



UNIVERSIDAD NACIONAL
de MAR DEL PLATA



UNIVERSIDAD NACIONAL DE MAR DEL PLATA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

PROYECTO FINAL DE GRADO

“Modelado de redes para generación de eventos de tensión y validación de algoritmos de detección y clasificación de eventos”

Autor:

Nahuel Matías Jauregui

DNI: 39098288 Legajo: 10781

Mail de contacto: nahuel.m.jauregui@gmail.com

Carrera: Ingeniería Electromecánica

Director:

Dr. Ing. Jorge Strack

Co-director:

Dr. Ing. Carlos Orallo

Evaluadores:

Ing. Guillermo di Mauro

Mg. Ing. Daniel Anaut

Ing. Guillermo Murcia

Mar del Plata, 25 de abril de 2023



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



UNIVERSIDAD NACIONAL
de MAR DEL PLATA



UNIVERSIDAD NACIONAL DE MAR DEL PLATA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

PROYECTO FINAL DE GRADO

“Modelado de redes para generación de eventos de tensión y validación de algoritmos de detección y clasificación de eventos”

Autor:

Nahuel Matías Jauregui

DNI: 39098288 Legajo: 10781

Mail de contacto: nahuel.m.jauregui@gmail.com

Carrera: Ingeniería Electromecánica

Director:

Dr. Ing. Jorge Strack

Co-director:

Dr. Ing. Carlos Orallo

Evaluadores:

Ing. Guillermo di Mauro

Mg. Ing. Daniel Anaut

Ing. Guillermo Murcia

Mar del Plata, 25 de abril de 2023

Agradecimientos

Quiero agradecer a mi familia, especialmente a mi padre, madre, hermana y abuela por todo el amor y apoyo que siempre me brindaron.

También a Romina, mi compañera de vida, por todo su cariño y sus palabras que me ayudan a seguir cumpliendo mis metas.

Por supuesto a mis amigos por sus consejos y los momentos compartidos.

Finalmente, agradecer a mis directores por acompañarme y guiarme a lo largo de este proyecto.

Índice

1	Estado de arte	7
1.1	Redes eléctricas convencionales	7
1.1.1	Centrales de Generación	8
1.1.2	Subestación	10
1.1.3	Dispositivos de regulación	11
1.1.4	Redes de transmisión, subtransmisión y distribución	14
1.2	Redes eléctricas inteligentes	15
1.3	Calidad de la Energía	18
1.3.1	Eventos de tensión	20
1.3.2	Proceso de medición	26
1.4	Alimentadores de prueba	29
1.4.1	Modelos de alimentadores de prueba	31
1.4.2	Limitaciones de los Alimentadores de prueba	35
1.5	Selección de la red a simular	37
1.6	Conclusión	40
1.7	Bibliografía	42
2	Modelado computacional	45
2.1	Software ATP - ATPDraw	45
2.1.1	Cálculo numérico	46
2.1.2	Biblioteca de modelos de ATP	49
2.1.3	Rutinas de soporte	50
2.1.4	Preprocesador ATPDraw	51
2.2	Modelos de elementos de la red	52
2.2.1	Transformadores	53
2.2.2	Líneas eléctricas	60
2.2.3	Lenguaje MODELS	83
2.3	Conclusión	86

2.4	Bibliografía	88
3	Cálculos y automatización	91
3.1	Introducción.....	91
3.2	Algoritmo de inicialización	92
3.3	Algoritmo de convergencia	98
3.4	Algoritmo de fallas.....	102
3.4.1	Cálculo de valores RMS y fasores con MODELS.....	108
3.4.2	Tratamiento de datos	113
3.5	Algoritmo de clasificación de eventos.....	115
3.5.1	Clasificación ABC	115
3.5.2	Algoritmo de secuencias absolutas.....	121
3.5.3	Algoritmo de secuencias absolutas mejorado	128
3.6	Conclusión	130
3.7	Bibliografía	132
4	Alimentador de prueba de 34 nodos.....	135
4.1	Características de la red.....	135
4.1.1	Líneas eléctricas.....	136
4.1.2	Cargas eléctricas	139
4.1.3	Banco de capacitores	143
4.1.4	Transformadores	144
4.1.5	Reguladores de tensión	146
4.2	Resultados en régimen estacionario.....	150
4.2.1	Resultados de las tensiones	150
4.2.2	Resultados de las corrientes.....	153
4.3	Conclusión	155
4.4	Bibliografía	157
5	Desempeño de algoritmos de detección y clasificación de eventos de tensión.....	159
5.1	Análisis de resultados.....	159
5.1.1	Estación de medición 1	162

5.1.2	Estación de medición 2.....	173
5.1.3	Estación de medición 5.....	179
5.1.4	Estación de medición 6.....	181
5.1.5	Estación de medición 7.....	185
5.1.6	Estación de medición 8.....	191
5.1.7	Estación de medición 9.....	193
5.1.8	Estación de medición 10.....	195
5.2	Análisis de resultados sobre muestra representativa.....	196
5.2.1	Estación de medición 1: Nodo 800.....	198
5.2.2	Estación de medición 2: Nodo 850.....	201
5.2.3	Estación de medición 7: Nodo 888.....	203
5.2.4	Estación de medición 8: Nodo 834.....	206
5.2.5	Desempeño general.....	208
5.3	Conclusión	210
5.4	Bibliografía	213
6	Anexo.....	214
6.1.1	Algoritmo de inicialización.....	214
6.1.2	Algoritmo de convergencia.....	218
6.1.3	Algoritmo de fallas	226
6.1.4	Códigos de generación de gráficos.....	230
6.1.5	Algoritmo de detección de eventos de tensión	247

Resumen

El objetivo de este trabajo es el de implementar algoritmos de detección y clasificación de eventos de tensión en una red eléctrica validada luego de generar fallas eléctricas en distintos puntos del sistema.

Para cumplir este objetivo se realizó un relevamiento de las redes eléctricas estandarizadas disponibles por el organismo IEEE (*Institute of Electrical and Electronics Engineers*) y se analizaron sus características principales. En conjunto, se relevó una muestra de 100 papers de los últimos 10 años donde estas redes normalizadas eran implementadas. A partir de esto, se realizó una estadística que ayudó a seleccionar la red eléctrica utilizada durante este trabajo. Se optó por utilizar el Alimentador de prueba de 34 nodos-IEEE.

Luego se desarrollaron las bases del modelado computacional en redes eléctricas tomando como referencia al software ATP-ATPDraw. El objetivo fue seleccionar los modelos computacionales para cada elemento de la red que mejor desempeño presentan en este tipo de simulaciones.

Luego de analizar el comportamiento de la red eléctrica simulada en régimen estacionario y de comparar los resultados obtenidos con los brindados por la IEEE, se llegó a la conclusión de que el alimentador de prueba de 34 nodos fue validado satisfactoriamente. Para llevar a cabo esta validación, se desarrollaron algoritmos que permitieron parametrizar a la red eléctrica y realizar mediciones de tensión y corriente en diferentes puntos de la misma.

Luego se prosiguió con la generación de fallas eléctricas en diferentes puntos de la red. Para poder llevar a cabo las simulaciones de las fallas eléctricas, se desarrollaron una serie de algoritmos que agilizaron el proceso de simulación y permitieron generar una base de datos con todos los resultados obtenidos.

Finalmente se aplicaron los algoritmos de detección y clasificación de eventos de tensión ASA y eASA sobre la base de datos de resultados obtenidos, analizando sus desempeños y realizando una comparativa entre ellos.

CAPÍTULO 1

MODELADO DE REDES

ESTADO DE ARTE



1 Estado de arte

Resumen

En este capítulo se buscará contextualizar la actualidad de las redes eléctricas de distribución y como estas redes convencionales están sufriendo cambios hacia las redes eléctricas inteligentes. A partir de esto se analizarán cuáles son las problemáticas de este cambio y cómo estos afectan a la calidad de la energía.

Entre las perturbaciones que sufren las redes eléctricas, se profundizará en los eventos de tensión, sus causas y consecuencias, y en cómo son detectados y clasificados mediante algoritmos específicos.

Además, se buscará recopilar las redes eléctricas sobre las cuales se simulan estas perturbaciones en la red, es decir los alimentadores de prueba o *node test feeder*, sus características principales, sus falencias y sus usos en el ámbito de la investigación y desarrollo. Posteriormente, se seleccionará y validará una de estas redes eléctricas de prueba normalizadas, para posteriormente utilizarla en simulaciones de fallas eléctricas. El objetivo final será analizar el desempeño del algoritmo ASA [10] y del algoritmo eASA [63] en la clasificación de los eventos de tensión generados.

1.1 Redes eléctricas convencionales

Una red eléctrica convencional es un sistema físico que conecta los diferentes generadores de energía eléctrica con los consumidores finales a través de líneas de transmisión, subtransmisión, y distribución. Estas redes eléctricas han ido evolucionando a lo largo de los años ante la necesidad de abastecer una creciente demanda energética y a su vez un aumento en las distancias del transporte de esta energía a raíz de áreas en constante expansión [1].



Figura 1-1. Esquema de una Red Eléctrica Convencional.

La topología de una red eléctrica convencional puede dividirse en los siguientes grupos, donde cada uno de estos elementos operando en conjunto conforman al sistema:

1.1.1 Centrales de Generación

Las Centrales de Generación son las encargadas de producir energía eléctrica a gran escala por medio de generadores que convierten energía térmica (combustión de carbón, gas, petróleo, etc.), energía nuclear o energía renovable (hidráulica, eólica, solar fotovoltaica, solar de concentración, biomasa, undimotriz, etc.) en energía eléctrica.

