luego se debe indicar el comando *"bus2=B.4.4"* . Por último se coloca la tensión base del sistema y los comandos *"calcv"* y *"solve"* para que realice los cálculos respectivos.

Algoritmo para calcular la falla bifásica

```
New Fault.F2 phases=2 bus1=B.4.4 r=0
set voltagebases=[230]
calcv
solve
```

3.12.3. Falla bifásica a tierra

La falla bifásica a tierra se calcula en componentes de secuencia con la Ecuación 8:

$$\begin{bmatrix} I_0 \\ I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} \frac{E+I_1 Z_1}{Z_0+3Z_F} \\ \frac{E}{\frac{Z_2+Z_2(Z_0+3Z_F)}{Z_2+Z_0+3Z_F}} \\ \frac{-E+I_1 Z_1}{Z_2} \end{bmatrix} \tag{8}$$

Posteriormente se pasa de componentes de secuencia a componentes de fase usando la Ecuación 9.

$$\left[I_{abc}\right] = \left[A\right]\left[I_{012}\right] \tag{9}$$

En OpenDSS, para calcular la falla bifásica a tierra se usa los siguientes comandos:

Algoritmo para calcular la falla bifásica a tierra-Primera opción

```
New Fault.F2T phases=2 bus1=B.2.3 bus2=B.0.0 r=0
set voltagebases=[230]
calcv
solve
```

Se declara el comando *"New Fault.F2T"* para ingresar una nueva falla, con el nombre de F2T, luego el número de fases de la falla que para nuestro caso es 2 con el comando *"phases"*. Posteriormente se declara el nodo de falla que será el bus B y en el comando *"bus1"* se indica que la falla se presenta en las fases 2 y 3. En el comando *"bus2"* se ingresa B.0.0 para indicar que es hay una conexión a tierra en ese nodo. Por último se agregan los comandos *"set Voltagebases"* con el valor de la tensión base, *"calcv"* y *"solve"* para solucionar y calcular la falla propuesta.

Otra forma de escribir el código y que funciona de la anterior manera es la siguiente en OpenDSS:

Algoritmo para calcular la falla bifásica a tierra-Segunda opción

```
New Fault.F2T_2 phases=1 bus1=B.2 bus2=B.0
New Fault.F2T_3 phases=1 bus1=B.3 bus2=B.0
set voltagebases=[230]
calcv
solve
```

3.12.4. Falla trifásica

Teóricamente para calcular la falla trifásica, se usa la Ecuación 10. En el caso de las corrientes en secuencia cero y negativa, estas serán cero.

$$I_1 = \frac{E}{Z_1} \tag{10}$$

De forma matricial se puede escribir como se ve en la Ecuación 11.

$$\left[I_{012}\right] = \begin{bmatrix} 0 \\ \frac{E}{Z_1} \\ 0 \end{bmatrix} \tag{11}$$

Con la Ecuación 12 se calcula de componentes de secuencia a componentes de fases.

$$\left[I_{abc}\right] = \left[A\right]\left[I_{012}\right] \tag{12}$$

Para calcular una falla trifásica en OpenDSS se debe ingresar el comando *"New Fault.F3"*, seguido del comando *"New Fault.F3"* phases y se iguala a 3 y por último es esa misma línea se escribe *"bus1=B"*, donde B puede ser el nodo de falla o también puede ser algún otro nodo. Como no hay impedancia de falla, se escribe el comando *"r=0"*. De igual forma se declara la tensión base con el comando *"set Voltagebases"*, el comando *"calv"* y *"solve"*, para calcular las corrientes que hay en la falla propuesta.

Algoritmo para calcular la falla trifásica

```
New Fault.F3 phases=3 bus1=B r=0
set voltagebases=[230]
calcv
solve
```

3.13. Modo de ejecutar el código

Teniendo el código del sistema propuesto y se quiere ejecutar para ver los resultados, OpenDSS permite ejecutar el código de dos maneras. La primera opción es seleccionar el código y posteriormente dar click derecho y seleccionar la opción que dice *"Do Selected"*, tal como se ve en la Figura 9.

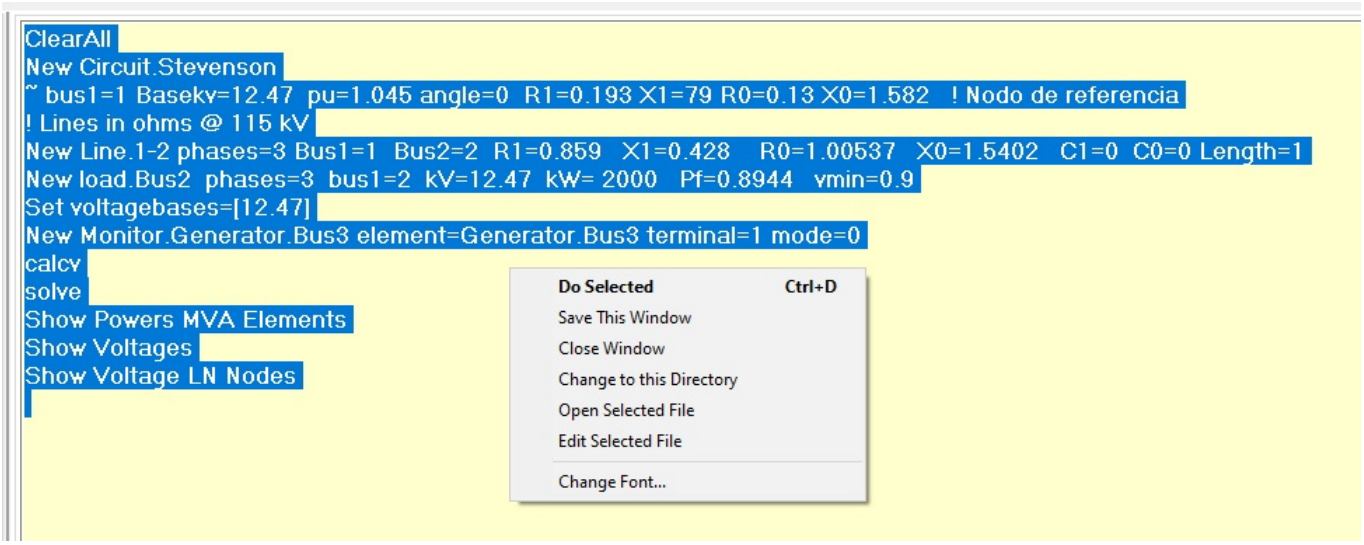


Figura 9: Modo de ejecución del código

La segunda opción es seleccionar todo el código y en el teclado oprimir simultáneamente las teclas (Control+D). Se debe tener presente que para el estudio que se este realizando, se debe agregar el comando *"solve"* para que resuelva el problema.

3.14. Modo de exportar

OpenDSS permite exportar datos en archivos con extensión .CSV o en .TXT. En caso de que se requiera archivos .CSV, los cuales permiten trabajar en Excel, en la ventana superior donde dice *"Export"*, se selecciona y se indica el tipo de datos que se van a exportar, como se puede observar en la Figura 10.

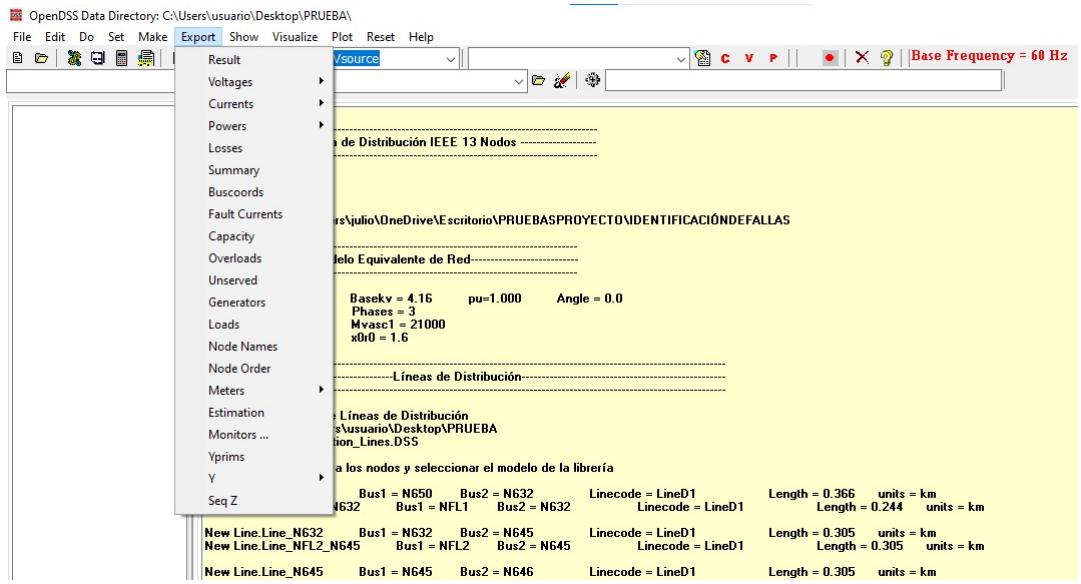


Figura 10: Ventana de exportación en OpenDSS

Luego en la parte derecha hay un cuadro llamado *"Compiled File Listed"*, donde se genera una ruta, la cual es donde se guarda el archivo exportado. Se selecciona esa ruta, se copia y se pega en una ventana del explorador de archivos, tal como se puede observar en la Figura 12.

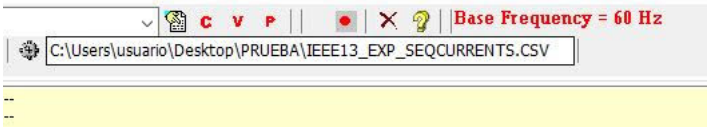


Figura 11: Ruta de generación de archivos .CSV en OpenDSS

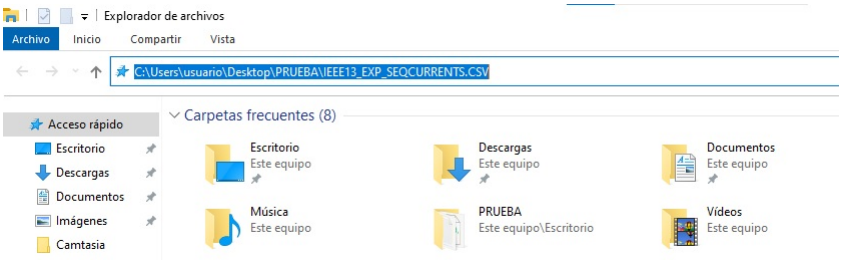


Figura 12: Explorador de archivos.

Al pegar la ruta de generación del archivo .CSV, se abrirá un archivo de Excel con los datos de exportación que se han indicado, como se puede ver en la Figura 13.