A continuación, se describe brevemente el funcionamiento de algunos de los métodos de generación eléctrica:

- Central térmica: el combustible es carbón pulverizado o gas natural, y una vez mezclado con aire, se quema en calderas que generan vapor. El vapor a alta presión y temperatura impulsa una turbina, que a su vez hace girar el generador eléctrico, convirtiendo energía mecánica en energía eléctrica.
- Planta de energía nuclear: se utiliza uranio enriquecido para producir fisión atómica, la cual calienta el agua y produce vapor. El vapor generado impulsa una turbina acoplada al generador eléctrico.
- Centrales Hidroeléctricas: se utilizan represas que aumentan el nivel del agua en un río y luego se genera un flujo de agua controlado para impulsar una turbina hidráulica que, a su vez, impulsa un generador eléctrico. Para nombrar algunos ejemplos, en Argentina las cinco principales centrales hidroeléctricas son Yacretá (3100 [MW] – Río Paraná), Salto Grande (1980 [MW] – Río Uruguay), Piedra del Águila (1400 [MW] – Río Limay), El Chocón (1260 [MW]- Río Limay) y Alicurá (1050 [MW] – Río Limay).
- Turbina de gas: el gas natural se mezcla con aire y se quema, generando un flujo de gas de alta velocidad que impulsa la turbina, que hace girar el generador eléctrico.
- Planta de energía de ciclo combinado: esta planta contiene una turbina de gas que genera electricidad y, a su vez, como parte del ciclo de combustión, el escape de la turbina libera gas a alta temperatura. El gas luego alimenta un intercambiador de calor para precalentar el aire comburente de la caldera de una central térmica, cuyo vapor generado impulsa una segunda turbina, que impulsa el segundo generador. Esta operación en dos etapas aumenta la eficiencia de la planta.
- Parques eólicos: la energía eólica se obtiene al convertir el movimiento de las palas de un aerogenerador en energía eléctrica. Un aerogenerador es un generador eléctrico movido por una turbina accionada por el viento. Está conformado por la

torre, un sistema de orientación ubicado al final de la torre, un armario de acoplamiento a la red eléctrica, una góndola que es el armazón que resguarda a los componentes mecánicos del molino y que sirve de base a las palas, un eje y el mando del rotor, un freno, un multiplicador, el generador eléctrico y el sistema de regulación eléctrica. [13]

- Sistema fotovoltaico: es el conjunto de equipos eléctricos y electrónicos que producen energía eléctrica a partir de la radiación solar. El principal componente de este sistema es el módulo fotovoltaico, compuesto por células capaces de transformar la energía luminosa incidente en energía eléctrica de corriente continua. El resto de equipos incluidos en un sistema fotovoltaico depende en gran medida de la aplicación a la que está destinado. A grandes rasgos, los sistemas fotovoltaicos pueden clasificarse en tres grandes grupos: conectados a red (*grid connected*), autónomos (*off-grid*) y de bombeo. Para permitir el correcto acoplamiento con la red eléctrica estos sistemas incorporan un equipo inversor que adecúa la potencia producida por el generador fotovoltaico a las condiciones de la red convencional.[14]

Los generadores eléctricos nombrados previamente, en su mayoría, constan de máquinas rotativas sincrónicas y/o inversores electrónicos, ambos controlados para mantener una tensión con amplitud y frecuencia estables, inyectando a la red eléctrica cantidades controladas de energía activa y reactiva. En una central eléctrica, varios generadores operan en paralelo, estando conectados en un punto común proporcionando la energía total demandada. [2]

Con el fin de poner en contexto el estado actual de la Generación Eléctrica en Argentina se presenta un resumen confeccionado por CAMESSA [15] de la potencia instalada en el país, donde se puede observar un incremento considerable de la potencia instalada del tipo eólica y solar en los últimos años:

Tabla 1-1. Potencia instalada en Argentina. [15]

Potencia instalada [MW]	2017	2018	2019	2020	2021	2022
Térmico	22896	24.531	24.562	25.365	25.398	25.275
Hidráulico	10746	10.790	10.812	10.834	10.834	10.834
Hidro Renovable	496	498	498	510	511	524
Nuclear	1.755	1.755	1.755	1.755	1.755	1.755
Eólico	227	750	1.609	2.623	3.291	3.309
Solar	8	191	439	759	1.060	1.086
Biomasa/Biogas	22	23	44	105	139	143

Total	36.150	38.538	39.719	41.951	42.989	42.927
--------------	--------	--------	--------	--------	--------	--------

En la Figura 1-2 se muestra como evolucionó en los últimos años la potencia instalada de energía eólica y solar en Argentina.

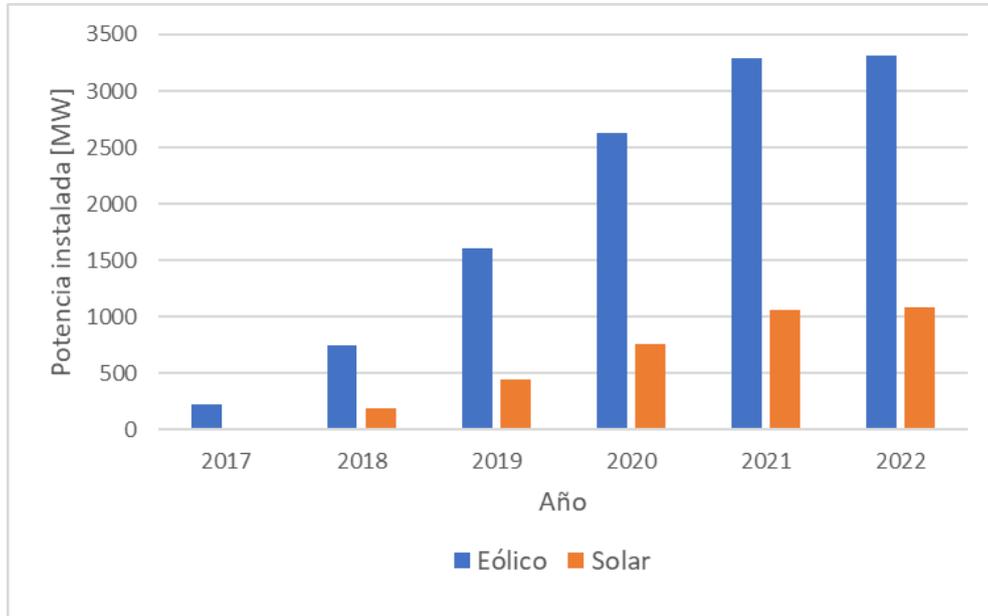


Figura 1-2. Evolución de potencia instalada de energía eólica y solar en Argentina.

1.1.2 Subestación

La operación segura de los sistemas eléctricos requiere que los interruptores abran las líneas automáticamente en caso de falla, o manualmente cuando la operación lo requiera. A modo de ejemplos, en la Figura 1-3 se muestra el diagrama de conexión simplificado de una estación generadora:

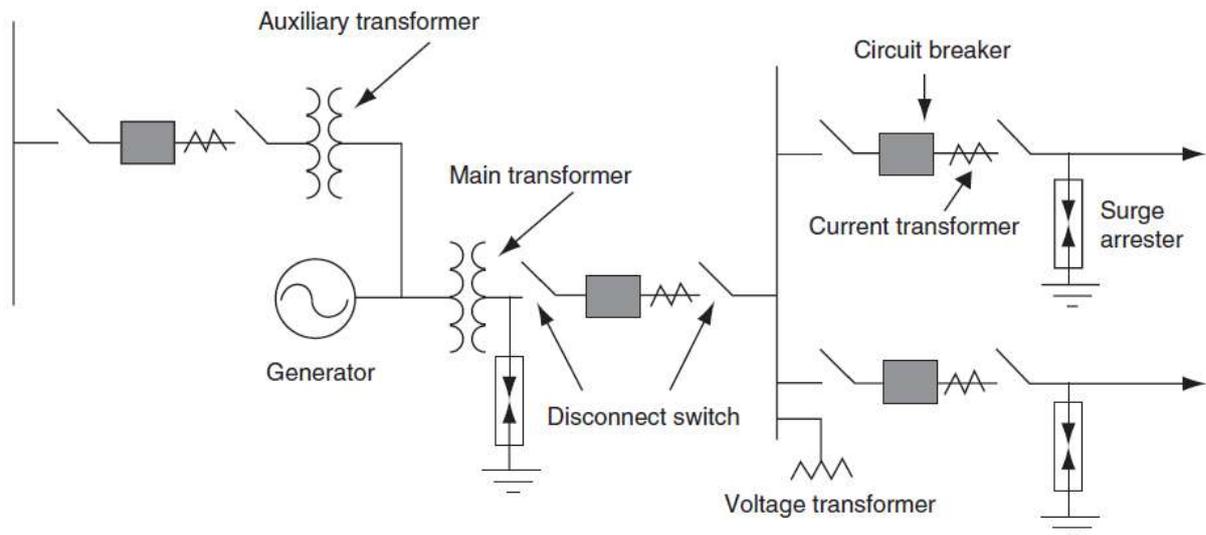


Figura 1-3. Esquema de conexión simplificado de una estación generadora. [13]

El generador está conectado directamente al devanado de baja tensión del transformador principal, y el devanado de alta tensión está conectado al bus a través de un interruptor (interrumpe la gran corriente de cortocircuito de los generadores), un seccionador de desconexión y un transformador de corriente. Otros elementos como los transformadores de potencial (PT) y los transformadores de corriente (CT) son utilizados para reducir la tensión y la corriente respectivamente a valores bajos, típicamente 120 V y 5 A, utilizados para el control de relés de protección y elementos de medición. [13]

1.1.3 Dispositivos de regulación

En los sistemas eléctricos se puede controlar el voltaje y la corriente. Algunos de los métodos normalmente utilizados son desarrollados en la Sección 5.3 del libro de Short [19]:

- Transformadores de cambio de tomas de carga de subestación (LTC)
- Reguladores de tensión de bus o alimentador de subestación
- Reguladores de voltaje de línea
- Condensadores fijos y conmutados

Los transformadores de potencia con conmutadores de tensión bajo carga (LTC) se utilizan para cambiar la relación de transformación y por ende controlar la tensión en el lado secundario.

Por otro lado, se encuentran los reguladores de tensión, los cuales son dispositivos que proporcionan una salida de voltaje constante cuando la tensión de entrada y las corrientes de carga son variables. Estos reguladores se sitúan en puntos estratégicos de la red, principalmente en zonas donde mantener el perfil de tensión en los niveles aceptables sea difícil. En general, las características más importantes de un regulador de tensión son:

- Regulan $\pm 10\%$ de tensión (en la mayoría de los casos).
- Tiene un total de 33 pasos cuya variación es de $5/8\%$ de voltaje por toma. Sin embargo, también se fabrican reguladores con cantidades menores de pasos.
- Estos pasos se distribuyen de la siguiente manera: 16 pasos en la dirección de Elevación, 16 pasos en la dirección de Bajada y una posición Neutral.