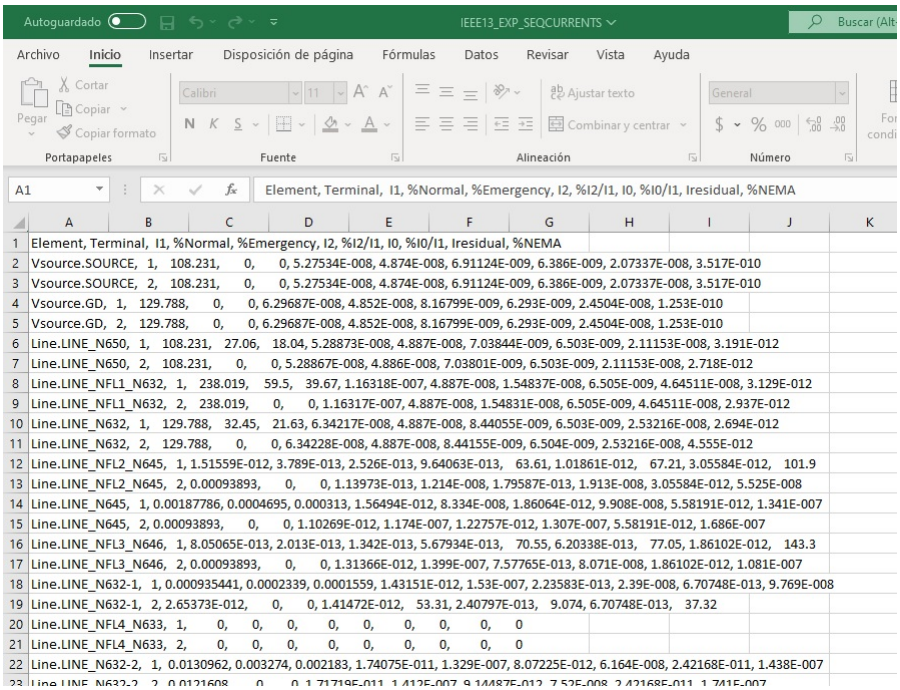


Figura 13: Archivo .CSV generado por OpenDSS

3.15. Mostrar resultados

Para mostrar los resultados, existen comandos los cuales permiten conocer valores necesarios después de haber corrido el flujo de carga ó un caso de estudio de fallas. Todos los resultados son exportados en ventanas de block de notas. Estos comandos se deben escribir al final del código, después de haber declarado todos los elementos del caso de estudio propuesto. Algunos de los comando comunes son:

- Show Voltages: Muestra las tensiones en componentes de fase para cada nodo, el valor en por unidad y la relación de la tensión en secuencia cero y negativa sobre la tensión en secuencia positiva.
- Show Voltage LN Nodes: Muestra las tensiones de línea, fase, el ángulo, el valor en por unidad, la base en cada nodo.

- Show Currents: Este comando muestra las corrientes en componentes de fase entre líneas, por cada uno de elementos, una relación porcentual entre la secuencia negativa y cero con la secuencia positiva y un valor porcentual de corriente entre un valor normal y de emergencia.
- Show mismatch Corresponde a la corriente en amperios en cada nodo y su porcentaje de máxima corriente.
- Show Power MVA: Muestra los datos del flujo de carga como son la potencia activa, reactiva, aparente, factor de potencia por cada nodo y las pérdidas totales del sistema.
- Show Power MVA elements: Muestra la potencia activa y reactiva en cada elemento en de componentes de fase. También muestra las pérdidas totales del sistema.
- Show Lineconstants: Muestra la matriz de impedancias, separando la resistencia, reactancia, capacitancia, susceptancia, inductancia y capacitancia. Se debe definir la frecuencia y las unidades en que se requiere que muestre los datos.
- Visualize: Es un comando que muestra un panel para mostrar tensiones y corrientes en componentes de fase. Por ejemplo es útil cuando se calculan valores de falla, como por ejemplo corrientes. Si se quiere mostrar las corrientes de falla usamos el comando Visualize Currents Fault.F3.

También se puede ver algunos resultados sin la necesidad de declarar el comando específico. Para esto en la parte superior se busca la opción que dice *"Show"* y se selecciona, como se puede ver en la Figura 14. Posteriormente se debe desplegar una ventana con una serie de opciones y sólo es seleccionar la que se necesite para el caso de estudio.

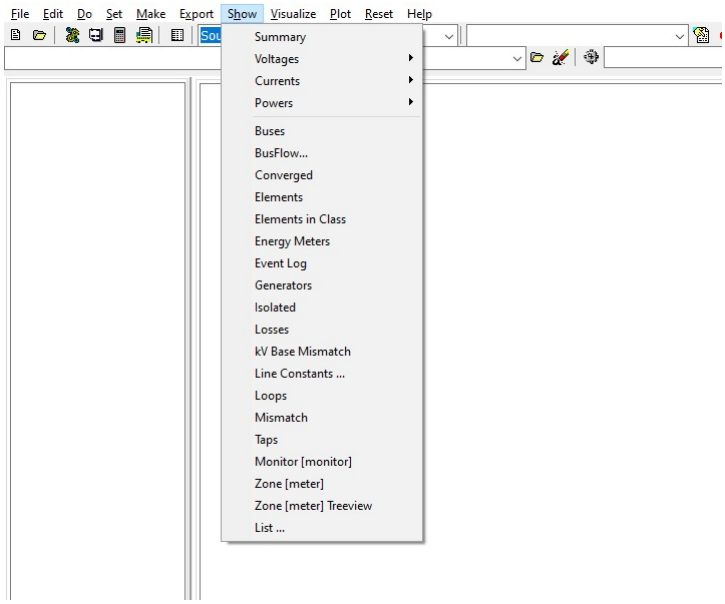


Figura 14: Ventana para mostrar datos en OpenDSS

3.16. Comentar texto

Para que OpenDSS no compile textos, indicaciones o títulos se puede usar el símbolo "!" y "//".

Algoritmo para el generador

```
ClearAll
! Creando un nuevo sistema
// Generador
```

4. Ejemplos

Los siguientes ejemplos muestran algunas formas en que se puede escribir el código para simular algunos sistemas.

4.1. Ejemplo 1: Circuito básico de 3 nodos.

En la Figura 15 se tiene un sistema, tomado del documento "OpenDSS Introductory Training Level 1"[2]. Este sistema está compuesto por un generador, un transformador trifásico con conexión delta-estrella, una línea de transmisión aérea y una carga estática.

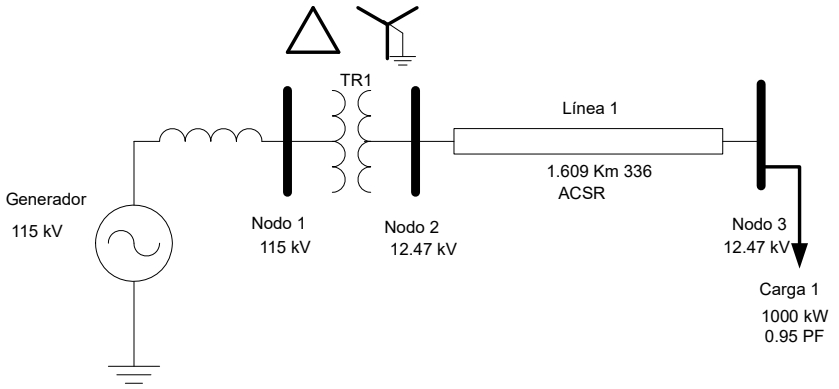


Figura 15: Circuito Básico de 3 nodos

4.1.1. Modelamiento del generador

Inicialmente se declara el comando *"ClearAll"* para borrar los datos de circuitos previamente compilados. Posteriormente se declara un nuevo sistema usando el comando *"New Circuit"*, el cuál se le da el nombre de *"Simple"*. En la siguiente línea se procede a ingresar los datos del generador, donde se declara el comando *"Edit Vsource"*, usado para definir un nuevo generador, con el comando *"BasekV"* se define la tensión del generador, con el comando *"pu"* se establece el valor en por unidad y con los comandos *"ISC3"* e *"ISC1"* se define las corrientes de cortocircuito trifásica y monofásica.

Algoritmo para el generador

```
ClearAll
New Circuit.Simple ! Creando la fuente
Edit Vsource.Nodo1 BasekV=115 pu=1.05 ISC3=3000 ISC1=2500 !Datos del generador
```

4.1.2. Modelamiento del transformador

Para ingresar el transformador, se inicia con el comando *"New Transformer"* y tendrá el nombre de TR1. Con el comando *"Buses"*, se define los nodos en los cuales va a estar conectado el transformador, en el nodo primario tiene el nombre de SourceBus y el nodo secundario tiene el nombre de Sub_Bus. La conexión se define con el comando *"Conns"*, para el primario se establece en Delta y para el secundario en estrella.

Para la tensión del transformador se usa el comando *"kVs"* y entre llaves se define la tensión primaria en 115 y la tensión secundaria en 12.47, ambas en kV. La potencia aparente se define con el comando *"kVAs"* y entre llaves se establece un valor de 20000 en kVAs. Por último con el comando *"XHL"*, se define la impedancia de cortocircuito con un valor de 10. El símbolo ~ sirve para trabajar en otra línea de código y se obtiene con las teclas ALT+126.

Algoritmo para el transformador

```
New Transformer.TR1 Buses=[Nodo1, Nodo2] Conns=[Delta Wye] kVs= [115 12.47]
~ kVAs=[20000 20000] XHL=10
```

4.1.3. Modelamiento de la línea

Para definir los datos de la línea de transmisión, primero se define el comando *"New Linecode."* y un nombre de 336ACSR. Luego se define en secuencia cero y positiva la resistencia con el comando *"R0"* y *"R1"*, reactancia con el comando *"X0"* y *"X1"* y capacitancia con los comandos *"C0"* y *"C1"* y por último las unidades con el comando *"Units"*, que están en Km.

En la siguiente línea de código, se define la conexión de la línea de transmisión. Se escribe el comando *"New Line."* con el nombre Línea1. Después se define el nodo desde donde inicia con el comando *"Bus1"* y definiendo el nombre de Sub_Bus para el nodo donde termina se usa el comando *"Bus2"* con el nombre de LoadBus. El comando *"Linecode"* permite traer los valores de impedancia llamados en la anterior línea de código con el nombre de 336ACSR. El comando *"Length"* se utiliza para establecer la longitud de la línea y por último se define las unidades de la longitud con el comando *"Units"*.

Algoritmo para la línea de transmisión

```
New Linecode.336ACSR R1=0.058 X1=0.1206 R0=0.1784 X0=0.4047 C1=3.4 C0=1.6 Units=Km
New Line.Linea1 Bus1=Nodo2 Bus2=Nodo3 Linecode=336ACSR Length=1.609 Units=Km
```

4.1.4. Modelamiento de la carga

Para la carga, inicialmente se escribe el comando *"New Load."* con el nombre de Carga. Con el comando *"Bus1"* se indica el nodo en que está conectada la carga, con el comando *"kV"* se establece la tensión en kV, luego con el comando *"kW"* se define la potencia activa en KW y por último el factor de potencia con el comando *"PF"*.

Algoritmo para la carga

```
New Load.Carga Bus1=Nodo3 kV=12.47 kW=1000 PF=0.95
```

4.1.5. Mostrar resultados

Al haber definido todos los parámetros, se procede a solucionar el sistema con el comando *"Solve"*. Posteriormente para mostrar los resultados, en el caso de las tensiones se usa el comando *"Show Voltages"*, para las corrientes *"Show Currents"* y para la potencia *"Show Power kVA elememts"*.