Las partes básicas son:

- Autotransformador: un transformador en el que parte de un devanado es común tanto al devanado primario como al secundario.
- *Load Tap Changer* (Cambiador de tomas de carga): un interruptor diseñado para funcionar bajo carga para cambiar la configuración de una bobina de transformador
- Control del regulador de voltaje: un control que detecta el sistema y controla automáticamente el cambiador de tomas.

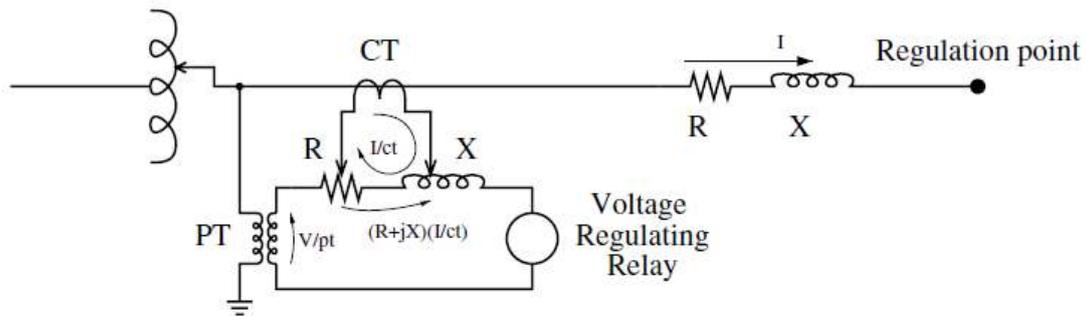


Figura 1-4. Circuito del compensador de caída de tensión. [19]

Para el correcto funcionamiento se deben configurar parámetros de Nivel de tensión (*Set Voltage* o SV por sus siglas en inglés) cuyo valor será el deseado y el control del regulador agregará o quitará tomas para acercarse lo más posible a este valor de tensión. También se debe configurar el rango de banda ancha (*Bandwidth* o BW) que representa el rango de voltaje total alrededor del voltaje establecido (SV) que el control considerará aceptable. El rango de voltaje aceptable se define como $\text{Rango} = \text{SV} \pm 1/2 \text{ BW}$. Finalmente, el tiempo de retraso (*Time Delay*), que representa la cantidad de segundos que espera el control, desde el comienzo de una condición fuera de banda, antes de iniciar un cambio de toma. [18] Normalmente los reguladores suelen configurarse en cascada, como se observa en la Figura 1-5:

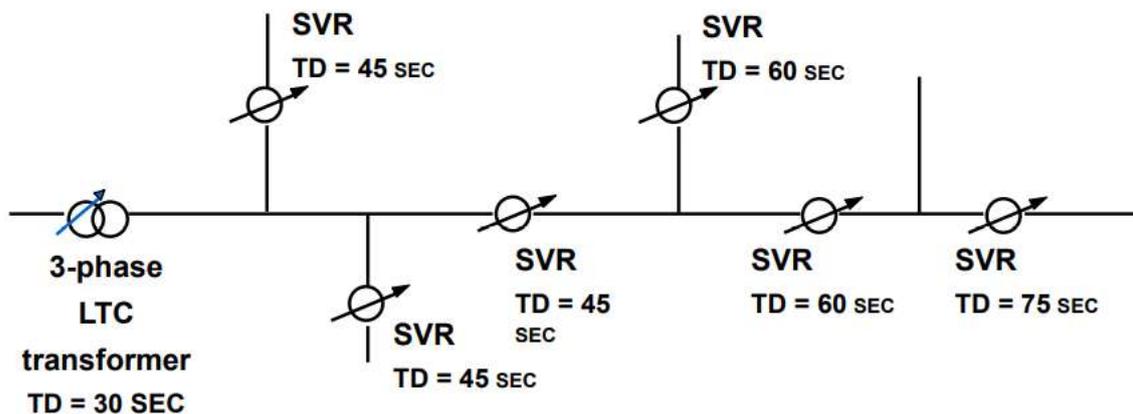


Figura 1-5. Reguladores de tensión en cascada, junto a los tiempos de delay configurados para un correcto funcionamiento.

Para el correcto funcionamiento se deben seguir las siguientes reglas:

- Cada regulador aguas abajo requiere tiempos de delay mayores a los de los reguladores aguas arriba.
- El tiempo mínimo de retardo de un regulador al siguiente en cascada es de 15 segundos

Respecto al método con capacitores, al instalarlos en la red se busca disminuir la potencia reactiva total del sistema mejorando el factor potencia. La potencia aparente disminuye (liberación de capacidad de la red) y, en consecuencia, la corriente resultante es menor, por lo que las pérdidas por efecto Joule (I^2R) también lo son. Otra razón por la que son instalados estos capacitores es mejorar el perfil de tensión de la red, cuya diferencia principal con los reguladores de tensión es que el banco de capacitores cambia el perfil de voltaje aguas arriba respecto del sitio donde están instalados, como se observa en la Figura 1-6. [19]

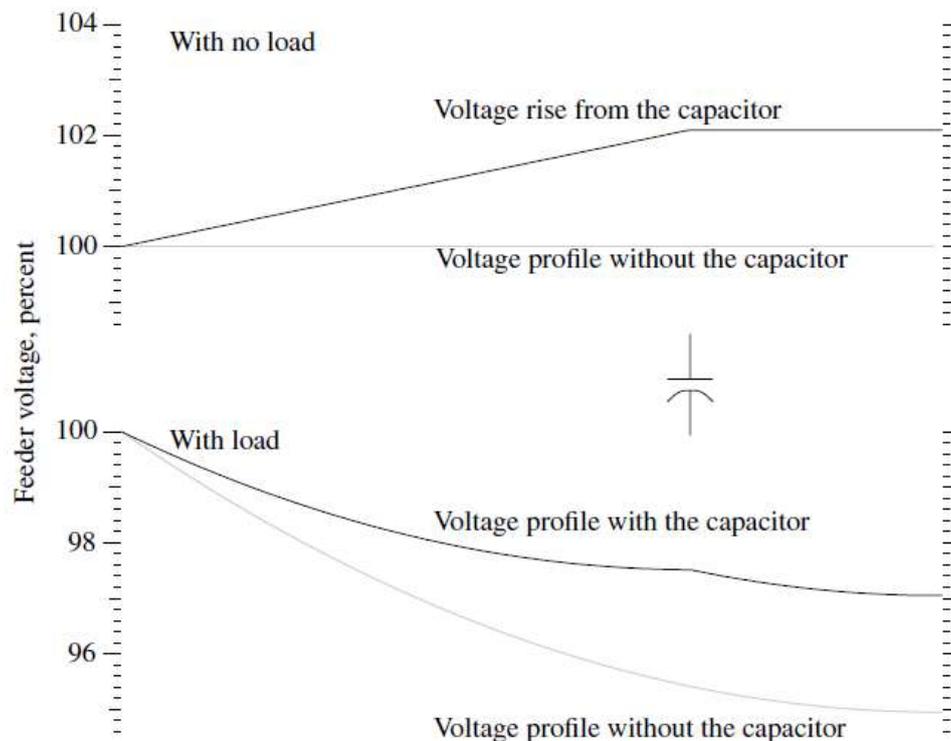


Figura 1-6. Perfil de tensión luego de conectar un banco de capacitores. [19]

En este método, se encuentra el Compensador estático de Var (SVC) que consta de un banco de capacitores conmutados y una inductancia controlada por tiristores que regula de forma continua la potencia reactiva. [13]

Los capacitores suelen instalarse en puntos estratégicos de la red, llamados capacitores de alimentación (*Feeder Capacitors*), o bien pueden instalarse en subestaciones. Las diferencias más notables son:

Tabla 1-2. Comparativa entre capacitores instalados en la subestación y los distribuidos en la red. [19]

Ventajas	desventajas
Feeder Capacitors	
Reduce las pérdidas de línea Reduce la caída de voltaje a lo largo del alimentador Libera capacidad del alimentador Costo más bajo	Más difícil de controlar de forma fiable La ubicación y la potencia reactiva son importantes
Substation Capacitors	
Se tiene mayor control El mejor lugar si se necesitan Vars para el soporte de voltaje del sistema	No reducen las pérdidas de línea No reducen caída de voltaje del alimentador Mayor costo

Los capacitores de alimentación están más cerca de las cargas, por lo tanto, liberan capacidad de la red de una manera más efectiva (disminuyendo la potencia aparente), mejoran los perfiles de voltaje y reducen las pérdidas de línea. Estos efectos se aprecian en alimentadores largos que tienen pérdidas de línea y caídas de voltaje considerables. Los capacitores de subestación son mejores cuando se necesita un control más preciso sobre los Vars aportados a la red. Sin embargo, esta ventaja disminuyó en los últimos años gracias a la evolución de las tecnologías de comunicación y control, permitiendo a los operadores controlar los capacitores de alimentación con una eficacia similar.