Algoritmo para mostrar resultados

```
Solve
Show Voltages
Show Currents
Show Powers kVA elements
```

4.1.6. Ejecución del código

Para compilar se selecciona el código y el usuario deberá oprimir las teclas control+D. Si el código está bien escrito tendrá que mostrar que el sistema fue solucionado, tal como se ve en la Figura 16. Si el software compilo, se deberán desplegar unas ventanas en block de notas con los datos que se indicaron anteriormente como son las tensiones de línea, las corrientes en componentes de fase y secuencia y el flujo de carga para cada elemento del sistema, de tal como se ve en la Figura 17.

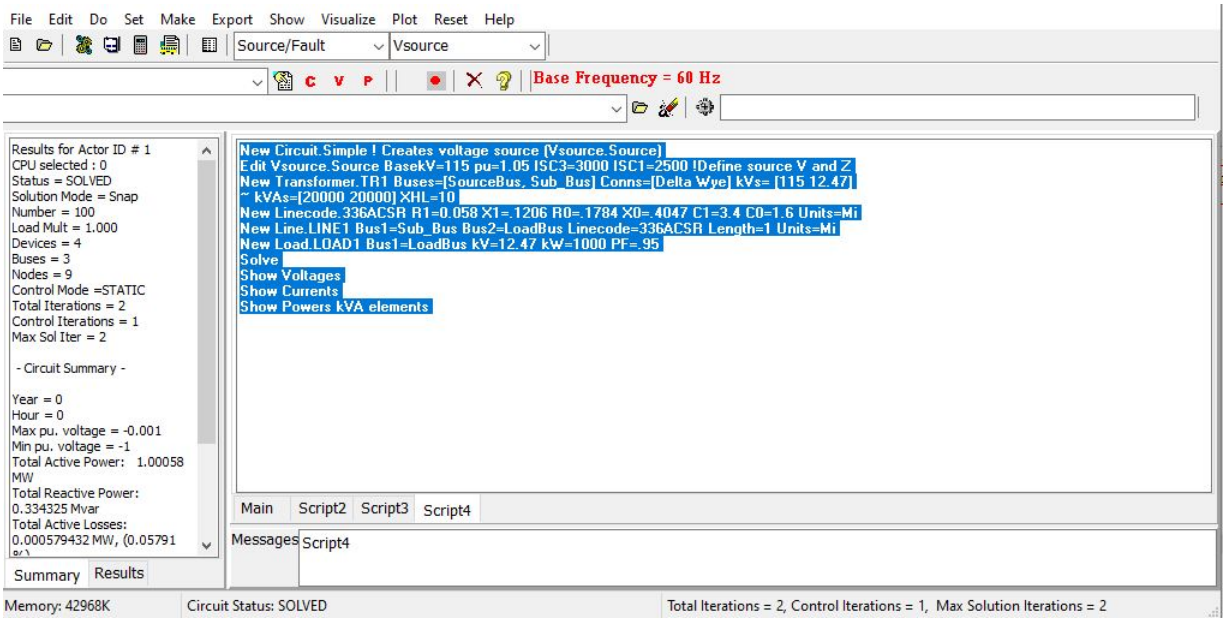


Figura 16: Modo de ejecución del ejemplo 1

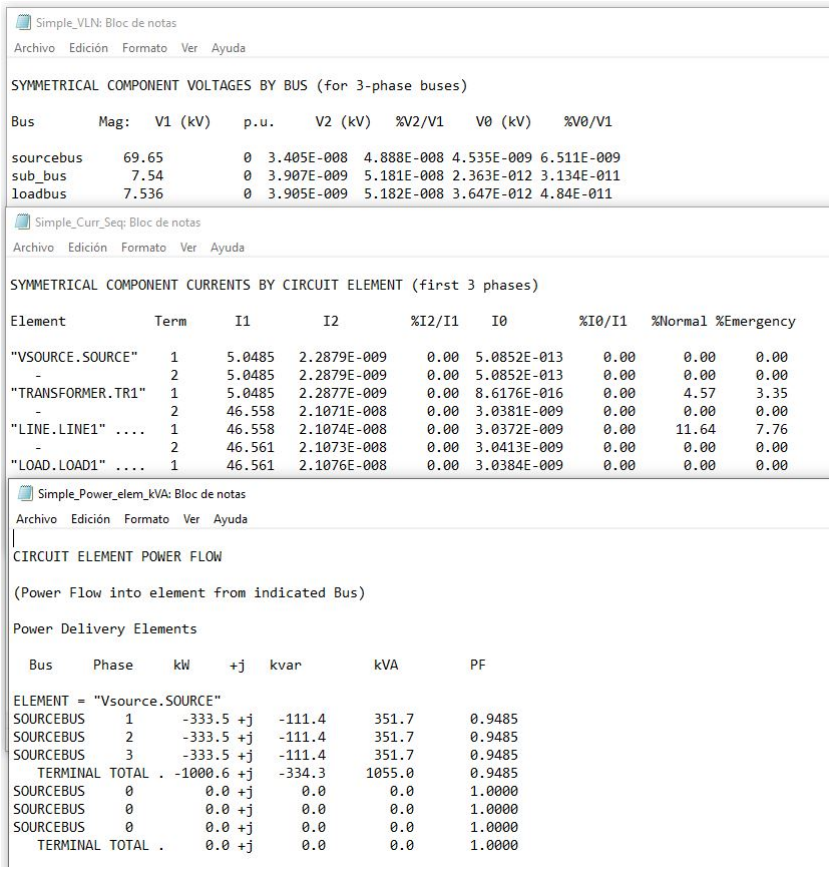


Figura 17: Resultados del Ejemplo 1

4.2. Ejemplo 2: Generador Slack y PV.

Se tiene el siguiente modelo tomado del libro "Análisis de Sistemas de Potencia de Stevenson 2˚ Edición"[3]. Consta de 5 nodos, un generados slack, un generador de potencia controlada y tres cargas estáticas, como se puede ver en la Figura 18.

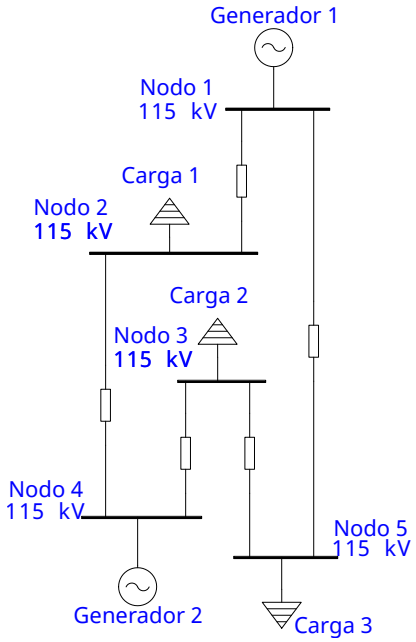


Figura 18: Sistema Eléctrico del ejemplo 3

4.2.1. Modelamiento del generador 1 ó nodo de referencia

Inicialmente se declara el comando *"ClearAll"* para borrar los datos de circuitos previamente compilados. Posteriormente declara un nuevo sistema usando el comando *"New Circuit"*, el cuál se le da el nombre de Stevenson. En la siguiente línea se procede a ingresar los datos del generador, iniciando con el comando *"bus1"* para indicar en donde está conectado el generador, con el comando *"BasekV"* se define la tensión del generador, con el comando *"pu"* se establece el valor en por unidad y con los comandos *"mvasc3"* y *"mvasc1"* se define las potencias de cortocircuito trifásica y monofásica.

Algoritmo para el generador

```
ClearAll
New Circuit.Stevenson
~ bus1=1 Basekv=115 pu=1.02 mvasc3=1000000 mvasc1=1000000 ! Nodo de referencia
```

4.2.2. Modelamiento de las líneas de transmisión

La siguiente parte del código corresponde a la declaración de cada línea del sistema. Inicialmente se declara el comando *"New Line"* y se asigna un nombre, además se define el nodo desde donde inicia con el comando *"Bus1"* y el nodo donde termina con el comando *"Bus2"*. Posteriormente se define en secuencia cero y positiva la resistencia con los comando *"R0"* y *"R1"*, la reactancia con los comandos *"X0"* y *"X1"* y la capacitancia con los comandos *"C0"* y *"C1"*. Por último se define la longitud con el comando *"Length"* y las unidades de medida con el comando *"Units"*. Estos códigos se deben realizar con cada una de las líneas que componen el sistema.

Algoritmo para las líneas de transmisión

```
New Line.1-2 phases=3 Bus1=1 Bus2=2 R1=13.225 X1=52.9 R0=13.225 X0=52.9 C1=0 C0=0 Length=1 Units=Km
New Line.1-4 phases=3 Bus1=1 Bus2=4 R1=19.8375 X1=79.35 R0=19.8375 X0=79.35 C1=0 C0=0 Length=1 Units=Km
New Line.1-5 phases=3 Bus1=1 Bus2=5 R1=6.6125 X1=26.45 R0=6.6125 X0=26.45 C1=0 C0=0 Length=1 Units=Km
New Line.2-3 phases=3 Bus1=2 Bus2=3 R1=6.6125 X1=26.45 R0=6.6125 X0=26.45 C1=0 C0=0 Length=1 Units=Km
New Line.2-4 phases=3 Bus1=2 Bus2=4 R1=13.225 X1=52.9 R0=13.225 X0=52.9 C1=0 C0=0 Length=1 Units=Km
New Line.3-5 phases=3 Bus1=3 Bus2=5 R1=6.6125 X1=26.45 R0=6.6125 X0=26.45 C1=0 C0=0 Length=1 Units=Km
```

4.2.3. Modelamiento del generador de potencia controlada

Para ingresar el generador de potencia controlada, inicialmente se escribe el comando *"New Generator."* y se especifica con el nombre de Bus3. Se sigue con el comando *"phases"* que es utlizado para indicar que el generador es trifásico, el comando *"kV"* indica la tensión en kV, el comando *"kW"* corresponde la potencia activa en kilovatios, el comando *"model"* es asignado para establecer un generador P-V.

El comando *"Vpu"* establece la tensión en por unidad, los comandos *"maxkvar"* y *"minkvar"* permiten definir un rango de potencia reactiva en kVar y por último el comando *"PVFactor"* corresponde a un factor de desaceleración para las generadores P-V.

Algoritmo para el generador de potencia controlada

```
New Generator.Bus3 phases=3 bus1=3 kV=115 kW=100000 model=3 Vpu=1.04 maxkvar=60000 minkvar=-45000
~ PVFactor=0.5
```

4.2.4. Modelamiento de las cargas

Para la carga, inicialmente se escribe el comando *"New Load."* con el nombre de Bus2. El siguiente comando es *"phases"* usado para indicar que es una carga trifásica, el comando *"Bus1"* se indica el nodo en que está conectada la carga, con el comando *"kV"* se establece la tensión en kV, luego con el comando *"kW"* se define la potencia activa en KW.

En la primera carga se define el factor de potencia con el comando *"PF"* y en las otras dos cargas se define la potencia reactiva en kVAr con el comando *"kvar"*. Por último se define la tensión mínima en por unidad con el comando *"vmin"*.

Algoritmo para las cargas

```
New load.Bus2 phases=3 bus1=2 kV=115 kW= 60000 Pf=0.89 vmin=0.9
New load.Bus4 phases=3 bus1=4 kV=115 kW= 40000 kvar= 10000 vmin=0.9
New load.Bus5 phases=3 bus1=5 kV=115 kW= 60000 kvar= 20000 vmin=0.9
```

Al definir todos los elementos que conforman el sistema, se define el valor de la tensión base del sistema con el comando *"Set voltagebases"*, el comando *"calv"* estima el valor base para cada nodo y por último el comando *"solve"* que sirve para solucionar el sistema.

Algoritmo para compilar el sistema

```
Set voltagebases=[115]
calcv
solve
```

Al haber solucionado el sistema, se puede proceder a ver los valores de tensión, corriente y potencia, seleccionando en la parte superior la opción *"Show"* y se selecciona el valor requerido.

4.3. Ejemplo 3: Generación distribuida.

Se tiene el siguiente sistema, el cual es tomado del manual de OpenDSS[4]. Se tiene generador, dos transformadores, tres líneas de transmisión, cargas, capacitores y dos monitores como se puede observar en la Figura 19.

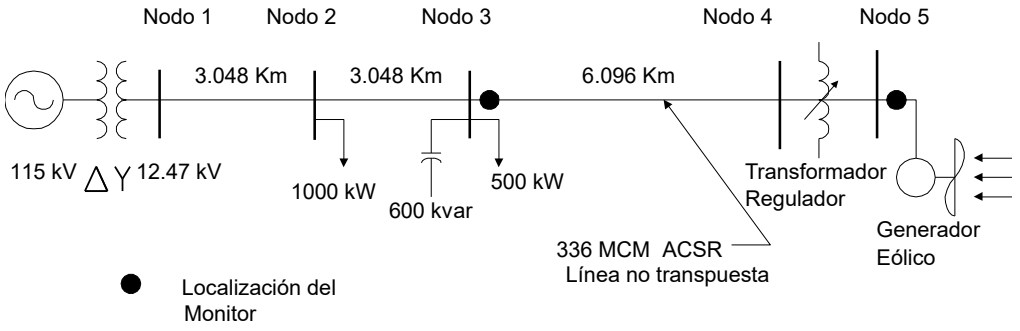


Figura 19: Circuito aplicando generación distribuida

4.3.1. Generador

Inicialmente se declara un nuevo circuito con el comando *"New Circuit."* y se asiganara el nombre de Simple. Posteriormente se procede a ingresar los datos del generador. Se iniciará con el comando *"Basekv"*, para indicar la tensión base del sistema. Luego se procede a ingresar unos valores numéricos sin asignar algún comando. OpenDSS permite ingresar valores sin tener presente el comando, pero se debe tener presente el orden para no presentar errores de calculo. El primer valor es 1.00 que corresponde a la tensión en por unidad, el valor 0.0 al ángulo en grados, el valor 60.0 a la frecuencia en Hz, el valor 3 corresponde a un sistema trifásico, los valores 20000 y 21000 a la potencia de cortocircuito trifásica en kVA y por último los valores 4.0 y 3.0 que son la relación de impedancia x_1/r_1 y x_0/r_0.

Algoritmo para el generador

```
New Circuit.Simple
basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0 ! Fuente de Tensión
```

4.3.2. Algoritmo para las curvas de carga

Para las simulación de carga con variación en el tiempo se usará el comando *"New loadshape"*, seguido del comando *"day"* para indicar que es simulación diaria. Seguido a esto irá el número 24, dado que es el número de puntos dela curva y el número 1.0 es el intervalo en horas. Posteriormente se escribe el comando *"mult"* para definir los puntos que van a estar entre 0 y 1.0. Además el comando *"day"* se puede cambiar de forma anual con el comando *"year"*.

Algoritmo para las curvas de generación

```
new loadshape.day 24 1.0
mult=(.3 .3 .3 .35 .36 .39 .41 .48 .52 .59 .62 .94 .87 .91 .95 .95 1.0 .98 .94 .92 .61 .60 .51 .44)
```

El segundo comando *"New loadshape"* corresponde a la carga del generador eólico. Se indica el comando *"wind"* para especificar que es de tipo viento, luego va el número 2400, que corresponde al número de puntos que contiene el vector y el número 0.00027777 corresponde a un factor para obtener los datos de una hora. Por último el comando *"action=normalize"* permite ajustar los multiplicadores para que su valor pico sea de 1.0.

Algoritmo para las curvas de generación

```
new loadshape.wind 2400 0.00027777 ! unit must be hours 1.0/3600.0 = .0002777
mult=(file=zavwind.csv) action=normalize !característica de la turbina eólica
```

Para obtener las características de la turbina, se debe ingresar a la siguiente página `https://raw.githubusercontent.com/tshort/OpenDSS/master/Distrib/Examples/Manual/ZavWind.csv`, el cual contiene 2400 datos. Para guardar el archivo en formato .csv simplemente se da click derecho y se da guardar en la carpeta donde se está trabajando el presente sistema.

4.3.3. Algoritmo para la impedancia de las líneas de transmisión

Para la impedancia de la línea de transmisión se deben ingresar los siguientes comandos. El comando *"New linecode."* establece una asignación para la impedancia y seguido de este comando, va el comando *"nphases"* para indicar que es un sistema trifásico. El siguiente comando es *"rmatrix"* que corresponde para ingresar la matriz con los valores de resistencia en Ω por km. El comando *"xmatrix"* permite ingresar la matriz con los valores de la reactancia en Ω por km. El comando *"cmatrix"* permite ingresar la matriz con los valores de la capacitancia en nF por km. Para las anteriores matrices, solo se ingresa los elementos que están debajo de la diagonal, ya que OpenDSS completa los demás datos de la matriz.

Por último se ingresa la corriente nominal en amperios de los conductores con el comando *"Normamps"* y el comando *"Emergamps"* corresponde a la corriente máxima de emergencia de los conductores.

Algoritmo para ingresar la impedancia de las líneas de transmisión

```
! define a linecode for the lines - unbalanced 336 MCM ACSR connection
new linecode.336matrix nphases=3 ! horizontal flat construction
rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455)
! ohms per 1000 ft
xmatrix=(0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 0.2025449)
cmatrix=(2.74 | -0.70 2.96 | -0.34 -0.71 2.74) !nf per 10 ft
Normamps = 400 Emergamps=600
```

4.3.4. Conexión de las líneas de transmisión

Para la conexión de las líneas, primero se debe agregar el comando *"New line."* y se asigna un nombre por línea. Posteriormente se agrega el comando *"Bus1"* para indicar el nodo desde donde inicia la línea y el comando *"Bus2"* para indicar el nodo donde termina. El comando *"linecode"* trae la matriz de impedancia y capacitancia descrita en el algoritmo para ingresar la impedancia. El comando *"length"* corresponde a la distancia de la línea entre cada nodo y por último el comando *"units"* son las unidades en que está la longitud de la línea. Estos mismos pasos se deben realizar para cada línea que compone el sistema.

Algoritmo para la conexión de las líneas de transmisión

```
! Definiendo la conexión de las líneas
New line.line1 Bus1=N1 Bus2=N2 linecode=336matrix length=3.048 units=km
New line.line2 Bus1=N2 Bus2=N3 linecode=336matrix length=3.048 units=km
New line.line3 Bus1=N3 Bus2=N4 linecode=336matrix length=6.096 units=km
```

4.3.5. Transformador

Para modelar el transformador, se debe ingresar inicialmente el comando *"New Transformer."*. Posteriormente se declara el comando *"Phases"* para indicar que es un transformador trifásico con el número 3, seguido se escribe el comando *"windings"* para establecer un transformador de 2 devanados, el comando *"buses"* establece los nodos de conexión de transformador, el comando *"conns"* indicará la conexión de los devanados, el comando *"kvs"* establece la tensión en kV para los devanados, el comando *"kvas"* indica la potencia aparente para los devanados en kVA y por último el comando *"XHL"* corresponde a la potencia de cortocircuito en forma porcentual.

Algoritmo para el transformador

```
!Transformador
New transformer.sub phases=3 windings=2 buses=(Nodo1 Nodo2) conns='delta wye' kvs="115 12.47" kvas="20000 20000" XHL=7
```

4.3.6. Transformador regulador

Para el transformador regulador se deben declarar los mismos comandos descritos en el paso anterior. El comando *"New transformer."* para declarar un nuevo transformador con su respectivo nombre. El comando *"phases"* establece el número de fases, el comando *"windigs"* indica el número de devanados, el comando *"buses"* corresponde a los nodos donde está conectado el transformador, el comando *"kvs"* indica la tensión en kV que hay en el devanado primario y secundario, el comando *"kvas"* establece la potencia aparente en kVA y el comando *"XHL"* corresponde a la potencia de cortocircuito en forma porcentual.

Posteriormente se deben definir los comandos para la regulación de tensión en los transformadores. El primer comando es *"new regcontrol."* usado para llamar un transformador regulador. El siguiente comando es *"transformer"* que sirve para llamar el transformador anteriormente ingresado, el comando *"windings"* corresponde al número de devanados que se están monitoreando, el comando *"vreg"* indica la variación de tensión en voltios para los devanados, el comando *"band"* es el ancho de banda para el bus que está controlado, el comando *"ptratio"* corresponde a la relación entre el transformador de potencial que convierte la tensión del devanado controlado a la tensión de control del regulador, que por defecto su valor es 60 y por último el comando *"delay"* corresponde al tiempo de retardo en segundos.

Algoritmo para el transformador regulador

```
! Transformador regulador
new transformer.reg1 phases=3 windings=2
buses=(Nodo4 Nodo5)
conns='wye wye'
kvs="12.47 12.47"
kvas="8000 8000"
XHL=1
! Control del transformador regulador
new regcontrol.sub transformer=sub winding=2 vreg=125 band=3 ptratio=60
delay=10
new regcontrol.reg1 transformer=reg1 winding=2 vreg=122 band=3 ptratio=60
delay=15
```

4.3.7. Cargas

Inicialmente se debe declarar el comando *"New load."* para indicar una nueva carga con un nombre específico. El comando a seguir es *"bus1"*, en el cual se indica donde está conectada la carga. El comando *"phases"* establece que la carga es trifásica con el número 3, el comando *"kv"* corresponde a la tensión en kV, el comando *"kw"* es la potencia activa en kW, el comando *"pf"* es el factor de potencia y el comando *"model"* establece el flujo de potencia constante de P y Q.

Las cargas al tener variación en el tiempo se deben agregar algunos comandos para que OpenDSS realice los cálcuos de forma correcta. El primer comando es *"class"* es un comando que representa el tipo de carga y por defecto se deja en 1, el comando *"yearly"* permite hacer simulaciones anuales y el comando *"daily"* permite hacer simulaciones diarias. EL comando *"conn"* se usa para establecer la conexión, que para este caso está en delta y el comando *"status"* permite dejar fijas las cargas sin multiplicarlas por los factores de carga.