1.1.4 Redes de transmisión, subtransmisión y distribución

El sistema de transmisión eléctrica (redes de transmisión, subtransmisión y distribución) se encarga de la interconexión de las centrales productoras de energía eléctrica o centrales generadoras con las cargas (zonas de consumo). En la mayoría de las líneas de transmisión se utiliza un sistema de CA trifásico cuyas frecuencias de funcionamiento son 60 y 50 Hz. Por nombrar unos pocos ejemplos, la frecuencia de funcionamiento de EE.UU y Brasil es de 60 Hz, mientras que en Europa, parte de Asia, y Argentina es de 50 Hz. [13]. En general, los centros de consumo se encuentran a grandes distancias de los centros de generación de energía eléctrica. Por este motivo, para transportar la energía eléctrica generada se eleva la tensión utilizando transformadores de elevación, obteniendo una disminución en la corriente transportada y, en consecuencia, una disminución en las pérdidas de energía por efecto Joule, es decir las pérdidas I^2R . Las líneas de transmisión operan en rangos que se denominan alta tensión (de 35 kV a 230 kV) y muy alta tensión (EHV) (de 245 kV a 1200 kV) de acuerdo a la norma IEC 60038. [3]. Estas líneas de transmisión transportan la energía eléctrica hacia subestaciones donde se disminuyen los niveles de tensión. Las líneas de

subtransmisión conectan subestaciones de alto voltaje con muchas estaciones de distribución locales ubicadas generalmente dentro de las ciudades. [13]

Cuando la energía eléctrica llega a las áreas de consumo urbanas o rurales, es reducida a niveles de tensión más bajos a fin de disminuir los costos de aislamiento y aumentar la seguridad de la población. Dependiendo las necesidades del usuario se distinguen tres niveles de tensión: alta tensión para parques industriales o zonas con varias industrias en un rango típico de 35 kV a 230 kV, que normalmente son alimentadas directamente por una línea de subtransmisión o una línea de distribución dedicada; media tensión para industrias individuales en un rango de 1 kV a 35 kV; y baja tensión para consumidores residenciales y comerciales, entre 100 V a 1000 V.[3] Estas redes pueden ser radiales, anilladas o malladas, describiéndose brevemente a continuación [4] [5] [16]:

La red radial es el sistema más utilizado para las redes de distribución de energía, siendo la topología más simple y económica para una red eléctrica. Es una estructura del tipo cascada que comienza en un nodo raíz, lugar donde se conecta la generación. Este nodo alimenta una línea principal que, a su vez, alimenta otras líneas en forma de derivación. Esta configuración tiene una desventaja importante, y es que, en caso de haber una falla, se verán afectadas las tensiones y corrientes de toda la red. Por ejemplo, un fallo en la red de transmisión dará lugar a una falla en la red de distribución, lo cual produce una interrupción del servicio para un gran número de usuario finales.

La red de distribución en anillo sigue una estructura de bucle que enlaza el servicio desde una o varias fuentes a través de una colección de cargas y regresa a la/s fuente/s, formando una estructura de circuito cerrado. Este sistema comienza en la estación central o subestación, hace un “ciclo” completo por el área a abastecer y regresa al punto de donde partió, lo cual permite que el área sea abastecida desde ambos extremos, teniendo una ventaja para aislar ciertas secciones en caso de alguna falla. En general estas redes operan con el anillo abierto y este se cierra en caso de una falla.

Las redes de distribución malladas están formadas por redes en anillo unidas en forma radial. Son redes muy complejas en donde la potencia de cortocircuito aumenta de forma drástica. En comparación con las configuraciones de anillo y radial, la red mallada tiene la configuración más compleja ya que incluye muchas conexiones alternativas entre nodos, lo cual aumenta la dificultad para lograr el correcto funcionamiento y protección de la red.

1.2 Redes eléctricas inteligentes

En los últimos años, las redes de distribución comenzaron a sufrir una transformación hacia el paradigma de Redes Eléctricas Inteligentes (REI) o también llamadas *Smart Grid* (SG). Estas redes son una conjunción de la red eléctrica convencional con tecnologías modernas

de la información y comunicación de datos (*Information and Communication Technologies* o ICTs por sus siglas en inglés) y tomas de decisiones, integrando sistemas de generación distribuida (*DER*), sistemas de almacenamiento y microrredes. Es decir, son redes eléctricas que pueden integrar de manera inteligente el comportamiento y las acciones de todos los usuarios conectados a ella como, por ejemplo, generadores y clientes, para entregar de manera eficiente suministros de electricidad sostenibles, económicos y seguros.

Algunos impulsores de esta transformación fueron factores políticos y/o ambientales como el aumento en la incorporación de fuentes de energía renovables que inyecten energía eléctrica a la red, una desregulación del mercado eléctrico y la inclusión de clientes como participantes activos en la gestión de la red.

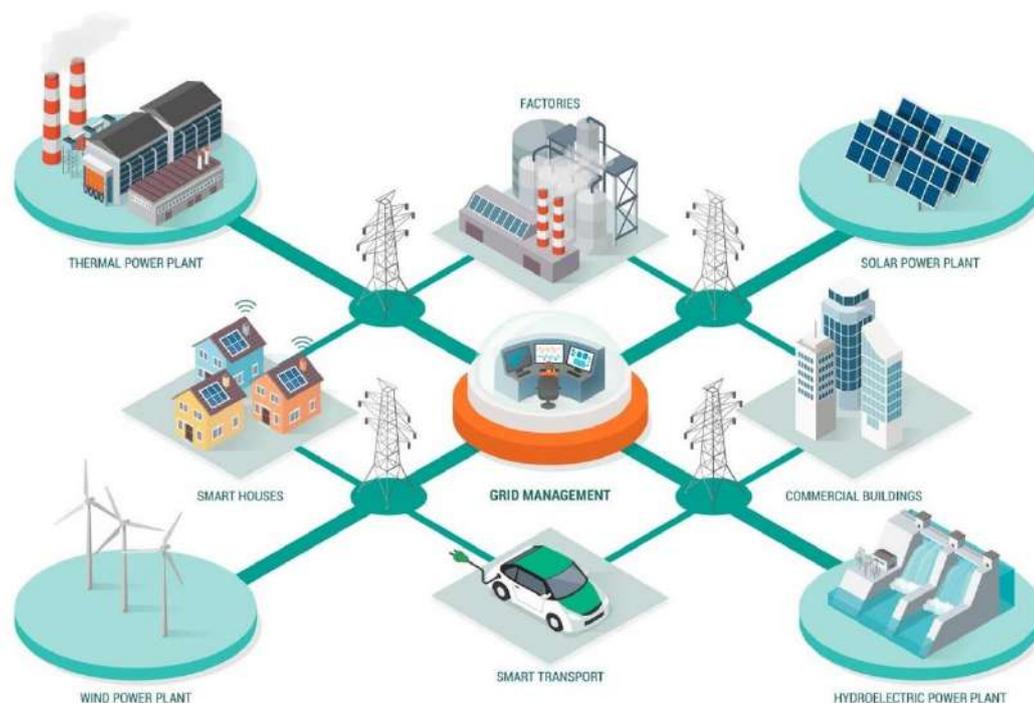


Figura 1-7. Esquema de una Red Eléctrica Inteligente

El agregado de recursos distribuidos a la red convencional da como resultado una arquitectura como se muestra en la Figura 1-8. Estas microrredes pueden aislarse del resto del sistema aprovechando la generación local, o bien mantenerse conectadas. La energía generada localmente puede dirigirse a cargas no críticas o venderse a la empresa de servicios públicos si así lo acuerda o lo permite la medición neta. Otra ventaja de las microrredes es que se pueden diseñar en función de los requisitos de los usuarios finales, una gran diferencia respecto al paradigma de generación central.

Hay varios desafíos técnicos que enfrentan las SG. Por ejemplo, la intermitencia de la generación de energía renovable que afecta la calidad de la electricidad, o las redes de gran escala conformadas por pequeños mecanismos de generación distribuida, como los paneles

fotovoltaicos (PV), baterías, energía eólica, etc. que aumentan en gran medida la complejidad del sistema.

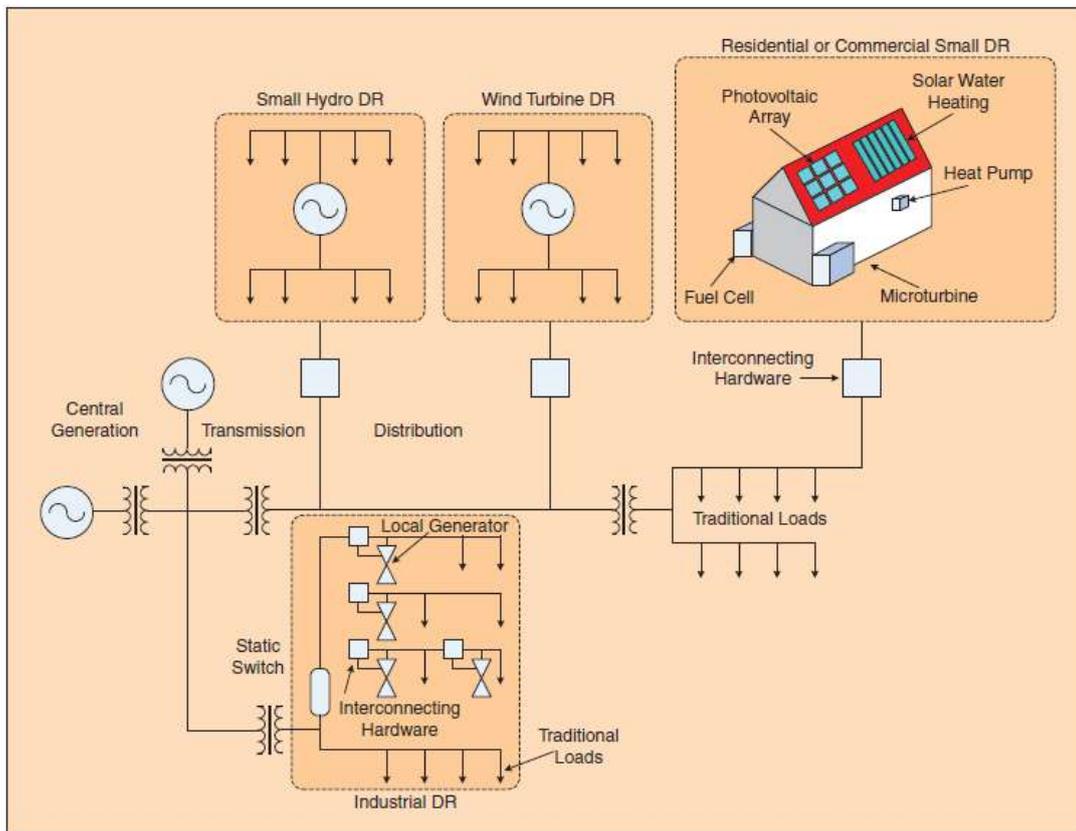


Figura 1-8. Arquitectura de una microrred. [17]

Las redes inteligentes como campo multidisciplinario presentan muchos desafíos y oportunidades para la investigación y el desarrollo de la electrónica industrial, que se relacionan con la aplicación de la electrónica y las ciencias eléctricas. Estas aplicaciones mejoran los procesos industriales y de fabricación, abordando los últimos desarrollos en electrónica de potencia, sistemas de control informático e inteligente, robótica, adquisición de datos y procesamiento de señales. [17]