Algoritmo para las cargas

```
! Cargas

new load.load1 bus1=Nodo3 phases=3 kv=12.47 kw=1000.0 pf=0.88 model=1

class=1 yearly=year daily=day status=fixed

new load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1

class=1 yearly=year daily=day conn=delta status=fixed
```

4.3.8. Capacitor

Para el capacitor, inicialmente se declara el comando *"New Capacitor."* y se asigna el nombre de C1. Los siguientes comandos corresponden a *"bus1"*, el cual corresponde al nodo donde está conectado el capacitor, el comando *"phases"* al número de fases que tiene el capacitor, el comando *"kvar"* indica la potencia reactiva en kVAr y por último el comando *"kv"* a la tensión nominal en kV.

Algoritmo para el capacitor

```
! Capacitor

new capacitor.C1 bus1=Nodo3 phases=3 kvar=600 kv=12.47
```

4.3.9. Control del capacitor

Para el capacitor, inicialmente se declara el comando *"New Capacitor."* y se asigna el nombre de C1. Los siguientes comandos corresponden a *"bus1"*, el cual corresponde al nodo donde está conectado el capacitor, el comando *"phases"* al número de fases que tiene el capacitor, el comando *"kvar"* indica la potencia reactiva en kVAr y por último el comando *"kv"* a la tensión nominal en kV.

Algoritmo para el control del capacitor

```
! Control del Capacitor
new capcontrol.C1 element=line.line3 1 capacitor=C1
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

4.3.10. Generador eólico

El comando inicial para ingresar un generador eólico es *"New generator."* y se asgina el nombre de gen1. Luego el comando *"bus1"* corresponde al nodo donde está conectado el generador, el comando *"kV"* es la tensión de kV, el comando *"kW"* es la potencia de generación en kVA, el comando *"pf"* es el factor de potencia, el comando *"conn"* asigna la conexión que para este modelo está en delta, el comando *"duty"* establece la generación eólica y por último el comando *"Model"* para establecer un flujo de potencia constante de P y Q.

Algoritmo para el generador eólico

```
!Generador Eólico de 8MW
New generator.gen1 bus1=Nodo5 kV=12.47 kW=8000 pf=1 conn=delta duty=wind
Model=1
```

4.3.11. Monitor

Para generar un nuevo monitor se debe agregar el comando *"New Monitor."* con un respectivo nombre. El siguiente comando es *"element"* el cual se debe indicar el elemento al que se van a monitorear las variables y además se debe anexar el nombre propuesto al elemento. Por último el comando *"model"* asigna un valor numérico que registrará tensiones y corrientes.

Algoritmo para el monitor

```
! Monitor
New Monitor.gen1a element=generator.gen1 1 mode=48
New Monitor.line3 element=line.line3 1 mode=48
New Monitor.gen1 element=generator.gen1 1 mode=32
```

4.3.12. Ejecución del Código

Finalmente se agregan los comandos para ejecutar el código. El primer comando es *"Set voltagebases"* en el cual se debe declara todas las tensiones del sistema. Posteriormente el comando *"Calcv"* que permite ejecutar el código, el comando *"Set controlmode=time"* establece que el control dependerá del tiempo, el comando *"Set mode=duty"* establece un ciclo de trabajo, el comando *"number"* corresponde a un valor de 2400 segundos, el comando *"hour"* el número de horas, el comando *"h"* establece el número de pasos y el comando *"sec"* establece los segundos a partir de la hora de inicio de la solución del circuito.

Algoritmo para ejecutar el código

```
!Definiendo tensiones
Set voltagebases="115 12.47 .48"
Calcv
Set controlmode=time
Set mode=duty number=2400 hour=0 h=1.0 sec=0 !mode reinicia los monitores
```

4.4. Ejemplo 4: Línea no transpuesta

En la Figura 20 se tiene una línea de transmisión trifásica con circuito sencillo, en configuración horizontal, dos conductores de haz por fase y un cable de guarda. La S_{base} = 100MVA y V_{base}=115 kV y su longitud es de 1Km. La línea es no transpuesta y se obtendrá la matriz de impendancia. El presente ejemplo es tomado del material académico de la asignatura "Análisis de Fallas y Coordinación de protecciones".

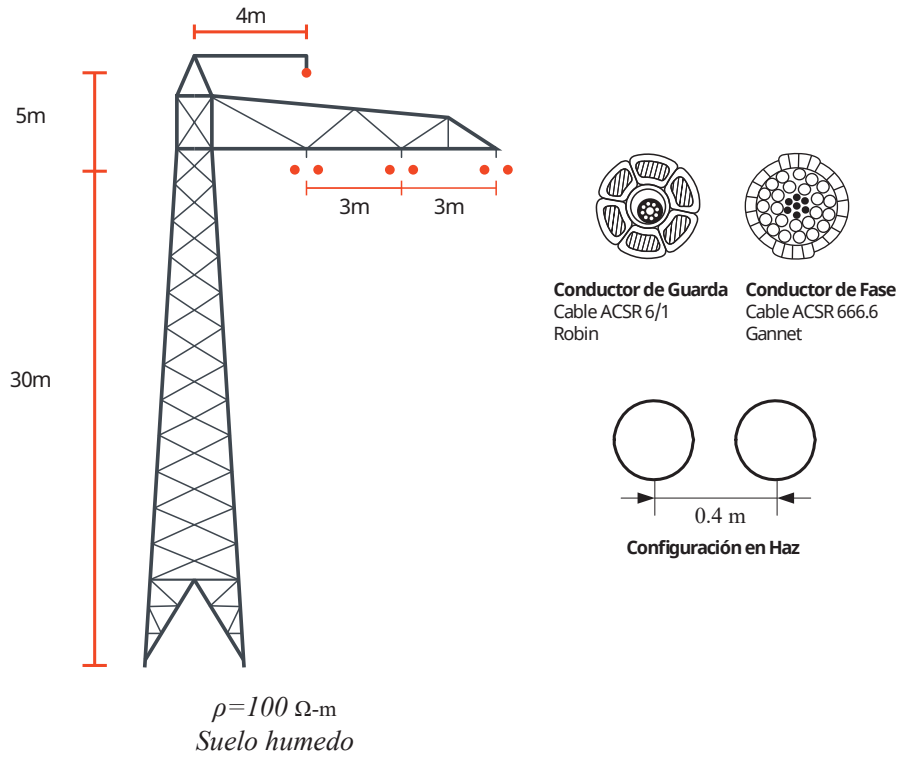


Figura 20: Configuración del circuito sencillo con haz de conductores

En la Tabla 1 se tienen los datos nominales de los conductores de fase y el conductor de guarda.

Cableado Aluminio/Acero	Código	Calibre AWG/Kcmil	Diametros (mm)					Peso Total Aprox (kg/km)			Carga de Rotura	Resistencia Eléctrica (Ohm/km)		Capacidad de Corriente	
			Hilos Individuales		Nucleo	Total	RMG							(2)	CC(3)
			Acero	Al				Al	Acero	Total	kg-f	DC a 20°C(1)	AC a 75°C	A	kA
Cableado 6/1	Robin	1	3,00	3,00	3,00	9,00	2,90	116,40	55,00	171	1612	0,658	0,884	211	6,4
Cableado 26/7	Gannet	666,6	3,16	4,07	9,49	25,75	10,46	936,5	429,4	1366	11979	0,0835	0,102	816	50,7

Cuadro 1: Tabla de conductores para líneas aéreas en Aluminio/Acero

4.4.1. Definiendo el nuevo circuito

Inicialmente se crea un nuevo circuito con el comando *"New Circuit."* y se asigna el nombre de LineaNoTranspuesta. Luego se específica la tensión base con el comando *"BasekV"* y por último se asigna el modelo de tierra de Carson con el comando *"set earthmodel"*.

Algoritmo para generar un nuevo circuito

```
ClearAll

New Circuit.LineaNoTranspuesta
~ BasekV = 115
set earthmodel=Carson
```

4.4.2. Características de los conductores

La siguiente sección del código corresponde a los datos nominales de los conductores de fase y guarda. Inicialmente se escribe el comando *"New Wiredata."* usado para definir los datos de la línea y se asigna el nombre del

conductor de fase GANNET. El siguiente comando es *"GMR"* que corresponde al radio medio geométrico, el comando *"DIAM"* es el diametro del conductor, el comando *"RAC"* es la resistencia eléctrica del conductor. Dentro del mismo comando *"RAC"* se escribe el número 2 y el símbolo /, que es para indicar que hay dos haz de conductores por fase.

Posteriormente el comando *"NormAmps"* corresponde a la corriente nominal del conductor en amperios. Dentro del paréntesis, además de la corriente nominal, se agrega el número 2 y el símbolo * para indicar que hay dos conductores por fase. Por último se definen las unidades de cada una de las variables descritas. En el caso de la resistencia se escribe el comando *"Runits"* y se indica que están en kilómetros, el comando *"radunits"* corresponde a las unidades del radio externo del conductor y se específica en metros y el comando *"gmrunits"* corresponde a las unidades en que está el radio medio geométrico, también en metros. Estos mismos pasos se deberán replicar con el conductor de guarda.

Algoritmo sobre las características de los conductores

```
// Definición de conductores de fase y de guarda
// Conductor de Fase
New Wiredata.GANNET GMR=0.01046 DIAM=0.02575 RAC=(0.102 2 /) NormAmps=(816 2 *)
~ Runits=km radunits=m gmrunits=m
// Conductor de Guarda
New Wiredata.ROBIN GMR=0.00290 DIAM=0.009 RAC=0.884 NormAmps=184.0
~ Runits=km radunits=m gmrunits=m
```

4.4.3. Posición de los conductores de fase y guarda

En el caso de la posición de los conductores, se deben hacer para cada una de las fases y el conductor de guarda. Se inicia con el comando *"New LineGeometry."* es usado para declarar la geometría de las líneas. El siguiente comando es *"Nconds"* que corresponde al número de conductores incluyendo el de guarda. Posteriormente se indica el número de fases con el comando *"Nphases"* igual a 3 y después se indica el comando *"reduce"* para establecer si se realiza la reducción de Kron.

Se da un salto de línea y se procede a indicar la posición de los conductores de fase. El comando *"cond"* se refiere al conductor de fase o al conductor de guarda, el comando *"wire"* corresponde al conductor si de es fase o guarda, el comando *"x"* es la posición en el eje horizontal, *"h"* es la altura y por último el comando *"units"* define las unidades. Se debe definir estos datos para cada haz de conductores de fase y para el conductor de guarda. No se define la posición exacta de los conductores de la fase 1, ya que anteriormente se estableció que el circuito era de dos haz de conductores por fase.

Algoritmo de posición de los conductores

```
// Conductores de fase
New LineGeometry.Linea Nconds=4 Nphases=3 reduce=no
~ cond=1 wire =GANNET x=4 h=30 units=m
~ cond=2 wire =GANNET x= 7 h=30 units=m
~ cond=3 wire =GANNET x= 10 h=30 units=m
// Conductor de guarda
~ cond=4 wire =ROBIN x=4 h=35 units=m
```

4.4.4. Matriz de impedancia

Para mostrar la matriz de impedancia se deben ingresar los siguientes comandos. El primer comando es *"New Line."* para establecer que se realizará la matriz equivalente. El comando *"Bus1"* corresponde al nodo inicial con el nombre de Businicial.1.2.3 y el comando *"Bus2"* corresponde al nodo final con el nombre de Busfinal.1.2.3. Los números 1.2.3 específica la transposición de la línea, pero como no hay transposición iniciará en un valor y finalizará en el mismo. Luego el comando *"Geometry"* trae los datos de los conductores de fase y guarda, el comando *"Length"* corresponde a la longitud de la línea y el comando *"units"* indica la unidades.

Para mostrar la matriz de impedancias se debe escribir el comando *"Show Lineconstants"*, además se debe

definir la frecuencia con el comando *"freq"* dado que al no hacerlo, el realizará los cálculos con una frecuencia de 1 Hz y por último para mostrar los resultados en Ω/km se usa el comando *"units"*.

Algoritmo para mostrar la matriz de impedancia

```
New Line.Equivalente Bus1=Businicial.1.2.3 Bus2=Busfinal.1.2.3 Geometry=Linea Length=1 units=km
Show Lineconstants freq=60 units=km
```

4.4.5. Resultados

Al compilar el código, se mostrarán dos ventanas de block de notas de tal forma como se puede ver en la Figura 21. Un block de notas muestra los valores de la resistencia y reactancia en [Ω/km] y la capacitancia de la línea en [nF/km] y el otro block de notas mostrará estos mismos datos, pero incluyendo la susceptancia en siemens, la inductancia en henrios y la capacitancia en nanofaradios por kilómetro.

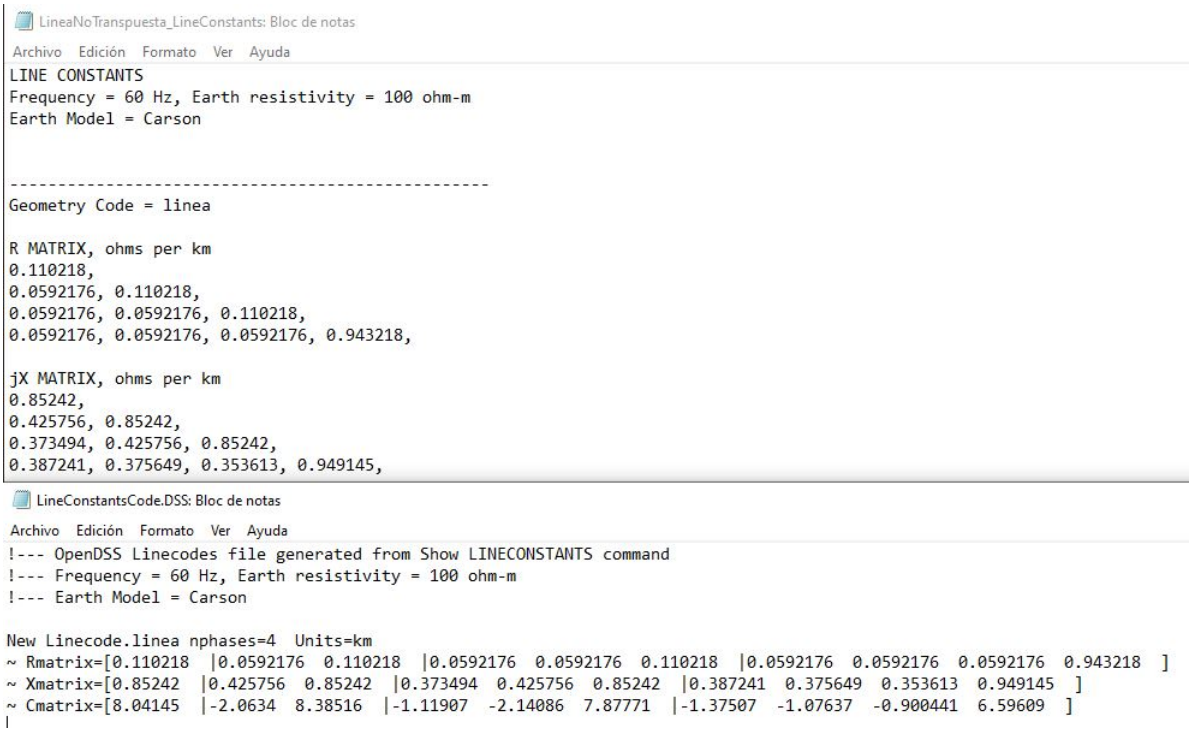


Figura 21: Matriz de impedancia para la línea no transpuesta

4.5. Ejemplo 5: Modelamiento de falla monofásica, bifásica, bifásica a tierra y trifásica.

Se tiene el sistema IEEE de 13 nodos obtenido de la página XENDEE[5], con una serie de modificaciones, como se ve en la Figura 22. El sistema está compuesto por un equivalente de red, un transformador y líneas de transmisión.

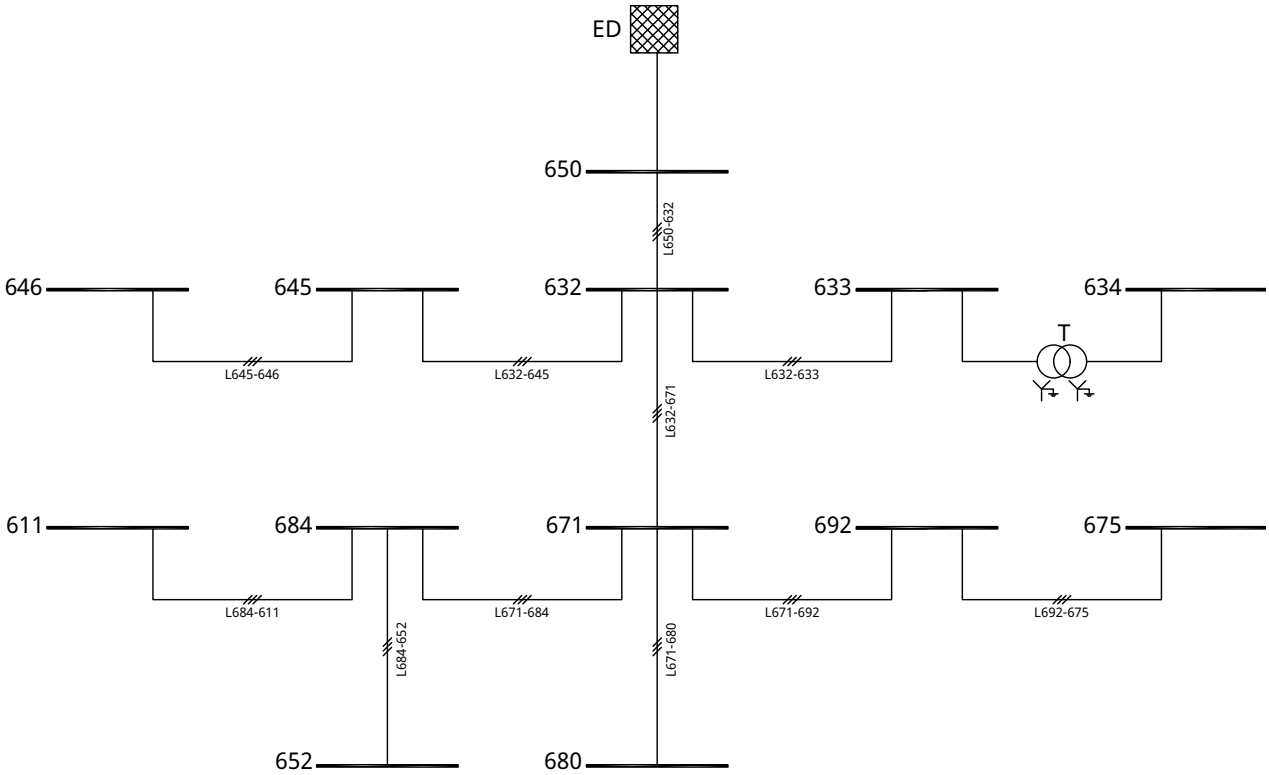


Figura 22: Esquema del sistema IEEE de 13 Nodos

4.5.1. Modelamiento del Equivalente de Red

Inicialmente se debe escribir el comando *"ClearAll"* para borrar los programas anteriormente ejecutados. Luego se debe declarar un nuevo circuito usando el comando *"New Circuit."* con el nombre de IEEE13. Posteriormente se comienza a escribir los comandos para el equivalente de red. El primer comando *"Bus1"* corresponde al nodo donde está conectado el equivalente, el comando *"BaseKv"* indica la tensión en kV, el comando *"pu"* es el valor en por unidad y el comando *"Angle"* es el comando para indicar el valor del ángulo en grados.

En una siguiente línea se debe escribir el símbolo "∼"para trabajar de forma continua el código. Se continua con el comando *"Frequency"* para indicar el valor de la frecuencia en Hertz y el comando *"Phases"* determina que el equivalente de red es trifásico.

Nuevamente en una siguiente línea se debe escribir el símbolo ∼ y se escribe el comando *"Mvasc3"* para ingresar la potencia de cortocircuito trifásica en MVA y el comando *"Mvasc1"* es la potencia de cortocircuito monofásico en MVA. En una siguiente línea se continua con el comando *"x1r1"* que corresponde a la relación x_1/r_1 en secuencia positiva y el comando *"x0r0"* corresponde a la relación x_0/r_0 en secuencia cero.

Algoritmo para el equivalente de red

```
ClearAll
New Circuit.IEEE13
!Equivalente de red
~ Bus1 = N650 Basekv = 4.16 pu = 1.00 Angle = 0.0
~ Frequency = 60.0 Phases = 3
~ Mvasc3 = 20000 Mvasc1 = 21000
~ x1r1 = 2 x0r0 = 1.6
```

4.5.2. Modelamiento de las líneas de transmisión

Para la impedancia de la línea se agrupa unos mismos valores para todas las líneas que conforman el presente sistema. Inicialmente se escribe el comando *"New Linecode."* con el nombre de ModeloLinea empezar a agrupar la impedancia de la línea. El siguiente comando es *"nphases"* con el número 3 para establecer que es un sistema trifásico y el comando *"BaseFreq"* corresponde a la frecuencia en hertz.
En una siguiente línea se debe escribir el símbolo "∼", para poder tranajar de forma líneal. Posteriormente se escribe los comandos *"R1"* y *"X1"* para ingresar la resistencia y reactancia en [Ω] para la secuencia positiva y los comandos *"R0"* y *"X0"* en [Ω] para la secuencia cero.

Algoritmo para la línea de transmisión

```
New Linecode.ModeloLinea          nphases=3          BaseFreq=60
~ R1=0.2121          X1=0.6430          R0=0.2153          X0=0.6325
```

Los siguientes comandos corresponden a la conexión de cada una de las líneas que conforman el sistema IEEE de 13 nodos. El primer comando que se debe escribir es *"New Line."* para describir cada una de las líneas y el nombre dado es el nodo desde donde inicia. El siguiente comando *"Bus1"* corresponde al nodo donde inicia la línea y el comando *"Bus2"* corresponde al nodo donde termina la línea. El comando *"Linecode"* trae los datos de la impedancia descrita con el nombre ModeloLinea, el comando *"Length"* corresponde a la longitud y por último el comando *"units"* indica las unidades en que está la longitud, que para el presente caso está en km. Para cada línea se debe escribir los mismos códigos y variar los nodos de conexión.

Algoritmo para la conexión de las líneas