Por tal motivo, hay un interés considerable por parte de las comunidades de investigación y proveedores para desarrollar herramientas y algoritmos innovadores que se adapten a las redes de distribución inteligentes y que proporcionen controles avanzados y automatización de la distribución. Un ejemplo de ello son las herramientas de Flujo de Energía óptimo (OPF por sus siglas en inglés *Optimal Power Flows*), optimización de Volt/Var o reconfiguraciones avanzadas de los alimentadores de la red, entre otros. Algunos de los objetivos comúnmente buscados en estas herramientas son la minimización de los costes totales del sistema, la minimización de pérdidas en la red, la maximización de la seguridad de suministro y la calidad de la energía (PQ por sus siglas en inglés *Power Quality*) [6]

1.3 Calidad de la Energía

Se entiende por calidad de la energía al estado en el cual la energía eléctrica es suministrada a los equipos y dispositivos con las características y condiciones adecuadas que les permita mantener su continuidad sin que se afecte su desempeño ni provoque fallas a sus componentes.[8]

Otra definición interesante de las muchas que se encuentran en la bibliografía es la utilizada en *Bollen y Gu* [4]: *“La calidad de la energía es la combinación de la calidad del voltaje y la calidad de la corriente. La calidad del voltaje se relaciona con las desviaciones del voltaje real del voltaje ideal, y análogo sucede con la corriente”*. Se define a la tensión ideal como una forma de onda de tensión sinusoidal con amplitud y frecuencia constantes y nominales, y a la corriente ideal a aquella cuya amplitud y frecuencia también son constantes y, además, están en fase con la tensión, Figura 1-9. Cualquier desviación de la tensión o de la corriente del ideal es una perturbación de PQ.

A menudo, no es posible distinguir entre una perturbación de tensión o de corriente debido a que cualquier cambio en la corriente da un cambio en la tensión y viceversa. Por este motivo, al hacer referencia a perturbaciones de tensión o corriente, se recurre a la causa como criterio para distinguir las: las perturbaciones de tensión se originan en la red eléctrica y afectan potencialmente a los clientes, mientras que las perturbaciones de corriente se originan en un cliente y potencialmente afectan a la red. Sin embargo, no siempre es posible distinguir el tipo de perturbación. Por ejemplo, el arranque de un motor de inducción grande conduce a una sobrecorriente. Visto desde la red esto es una perturbación de corriente, no obstante, la caída de tensión resultante es una perturbación de tensión para un cliente vecino. Esta dificultad de distinguir entre perturbaciones de tensión y corriente es una de las razones por las que generalmente se usa el término calidad de energía. El término calidad de tensión se reserva para los casos en que sólo se considera el voltaje en un lugar determinado y el término calidad de corriente se utiliza a veces para describir el rendimiento de los convertidores electrónicos de potencia conectados a la red eléctrica.

Las perturbaciones de la calidad de la energía pueden dividirse en dos grupos: las variaciones y los eventos. Las variaciones son perturbaciones de estado estacionario o casi estacionario que requieren (o permiten) mediciones continuas. Mientras que los eventos son perturbaciones repentinas con un principio y un final.

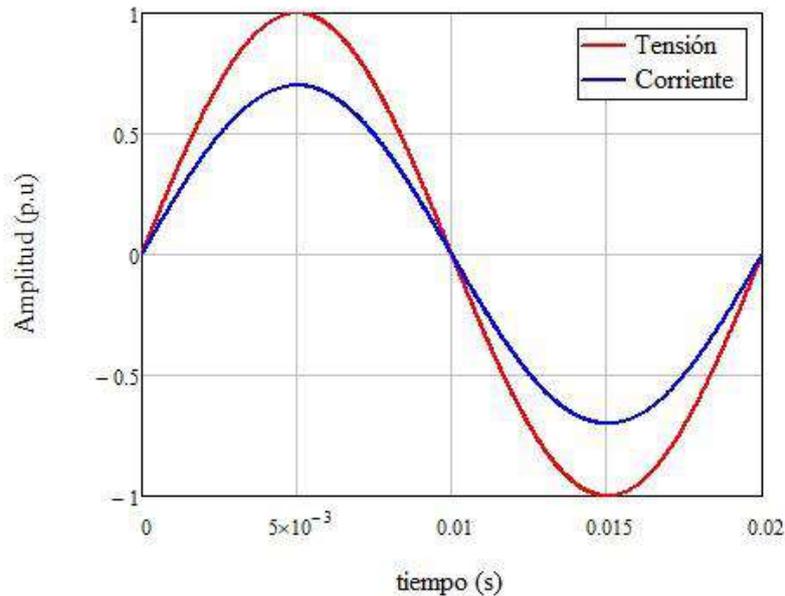


Figura 1-9. Formas de onda de tensión y corriente ideales, sin desfase y senoidales.

El interés creciente en la calidad de la energía tiene varias explicaciones, siendo las siguientes algunas de ellas:

- Los equipos tecnológicos se han vuelto más sensibles a las perturbaciones de tensión. Esto se debe a que la mayoría están basados en microprocesadores y electrónica de potencia.
- Los equipos tecnológicos causan perturbaciones de tensión. Los equipos de electrónica modernos no solo son sensibles a las perturbaciones de tensión, sino que también las causan, generando perturbaciones para otros consumidores de la red.
- Una creciente necesidad de estandarización y criterios de desempeño. En el pasado el consumidor de energía eléctrica era visto simplemente como una “carga” de la red, sin embargo, esa visión fue cambiando con el correr de los años. Hoy en día la electricidad se ve como un producto con ciertas características que los consumidores esperan que sean cumplidas y que puede ser medido, garantizado, mejorado, predicho, etc. Por este motivo, las empresas de servicios públicos quieren entregar un buen producto y, a su vez, los usuarios finales así lo exigen.
- La calidad de la energía puede ser medida. La creciente disponibilidad de equipos electrónicos que pueden medir las formas de onda de la red contribuye al interés en la calidad energética. En el pasado, las mediciones eléctricas estaban acotadas a valores RMS (valor eficaz, del inglés *Root Mean Square*), frecuencia y la medición de largas interrupciones, siendo difícil detectar eventos como caídas de tensión o armónicos en la red.

El estudio de la PQ es un tema importante tanto para las empresas eléctricas como para los usuarios, ya que permite identificar posibles problemas y adoptar las soluciones requeridas en cada caso. Con esto no solo se reducen gastos en términos de energía malgastada y tiempos de inactividad innecesarios, sino que también se reducen los riesgos de fallas y peligros. Para entender cómo se mide la calidad de la energía es necesario definir los siguientes conceptos: el concepto de calidad del servicio técnico prestado y el concepto de calidad de producto.

La continuidad del suministro eléctrico o calidad del servicio técnico prestado, involucra la frecuencia y duración media de las interrupciones en el suministro. Se evalúa a partir de datos estadísticos como la frecuencia de interrupciones (cantidad de veces en un periodo determinado que se interrumpe el suministro a un cliente) y la duración de cada una de ellas (tiempo total sin suministro para cada interrupción). Existen normativas que establecen los valores máximos admitidos para cada indicador y, en caso de excederse de dichos valores, establecen qué sanciones deben aplicarse.

La calidad de producto técnico se relaciona con el nivel de tensión en el punto de suministro y sus perturbaciones asociadas (variaciones rápidas, caídas lentas de tensión y distorsión armónica). Se aplican las respectivas sanciones en función del tiempo sobre el cual se excedieron los niveles exigidos.

Cuando se analiza la calidad de la energía se estudian perturbaciones como transitorios, que a su vez pueden dividirse en: transitorios de tipo impulsivos y oscilatorios, distorsiones en la forma de onda, fluctuaciones de tensión (*flicker*), variaciones de frecuencia (desviación de la frecuencia fundamental), *dips* y *swells* de tensión o interrupciones que, en base al rango de duración se distinguen las instantáneas, momentánea, temporal y sostenida.

En este trabajo se profundizará en los eventos de tensión, que engloba a los *dips* y *swells* de tensión, e interrupciones de corta duración ya que, como se verá en los capítulos posteriores, se simularán fallas en una red eléctrica normalizada y se analizará el desempeño del algoritmo de detección ASA desarrollado por Strack [10], el cual detecta y clasifica automáticamente el tipo de evento medido.

1.3.1 Eventos de tensión

En la norma IEEE 1159:2009 [11] los *dips* (o huecos de tensión), *swells* (o sobretensión) e interrupciones se clasifican como variaciones de corta duración que pueden ser instantáneas (0.5 a 30 ciclos), momentáneas (30 ciclos a 3 segundos) y temporales (3 segundos a 1 minuto).

Un *dip* de tensión es una reducción del valor RMS de la tensión de la alimentación del sistema eléctrico durante un breve periodo de tiempo. Se describe de acuerdo a la duración y su tensión remanente más baja medida durante el evento, expresada en un valor porcentual de la tensión RMS nominal, entre el 10% y 90% respecto a la nominal. También puede emplearse el término de profundidad del *dip*, y en este caso se está haciendo referencia al porcentaje de caída de tensión medida desde el valor nominal. Es decir, una profundidad del *dip* del 20% es análogo a decir que la tensión remanente es del 80%.

Un *swell* de tensión es un aumento del valor RMS de la tensión de la alimentación del sistema eléctrico durante un breve periodo de tiempo. Se describe de acuerdo a la duración y su tensión remanente más alta medida durante el evento, expresada en un valor porcentual de la tensión RMS nominal, por encima del 110% respecto a la nominal.

Una interrupción de tensión es una reducción por debajo del 10 % del valor RMS nominal, dejando sin servicio a los consumidores afectados.

Las consecuencias de estos eventos en la red son analizadas posteriormente.