```
!Modelo de la línea
New Line.Line_ND650_1   Bus1 = ND650   Bus2 = ND632   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND632_1   Bus1 = ND632   Bus2 = ND645   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND645_1   Bus1 = ND645   Bus2 = ND646   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND632_2   Bus1 = ND632   Bus2 = ND633   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND650_1   Bus1 = ND650   Bus2 = ND632   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND632_3   Bus1 = ND632   Bus2 = ND671   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND671_1   Bus1 = ND671   Bus2 = ND684   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND684_1   Bus1 = ND684   Bus2 = ND611   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND684_2   Bus1 = ND684   Bus2 = ND652   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND671_2   Bus1 = ND671   Bus2 = ND680   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND671_3   Bus1 = ND671   Bus2 = ND692   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND671_2   Bus1 = ND671   Bus2 = ND680   Linecode = ModeloLinea   Length = 0.405   units = km
New Line.Line_ND692_1   Bus1 = ND692   Bus2 = ND675   Linecode = ModeloLinea   Length = 0.405   units = km
```

4.5.3. Modelamiento del Transformador

Para el transformador, primero se debe declarar el comando *"New Transformer"* para ingresar el transformador. Seguido se debe declarar el comando *"Phases"* para indicar que el transformador es trifásico. El siguiente comando es *"windings"*, el cual define que el transformador está compuesto de 2 devanados y luego se declara el comando *"XHL"* que corresponderá con la impedancia de cortocircuito.

En una siguiente línea se debe escribir el símbolo "∼", para trabajar el código de forma continua. Después se escribe el comando *"Bus"* para establecer el nodo en que está conectado el transformador en el primario, el comando *"conn"* corresponde a la conexión, donde "Wye.[es] una conexión en estrella, el comando *"kv"* es la tensión en kV, el comando *"kva"* es la potencia aparente en kVA y el comando *" %r"* corresponde a las pérdidas de potencia en el devanado. Estos mismos se aplican para el devanado secundario, variando el devanado, el nodo de conexión y la tensión.

Algoritmo para el transformador

```
!Modelo del Transformador
New Transformer.T1 Phases = 3 Windings = 2 XHL = 2
~ wdg = 1 Bus = N633 conn = Wye kv = 4.16 kva = 500        %r=0.55
~ wdg = 2 Bus = N634 conn = Wye kv = 0.480 kva = 500       %r=0.55
```

4.5.4. Modelamiento de las Fallas de baja impedancia

Para el modelamiento de las fallas en el sistema IEEE de 13 nodos, inicialmente se debe escribir el comando *"New Fault."* y se asigna un nombre, acorde a la falla propuesta. El siguiente comando es *"phases"*, el cual indica el número de fases que están en cortocircuito. El comando *"bus1"* indica el nodo que está en falla, donde este comando sólo aplica para la falla monofásica y trifásica, en cambio para la falla bifásica y bifásica a tierra se debe agregar el comando *"bus2"*, agregando el nodo de falla. Por último se debe agregar el comando *"r"* para indicar el valor de la impedancia de falla en [Ω].

En el comando *"bus1"* al anexar el nodo de falla, se debe indicar las fases que están en falla y para el comando *"bus2"*, en el caso de la falla bifásica se debe anexar en la parte derecha del nodo en falla los números *".4.4"* para indicar una falla flotante y en el caso de la falla bifásica a tierra se debe anexar en la parte derecha del nodo de falla los números *".0.0"*, para indicar que hay una conexión a tierra.

Algoritmo para las fallas de baja impedancia

```
!Falla monofásica en la fase A
New Fault.F1    phases=1    bus1=ND645.1    r=0
!Falla Bifásica en las fases B y C
New Fault.F2    phases=2    bus1=ND645.2.3    bus2=ND645.4.4    r=0
!Falla bifásica a Tierra en las fases B y C
New Fault.F2T    phases=2    bus1=ND645.2.3    bus2=ND645.0.0    r=0
!Falla trifásica en las fases A,B y C
New Fault.F3    phases=3    bus1=ND645 r=0
```

4.5.5. Definición de Tensión base y Solución del Sistema

El siguiente algoritmo corresponde a la solución del sistema. Inicialmente se debe establecer las tensiones bases con el comando *"Set Voltagebases"* y dentro de unas llaves se escriben las tensiones del sistema. En una siguiente línea se escribirá el comando *"calv"* para estimar las tensiones en los nodos y por último el comando *"solve "* para solucionar el sistema.

Algoritmo para la solución del sistema

```
Set Voltagebases=[4.16 0.480]
calv
solve
```

4.5.6. Mostrar Resultados

Por último se ingresará los comandos para visualizar tensiones y corrientes de falla. En el caso del comando "Visualize Currents"la recomendación es ver cada falla a la vez y comentar las restantes, para que no se presenten cálculos erróneos. El comando *"Show"* permite desplegar un block de notas con los resultados, acorde si es tensión o corriente.

Algoritmo para visualizar resultados

```
!Visualize Currents Fault.F1
!Visualize Currents Fault.F2
!Visualize Currents Fault.F2T
Visualize Currents Fault.F3
Show Voltages
Show Currents
solve
```

Al haber compilado el código, se habilita una ventana la cual nos muestra las corrientes de falla en el nodo seleccionado, tal como se puede ver en la Figura 23.

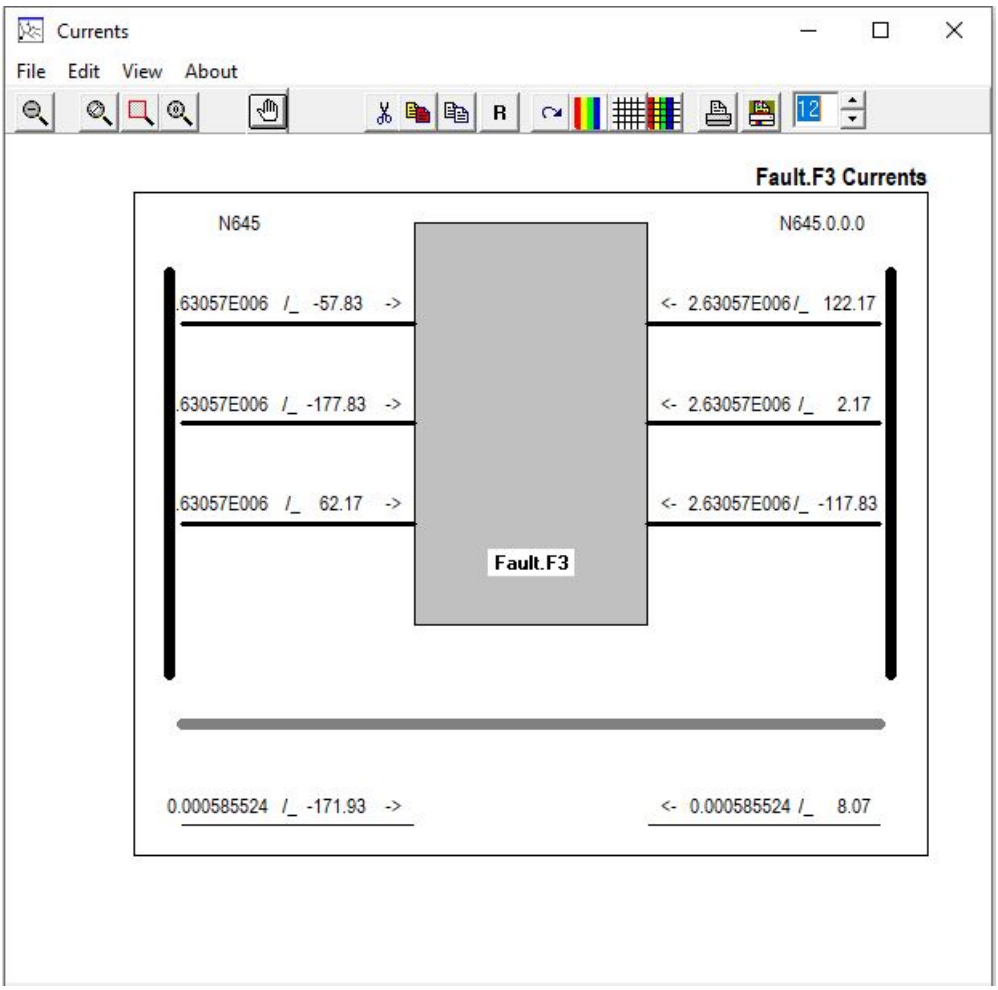


Figura 23: Corrientes de falla trifásica en el sistema IEEE de 13 nodos

5. Enlace con Matlab

Existe la posibilidad de trabajar en conjunto OpenDSS y Matlab, sin necesidad de tener hacer uso de la interfaz de OpenDSS. Esto permite ejecutar el código, extraer datos y analizarlos en Matlab. Además, es posible añadir y modificar elementos de un sistema previamente descrito en OpenDSS. La siguiente explicación, corresponde con algunos comandos básicos utilizados en Matlab.

5.1. Conexión de OpenDSS y Matlab

Para enlazar datos de OpenDSS con Matlab, inicialmente se escribe el siguiente código:

Algoritmo de enlace de OpenDSS con Matlab

```
clc
clear
DSSObj = actxserver('OpenDSSEngine.DSS');
if  DSSObj.Start(0)
disp('Unable to start the OpenDSS Engine');
return
end
```

El código "DSSObj"permite definir un objeto, el cual se usará para activar el servidor entre OpenDSS y Matlab junto con el comando "actxserver('OpenDSSEngine.DSS')". El condicional if se escribe para indicar si existe algún problema de conexión. Por ejemplo que OpenDSS no esté instalado, en pantalla nos mostrará el mensaje "Unable to start the OpenDSS Engine", por lo cual tendremos que proceder a instalarlo. Si no sale el mensaje de advertencia, la conexión fue exitosa.

5.2. Vinculación del circuito de OpenDSS

El siguiente código corresponde para activar el circuito ya descrito en OpenDSS. Se indica la ruta donde está el archivo ".dss", que para este caso está en el escritorio y por último se procede para que lo resuelva.

Algoritmo de vinculación de circuito

```
DSSText = DSSObj.Text;
DSSCircuit = DSSObj.ActiveCircuit;
DSSSolution = DSSCircuit.Solution;
DSSText.Command = 'Compile C:\Users\Escritorio\PRUEBAOPENDSS\NEWCIRCUIT.dss"';
DSSText.Command = 'solve';
```

5.3. Mostrar resultados

Para mostrar los resultados en formato de block de notas, como son tensiones, corrientes, flujos de cargas, entre otras, se escribe el comando "Show"de igual forma como se hace en OpenDSS y la variable que se necesite.

Algoritmo para mostrar resultados

```
DSSText.Command = ['Show currents'];
DSSText.Command = ['Show voltages'];
DSSText.Command = ['Show Powers'];
```

5.4. Exportar datos

Para guardar archivos en formato de excel, se escribe el comando "Export" y la variable que requiera exportar. Los siguientes comandos, exportan las valores de tensión y corriente para cada nodo, la tensión base en la que está calculada, la magnitud y el ángulo.

Algoritmo para exportar datos

```
DSSText.Command = 'Export Voltages Tensiones';
DSSText.Command = 'Export currents Corrientes';
```

5.5. Modificar datos de un circuito de OpenDSS desde Matlab

Cuando se compila un circuito en Matlab, previamente hecho en OpenDSS, este soluciona directamente lo descrito en el código de OpenDSS, pero si se quiere modificar algún dato desde Matlab, sin necesidad de hacerlo en OpenDSS, se debe usar los comandos "strcat" y "num2str".
El siguiente ejemplo corresponde a un equivalente de red, pero se quiere modificar algunos de sus datos, entonces vamos a ingresar en Matlab el siguiente código:

Algoritmo para modificar datos de OpenDSS desde Matlab

```
DSSText.Command = strcat('New Vsource.GD Bus1 = ',num2str(NGD),' Basekv = ',num2str(BkV),' pu = ',num2str(1.00),...
' Angle = ',num2str(0.0),' Frequency = ',num2str(60),' phases = ',num2str(3),' phases = ',num2str(3),...
' Mvasc3 = ',num2str(SCC3),' Mvasc1 = ',num2str(SCC1),' x1r1 = ',num2str(X1R1),' x0r0 = ',num2str(X0R0));
```

El comando "strcat" concatena horizontalmente el texto en sus argumentos de entrada[6] y el comando "num2str" convierte los números en un arreglo de caracteres. Entonces en algunas variables se tiene un valor ya definido, como es el valor en por unidad de 1, el ángulo que tiene un valor de 0 grados, la frecuencia de 60 Hz y el número de fases igual a 3. Pero hay otras que se les definió unas letras específicas, como es la potencia de cortocircuito trifásica y monofásica y la relación x1/r1 y x0/r0. Estos valores son llamados previamente, para que desde Matlab se le asigne cualquier valor y así al compilarlo nuevamente, este ya tomará el último valor asignado por el usuario.

Para modificar datos de un sistema en OpenDSS, también se puede aplicar por ejemplo para el transformador, las cargas, el modelamiento de fallas, entre otras. Se recomienda dejar dejar siempre unos espacios entre cada variable y una comilla sencilla, para evitar errores al compilar.

6. Otros recursos

En esta sección se mostrará algunos recursos que pueden complementarios para aprender más sobre OpenDSS.

- Canal de youtube de OpenDSS

  En el siguiente link `https://www.youtube.com/channel/UC4J6uTXtCGLkuNK8Xn_BQhA/featured`, se encontrará una serie de vídeos que explican varias temáticas sobre OpenDSS.

- Canal de youtube de OpenDSS realizado por el Ingeniero Paulo Radatz

  En el siguiente link `https://www.youtube.com/c/PauloRadatz`, encontraremos tutoriales sobre el modelamiento de elementos de un sistema de potencia, generación distribuida, uso de Python con OpenDSS, entre otros.

- Casos IEEE

  En la siguiente página `https://smartgrid.epri.com/SimulationTool.aspx`, encontraremos algunos casos de la IEEE, ya descritos en OpenDSS, los cuales se pueden copiar y verificar.

- Foro de consultas

  En la siguiente página `https://sourceforge.net/p/electricdss/discussion/861976/`, se encuentra el foro de OpenDSS, donde cualquier persona puede hacer preguntas y cualquier experto en el tema puede dar respuesta.

7. Créditos

El presente manual corresponde a una adaptación del manual de OpenDSS publicado por EPRI "ELECTRIC POWER RESEARCH INSTITUTE" en Junio de 2020, por los señores Roger C. Dugan, Davis Montenegro, donde se realiza un resumen de los códigos necesarios para el modelamiento de algunos elementos que conforman un sistema eléctrico de potencia, mejorando sus descripciones. Además, se extrae un ejemplo y se explica el código utilizado.

Referencias

[1] EPRI,"EPRI, Electrical Power Research Institute",2022.[Online]. Avalible: https://www.epri.com/pages/sa/opendss.

[2] R.Dugan,"OpenDSS Introductory Training Level 1",2009,pp. 50[Online]. Avalible: https://smartgrid.epri.com/doc/OpenDSS%20Level%201%20Training.pdf.

[3] http://svn.code.sf.net/p/electricdss/code/trunk/Distrib/Examples/Stevenson/StevensonPflow-3ph.dss.

[4] R.Dugan,D.Montenegro,"Reference Guide: The Open Distribution System Simulator (OpenDSS)"2020.[Avalible]:https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/Doc/OpenDSSManual.pdf

[5] "IEEE13 Node Feeder XENDEE Test Cases".Avalible: https://xendee.com/ieee-13-node-test-case/#:~:text=IEEE%2013%20Node%20Test%20Feeder,software%20operating%20at%204.16%20kV.

[6] "Concatenar cadenas horizontalmente - MATLAB strcat- MathWorks América Latina". MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink. Avalible: https://la.mathworks.com/help/matlab/ref/strcat.html.

3.5. Elaboración de videos sobre el uso de OpenDSS, modelamiento de elementos asociados a un sistema de Potencia y modelamiento de Fallas de baja impedancia.

Se graba y edita una serie de vídeos correspondientes al uso del software OpenDSS, donde se explica que es el software, dónde se puede descargar, se hace una introducción de la interfaz y se realiza el modelamiento del sistema IEEE de 13 nodos. En este sistema planteado, se explica cada uno de los elementos que lo conforman, se corre un flujo de carga realizando una modificación en el sistema y por último se calcula las tensiones y corrientes de falla en componentes de secuencia y fase en una de las líneas del presente sistema. La lista de vídeos elaborados es la siguiente:

- Introducción a OpenDSS
- Conociendo OpenDSS
- Modelo del equivalente de red
- Modelo del transformador
- Modelo de la línea
- Modelo de la carga
- Flujo de carga
- Cálculo de falla monofásica
- Cálculo de falla bifásica
- Cálculo de falla bifásica a tierra
- Cálculo de falla trifásica

4. Conclusiones

- Se realiza una investigación sobre los métodos de identificación de fallas de baja impedancia, encontrando que la gran mayoría usan algoritmos que tienen una efectividad mayor al 90 % identificando la falla, además algunos de ellos fueron adaptados para localizar fallas bajo ciertas condiciones. Además se buscó información sobre el uso del software OpenDSS, encontrando información útil aplicada al modelamiento de los componentes que conforman un sistema eléctrico de potencia.
- El modulo de identificación de fallas de baja impedancia implementado en la aplicación APP LF-UD presenta resultados de manera satisfactoria, donde el algoritmo desarrollado en Matlab realiza un análisis de las magnitudes y ángulos de las corriente en componentes de fase y determina el tipo de falla ocurrida en algún punto asignado para los sistemas IEEE de 13 y 34 nodos, junto con una representación gráfica del comportamiento de cada corriente de falla en un diagrama vectorial.
- El uso del software libre OpenDSS permitió realizar el modelamiento de dos sistemas eléctricos de potencia, además de programar y calcular fallas simétricas y asimétricas, se logró establecer un enlace con el software de Matlab , permitiendo modificar parámetros del sistema, desde una aplicación previamente desarrollada, sin necesidad de para extraer y analizar datos con el fin de determinar el tipo de falla.

5. Trabajos futuros

En la aplicación APP LF-UD para futuros proyectos, se puede implementar algoritmos más robustos aplicados para localizar e identificar de fallas en líneas de transmisión, ingresando nuevas condiciones de falla, coordinar protecciones, modificar parámetros de transformadores y equivalentes de red .

Referencias

[1] S. Edición and T. Wildi, "Theodore Wildi Wildi Sexta Edición Máquinas Eléctricas y Sistemas de Potencia." [Online]. Available: www.pearsoneducacion.net

[2] S. Amini, F. Pasqualetti, M. Abbaszadeh, and H. Mohsenian-Rad, "Hierarchical location identification of destabilizing faults and attacks in power systems: A frequency-domain approach," IEEE Transactions on Smart Grid, vol. 10, no. 2, pp. 2036–2045, 2019, doi: 10.1109/TSG.2017.2787690.

[3] J. V. Landa, "UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA REDES ELÉCTRICAS 2 DIVISIÓN DE INGENIERÍA ELÉCTRICA DEPARTAMENTO DE INGENIERÍA ELÉCTRICA DE POTENCIA."

[4] H. H. Goh et al., "Fault location techniques in electrical power system: A review," Indonesian Journal of Electrical Engineering and Computer Science, vol. 8, no. 1. Institute of Advanced Engineering and Science, pp. 206–212, Oct. 01, 2017. doi: 10.11591/ijeecs.v8.i1.pp206-212.

[5] F. G. Y. Souhe, A. T. Boum, P. Ele, C. F. Mbey, and V. J. F. Kakeu, "Fault Detection, Classification And Location In Power Distribution Smart Grid Using Smart Meters Data," Journal of Applied Science and Engineering (Taiwan), vol. 26, no. 1, pp. 23–34, 2022, doi: 10.6180/jase.202301_26(1).0003.

[6] A. Rahmati and R. Adhami, "A fault detection and classification technique based on sequential components," in IEEE Transactions on Industry Applications, Nov. 2014, vol. 50, no. 6, pp. 4202–4209. doi: 10.1109/TIA.2014.2313652.

[7] M. Jamil, R. Singh, and S. K. Sharma, "Fault identification in electrical power distribution system using combined discrete wavelet transform and fuzzy logic," Journal of Electrical Systems and Information Technology, vol. 2, no. 2, pp. 257–267, Sep. 2015, doi: 10.1016/j.jesit.2015.03.015.

[8] J. P. N. Gonzalez, L. E. G. Castañon, R. Morales-Menendez, A. el Hajjaji, and A. Rabhi, "Fault diagnosis of electrical power systems using soft computing," in IFAC Proceedings Volumes (IFAC-PapersOnline), 2013, pp. 863–868. doi: 10.3182/20130204-3-FR-2033.00037.

[9] A. Asadi Majd, H. Samet, and T. Ghanbari, "k-NN based fault detection and classification methods for power transmission systems," Protection and Control of Modern Power Systems, vol. 2, no. 1, Dec. 2017, doi: 10.1186/s41601-017-0063-z.

[10] P. Balakrishnan and S. Gopinath, "A new intelligent scheme for power system faults detection and classification: A hybrid technique," International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, vol. 33, no. 5, Sep. 2020, doi: 10.1002/jnm.2728.

Anexos

Contenido audiovisual

Los siguientes videos corresponden a una explicación sobre el software OpenDSS, su modo de instalación, modelamiento del sistema IEEE de 13 nodos y el modelamiento de fallas en sus líneas.

Descripción del video	Enlace
Introducción a OpenDSS	https://youtu.be/jpcCU98L0dE
Conociendo OpenDSS	https://youtu.be/j5XVEnhXjuQ
Modelo del equivalente de red	https://youtu.be/r7-fI9FbnjM
Modelo del transformador	https://youtu.be/2V2gx6dSvO0
Modelo de la línea	https://youtu.be/EeED5EHDSHY
Modelo de la carga	https://youtu.be/tZREE5Zfm4M
Flujo de carga	https://youtu.be/VUlmuEyPl7Y
Cálculo de falla monofásica	https://youtu.be/N5yY32JOrCk
Cálculo de falla bifásica	https://youtu.be/ztpbutQQhNU
Cálculo de falla bifásica a tierra	https://youtu.be/_ROuwSFgPcs
Cálculo de falla trifásica	https://youtu.be/pMt1Y5f-ixA