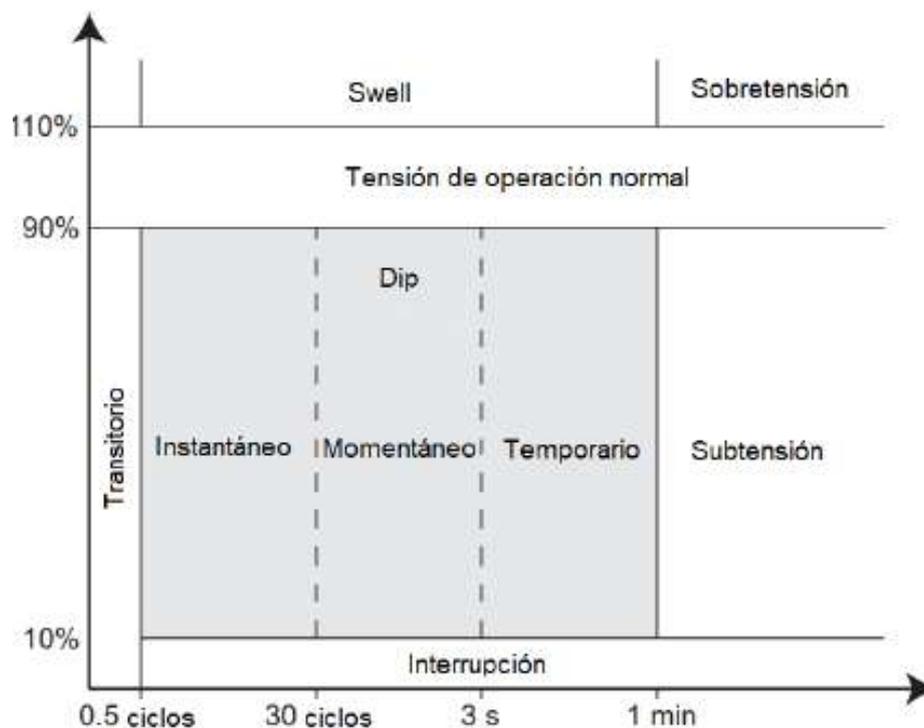


Figura 1-10. Definición de dips, swells e interrupción de acuerdo a la norma IEEE 1159-2009. [11]

En lo que respecta a los *dips* de tensión, los mismos resultan de aumentos de corta duración en la magnitud de la corriente. Como se verá posteriormente, este aumento en la corriente puede darse en zonas diferentes al lugar en donde se está experimentando un *dip* de tensión, debido a que el evento se propaga por la red.

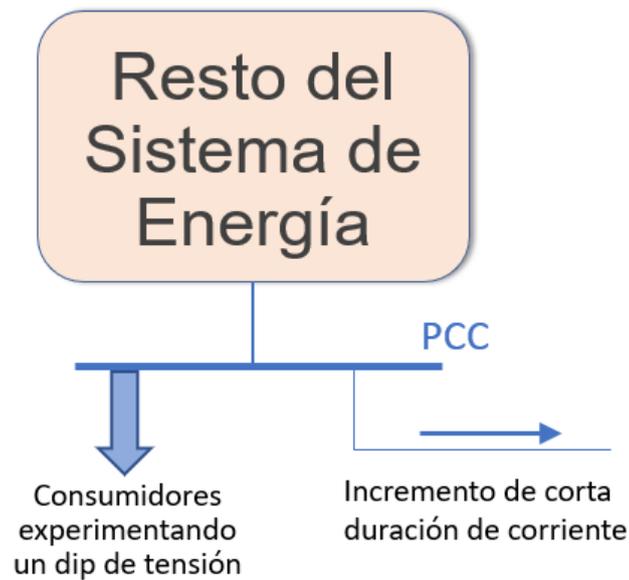


Figura 1-11. Origen de un *dip* de tensión.

El aumento de corta duración en la corriente proporciona hueco de tensión en el bus “PCC”, conocido como el punto de acoplamiento común (entre el cliente y la falla). Todos los clientes que son suministrados desde el PCC indicado experimentarán una reducción en la tensión (*dip* de tensión). Hay muchas causas que generan un aumento temporal en la corriente, pero las tres causas principales son:

1. Fallas en la red: Las fallas pueden dividirse en fallas monofásicas (falla entre una fase y tierra, siendo estadísticamente las más comunes y más severas), bifásica (falla entre dos fases o dos fases y tierra), y en el segundo grupo las fallas trifásicas (falla entre las tres fases). En *Bollen y Gu* [4] se realiza un análisis de fallas simétricas y asimétricas en sistemas de transmisión y en sistemas de distribución. Se desarrollará brevemente el caso de una falla asimétrica en un sistema de transmisión y la propagación del mismo a otros niveles de tensión en la red.

En la Figura 1-12 se muestra una caída de voltaje medida a 11 kV RMS debido a una falla no simétrica en el nivel de transmisión. Se muestran las formas de onda y sus valores RMS para las 3 fases. Se observa que una fase sufrió una caída significativa de tensión, mientras que las otras dos muestran una caída menor. La caída en las otras dos fases es muy similar, comportamiento típico en fallas no simétricas en el nivel de transmisión, pero no así en fallas del sistema de distribución. En el gráfico de valores RMS también puede observarse la recuperación de la red en dos etapas (2 escalones bien definidos).

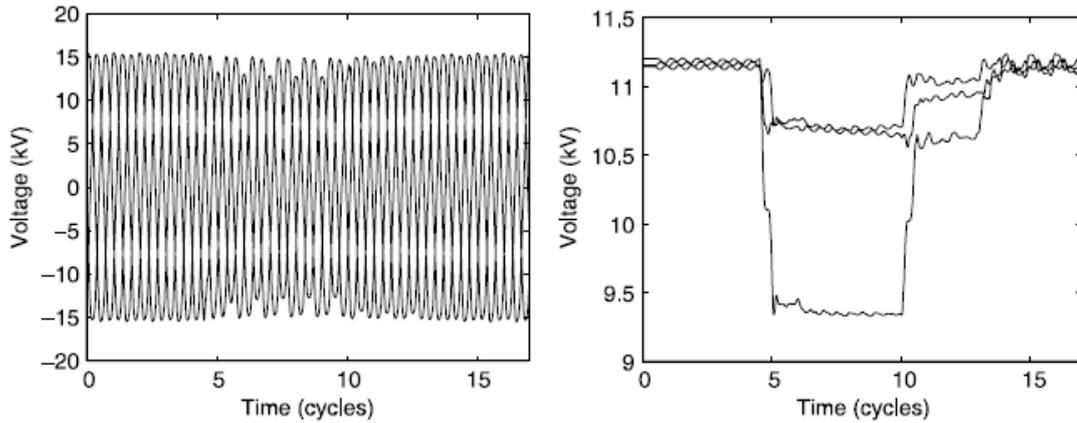


Figura 1-12. Hueco de tensión debido a una falla asimétrica en una red de transmisión. A la izquierda la forma de onda, y a la derecha los valores RMS de las 3 fases. [4]

Un comportamiento interesante para analizar es la propagación por la red de estos eventos, ya que los valores de tensión no permanecen iguales al propagarse. Para ejemplificar este fenómeno se muestran las caídas de tensión (sus valores RMS) debido a una falla monofásica a un nivel de tensión de 400 kV que fue registrada en tres niveles de tensión: 132 kV, 10 kV y 400 V.

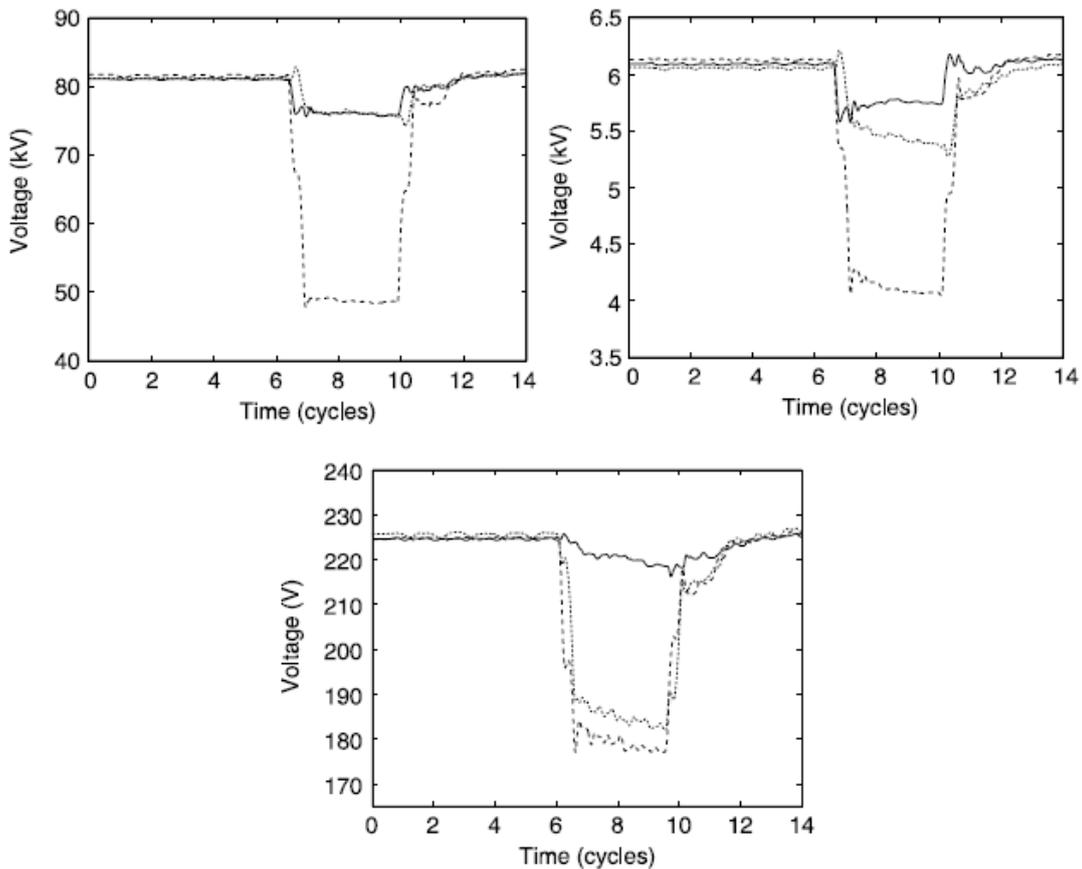


Figura 1-13. Hueco de tensión provocado por una falla monofásica a 400 kV: medido a 132 kV (izquierda), a 10 kV (derecha) y a 400 V (abajo). [4]

La caída de voltaje a 132 kV muestra una gran caída en la magnitud de voltaje en una fase, junto con una pequeña caída en las otras dos fases y que resultan ser casi idénticas (como se verá en el Capítulo 3, este evento de tensión corresponde a un Tipo D). La caída de voltaje medida a 10 kV también muestra una gran caída en una fase y una caída menor en las otras dos fases. Sin embargo, hay dos diferencias en comparación con la medición a 132 kV. La primera diferencia es que los voltajes RMS ya no son constantes durante la falla. La segunda diferencia es que el voltaje RMS en las dos fases menos afectadas ya no son los mismos. Ambos efectos se deben al hecho de que las corrientes de carga varían durante la falla. La caída de voltaje a 400 V es la que presenta mayor diferencia: muestra una gran caída de tensión en dos fases y muy poca en la restante (como se desarrollará en el Capítulo 3, este evento de tensión corresponde a un Tipo C). La diferencia entre la medición de la falla en 10 kV y 400 V se debe al transformador DY entre estos niveles de voltaje. En los eventos de tensión además de analizarse los valores de tensión RMS, se analizan los saltos de fases.

Este análisis en conjunto permite clasificar al tipo de evento más probable ocurrido, es decir se analiza la “firma” del evento sobre la red. En estos eventos también se analiza el tiempo de duración del *dip* de tensión, el cual depende principalmente del tiempo que tarden en actuar los elementos de protección provocando la recuperación de la red.

En lo que respecta a las causas de estas fallas, las más frecuentes suelen ser los contactos accidentales con vegetación o animales, envejecimiento de aisladores, caídas de líneas por fuertes vientos, sabotajes, etc.

2. Arranque de motores de inducción: El arranque del motor eléctrico provoca una repentina caída de tensión simultánea en las tres fases, o hueco de tensión equilibrado, seguido de una recuperación lenta de la red. Esto se debe a que, durante el arranque directo, la corriente consumida aumenta entre cuatro y doce veces respecto a la nominal, y puede mantenerse por varios segundos. Estos valores dependen de la carga mecánica del motor en el momento del arranque, siendo el peor caso un arranque pesado. Estos valores elevados de corriente provocan grandes pérdidas en la línea, y por ende una disminución en la tensión en la red de suministro y en la instalación. Normas como la AEA 90364 [20] exigen que el dimensionamiento de los cables que alimentan estos motores sea tal que las caídas de tensión se mantengan por debajo de un 15%, medido desde la acometida de la instalación.

3. Energización de transformadores: La energización de un transformador también provoca *dip* de tensión repentino seguido de una recuperación lenta del sistema, similar al caso del arranque del motor eléctrico. Sin embargo, la caída es diferente en las tres fases existiendo además una gran distorsión armónica uniforme.

En la Figura 1-14 se muestra un ejemplo de una caída de tensión debido a la energización de un transformador sin carga.

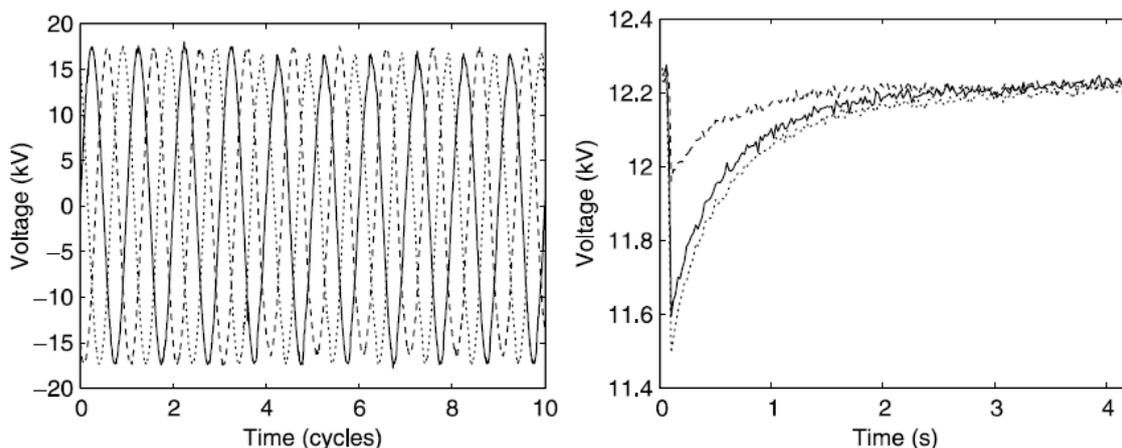


Figura 1-14. Hueco de tensión debido a la energización de un transformador: a la izquierda se presentan las formas de onda de tensión y a la derecha los valores de tensiones RMS. [4]

Al energizar el transformador, aparece una corriente de irrupción (*inrush*) elevada que decae lentamente, y que es la causante del *dip* de tensión. El desequilibrio entre las fases es claramente visible, como así también la recuperación de la tensión en forma exponencial determinada por la constante de tiempo con la que decae la compensación de flujo en el núcleo del transformador. Los *dip* de tensión también aparecen cuando se energizan los transformadores cargados. En este caso la corriente de irrupción se ve disminuida si se la compara con el caso del transformador en vacío y también se ve reducida la constante de tiempo con la que la corriente decae. La energización de transformadores cargados ocurre cuando se vuelve a energizar la red de distribución luego de una interrupción.

Otras causas de eventos de tensión son las maniobras con grandes cargas, descargas de origen atmosférico, o una combinación de las nombradas previamente. La severidad del *dip* de tensión a raíz de las causas enumeradas previamente depende de la puesta a tierra del sistema, la impedancia y localización de la falla, las conexiones de los transformadores, la forma de actuación de las protecciones y el modo de conmutación de la carga y el tipo de carga que se conecta o desconecta.

En lo que respecta a las consecuencias [9], para el caso de motores eléctricos, una tensión menor a la nominal disminuye el par en la máquina (el par es proporcional al cuadrado de la tensión). Una vez que el hueco de tensión desaparece, la tensión en los bornes del motor aumentará al nominal, generando un aumento en la corriente consumida, similar a lo que sucede en el arranque del motor. El aumento de la corriente consumida genera calentamiento y esfuerzos electrodinámicos en los arrollamientos de la máquina, pudiendo disminuir la vida útil por un deterioro en el aislamiento. Esto también afecta al aparataje del sistema como los contactores y también a los interruptores automáticos cuyas bobinas de accionamiento funcionan en un rango determinado de tensión.

Los huecos de tensión también llamaron la atención en el estudio de grandes parques eólicos y en la generación distribuida. La pérdida repentina de una gran cantidad de generación puede causar problemas de estabilidad en el sistema.[9]

En caso de haber *swells* de tensión, algunas de las consecuencias pueden ser la disminución de la vida útil de algunos dispositivos, como por ejemplo las lámparas incandescentes y fluorescentes, el aumento del riesgo de fallas en el aislamiento de los equipos, aumento del torque y las corrientes de encendido de motores, la disipación de calor en resistencias y la corriente en transformadores, lo cual resulta en una mayor distorsión en la forma de onda de tensión.

1.3.2 Proceso de medición

Para detectar los eventos de tensión debe existir una metodología en la cual se lleva a cabo la medición de la calidad de la energía. La misma consiste en:

Definir características de las variaciones. La elección de las mismas está relacionada a la esencia de la variación. Por ejemplo, las variaciones de tensión están asociadas a la variación de magnitud de la forma de onda de tensión. Para cuantificar la severidad se miden valores absolutos, valores RMS o valores pico, tanto de la tensión como de la corriente. También se debe precisar el método de medición y sus parámetros asociados, como la frecuencia de muestreo, la longitud de la ventana sobre la cual es extraída la medición, la frecuencia de repetición de la medición, y el modo de procesar la serie de valores. La elección de los métodos de medición puede influir sobre el valor resultante, por lo cual es un factor decisivo para cumplir con una norma.

Distinguir entre una variación y un evento. Para esta tarea se necesita un mecanismo de trigger. El método más comúnmente utilizado compara un valor RMS de ventana deslizante con un valor umbral. Esto requiere la definición de un número de valores, tal como el tamaño de la ventana, el solapamiento entre ventanas sucesivas, y la elección del umbral, aunque por supuesto existen también otros métodos apropiados.

Caracterizar cada evento a través de parámetros. Una vez que el evento es detectado, por ejemplo, un *dip* de tensión, el mismo se caracteriza por medio de parámetros como la duración, magnitud, fase de inicio, salto de fase, etc.

Clasificar cada evento de acuerdo a su causa principal. Frecuentemente, este es el objetivo final del análisis. Un tema esencial es la elección del criterio y del método de clasificación. En general, los criterios de clasificación se basan en un evento de tensión idealizado, donde las impedancias de red son modelizadas como resistencias, no contemplando las inductancias de los generadores, transformadores y líneas. Otras simplificaciones como no contemplar máquinas rotantes, cargas desbalanceadas, entre otros,

generan que el evento simulado diste en gran medida con la realidad. Se enumerarán las perturbaciones en los eventos de tensión responsables de clasificaciones erróneas. El desarrollo de cada una de estas perturbaciones fue realizado en *Strack* [10].

- Saltos de fase debidos a la falla (SFDF): cambio en el ángulo de fase producido durante el evento únicamente en las tensiones de las líneas afectadas. Su principal causa es el cambio brusco de impedancia en las líneas afectadas debido a la aparición de una impedancia de falla, frecuentemente de carácter resistiva.
- Saltos de fase debidos a la red (SFDR): cambio en el ángulo de fase que se debe principalmente a la diferencia en la relación X/R (reactancia/resistencia) de las impedancias comprendidas entre el PCC y la falla (Z_f) y entre el PCC y la fuente (Z_s)
- Rotaciones de fase asimétricas (RFA): se origina cuando las impedancias de secuencia positiva y negativa entre el PCC y la fuente son distintas. Esto ocurre cuando la falla que da lugar al evento se origina en cercanías de máquinas rotantes que, a diferencia de los cables y transformadores, presentan impedancias de secuencia positiva y negativa de magnitudes significativamente diferentes.
- Rotaciones de fase simétricas (RFS): cambio de fase de igual argumento en las tres tensiones simples, cuyo origen puede encontrarse en los cambios bruscos de grandes cargas balanceadas muy inductivas. Un claro ejemplo es el de los grandes motores trifásicos de inducción, que en el momento del arranque presentan el rotor bloqueado, por lo cual su impedancia es altamente inductiva y su corriente alcanza valores de hasta 7 veces la nominal (caso de arranque directo)
- Desviaciones de la tensión de prefalla respecto de su valor nominal (DTP): La carga puede modificar la magnitud de la tensión de prefalla generando una desviación de la tensión de prefalla respecto de su valor nominal. A su vez, en una red eléctrica la magnitud de la tensión en cada punto de la misma depende no solo de la carga, sino de otros factores, como el nivel de *taps* de regulación con los que cuentan los transformadores, el paso en ese instante de los reguladores de tensión, la cantidad de energía reactiva inyectada a la red aportada por los bancos de capacitores utilizados para corregir el factor de potencia, etc.

Para ejemplificar cómo las perturbaciones afectan a la clasificación de los eventos de tensión, se muestra en la Figura 1-15 como dos eventos de tensión diferentes, uno tipo C y el otro tipo D pueden generar la misma firma en el sitio de medición al estar presente saltos y rotaciones de fases:

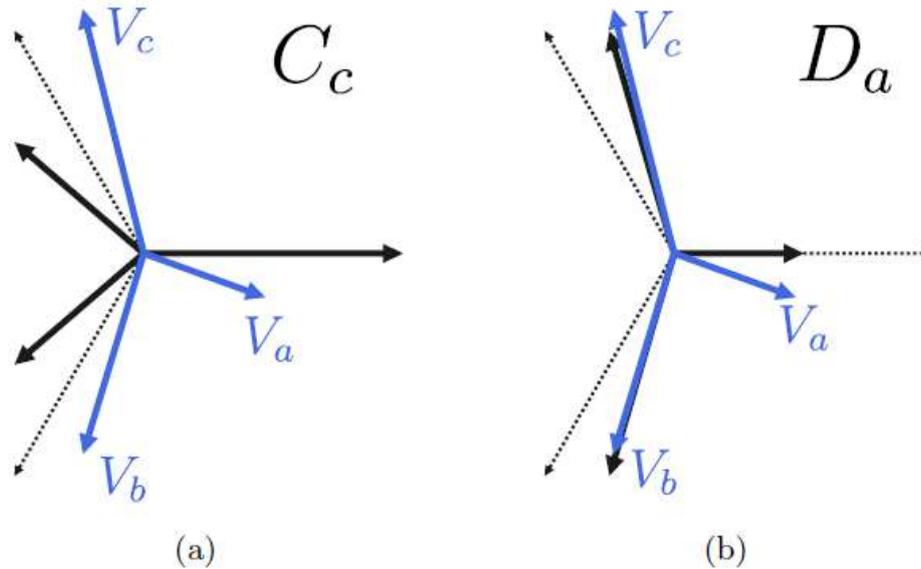


Figura 1-15. Evento tipo Cc (a) y tipo Da (b). En línea punteada se representa la terna ideal. En línea llena negra se representa el evento ideal y en línea azul se representan las tensiones resultantes al incluir saltos y rotaciones de fase durante la ocurrencia de los eventos. [10]

La clasificación de eventos de tensión será desarrollada en profundidad en el Capítulo 3 del presente trabajo.

Como se comentó previamente, muchas veces el desarrollo y desempeño de los algoritmos suele realizarse bajo simulaciones computacionales de escenarios idealizados. En el presente trabajo, esta afirmación referencia a los algoritmos de detección de eventos, pero puede extenderse a la gran cantidad de algoritmos aplicados en otras temáticas.

También es cierto que, con los avances tecnológicos, las simulaciones computacionales han ido ganando mayor importancia en la Ingeniería, siendo practicable incluso en computadoras de escritorio o, para los casos más complejos, en clústeres computacionales para acelerar los cálculos numéricos. Sin embargo, esto no siempre significa que los resultados obtenidos sean representativos de la realidad ya que pueden introducirse errores propios de los métodos numéricos utilizados, errores por las simplificaciones realizadas, errores en el modelizado del fenómeno por falta de información o en algunos casos por simular el fenómeno en un escenario que dista de la realidad. Para el caso de simulaciones de fenómenos eléctricos, surge la necesidad de tener disponibles redes eléctricas validadas sobre las que se aplicarán los algoritmos con el fin de detectar falencias en los mismos y así poder mejorarlos. De esta manera, los resultados de las simulaciones se contrastan con los resultados de ensayos reales y, en conjunto, se obtienen desarrollos tecnológicos exitosos.

En este punto es necesario la explicación de los *Node Test Feeder* o Alimentadores de prueba.

1.4 Alimentadores de prueba

Un alimentador de prueba es un modelo de red de distribución que puede replicar el comportamiento de un alimentador de distribución real. El propósito de estos alimentadores de prueba es proporcionar modelos de sistemas de distribución que reflejen la amplia diversidad en el diseño y sus diversos desafíos analíticos. Debido a su utilidad y accesibilidad, los alimentadores de prueba se han utilizado para una amplia gama de investigaciones. [7]

Los alimentadores de prueba de distribución se pueden construir utilizando diferentes métodos, como se ilustra en la Figura 1-16. El más sencillo es el primer método, que consiste en seleccionar un alimentador real de una red de distribución real y anonimizarlo para eliminar datos privados o confidenciales.

El resto de los métodos construyen redes sintéticas utilizando redes reales como base: el segundo método utiliza técnicas de agrupamiento donde se agrupan varias redes reales y luego unen para construir un alimentador de prueba representativo sintético para cada grupo de redes reales. El tercer método construye un alimentador de prueba a través del diseño manual, permitiendo el modelizado de características específicas de la red, pero solo es posible con redes de pequeña escala debido a su complejidad. El cuarto método consiste en construir redes de prueba sintéticas mediante el uso de herramientas de planificación que están diseñadas para crear redes realistas utilizando los mismos criterios técnicos y económicos que utilizan los planificadores de distribución. Esta opción permite el diseño de nuevas redes de prueba desde cero con características hechas a la medida.

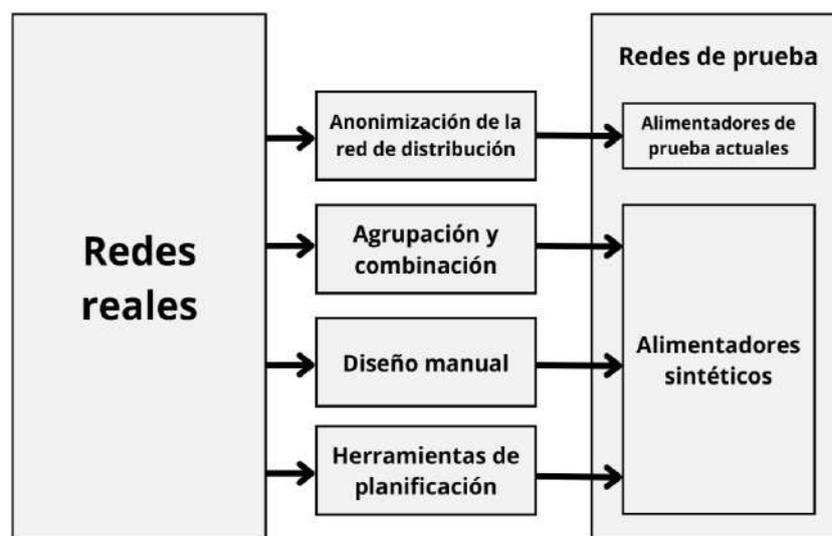


Figura 1-16. Métodos de construcción de alimentadores de prueba

Los alimentadores de prueba obtenidos deben ser representativos de un conjunto de redes de distribución reales con características específicas. La representatividad es un criterio de

validación y por lo tanto no está rígidamente ligada a ninguno de los métodos de diseño, y se puede obtener con cualquiera de los cuatro procedimientos descritos. [6]

El *TFWG* (Grupo de Trabajo de Alimentadores de Prueba, por sus siglas en inglés.), han desarrollado varios alimentadores de prueba en las últimas décadas. Algunos de estos alimentadores se desarrollaron con fines académicos, mientras que otros se desarrollaron para necesidades más industriales. Los alimentadores desarrollados con fines académicos se crearon para probar la gran cantidad de nuevos métodos de solución de flujo de potencia que se proponían en ese momento. Los alimentadores centrados en la industria se desarrollaron para resaltar aspectos de la construcción o el funcionamiento del sistema que no eran muy conocidos por la comunidad académica.

Las redes de distribución son consideradas una infraestructura crítica de un país, y los datos detallados de carga/cliente generan preocupaciones sobre la privacidad. Como resultado, solo unas pocas redes reales están disponibles públicamente para su uso como red de prueba. Esto ha obligado a la comunidad de investigadores a utilizar solo los alimentadores de prueba disponibles que se han publicado en la literatura hasta el momento.

En 1991, el *TFWG* lanzó el primer conjunto de alimentadores de prueba disponibles abiertamente. Este conjunto original de cinco modelos estaba destinado a proporcionar a los investigadores modelos que incluían cargas desequilibradas y sistemas de distribución no transpuestos con el fin de probar nuevos métodos de solución de flujo de potencia. Desde entonces, el *TFWG* ha publicado una serie de modelos adicionales que han sido diseñados para examinar desafíos computacionales específicos asociados con los sistemas de distribución y para reflejar los cambios en la forma en que se operan. A pesar de esto, los investigadores han utilizado rutinariamente estos modelos en investigaciones para las que no estaban destinados originalmente debido principalmente a la escasa cantidad de alimentadores de prueba. Si bien este uso ampliado de los alimentadores de prueba destaca sus contribuciones a la industria y es alentado por el *TFWG*, los investigadores no siempre han sido conscientes de que muchos de los alimentadores de prueba existentes no se desarrollaron para representar un sistema de distribución "típico", sino que fueron diseñados para probar la capacidad de nuevos algoritmos. Por ejemplo, el alimentador de prueba de 13 nodos IEEE se diseñó originalmente para probar la capacidad de los solucionadores de flujo de potencia para manejar sistemas altamente desequilibrados, y no pretende representar un circuito de distribución de "tamaño completo". Hoy en día, el número de alimentadores de prueba publicados sigue siendo limitado, a pesar de los esfuerzos significativos para aumentar la disponibilidad de los mismos.

Los desafíos en el desarrollo de modelos de alimentadores radican en la existencia de diferencias regionales en la estructura y los circuitos de operación en todo el mundo. La mayoría de los sistemas de energía modernos se construyeron a lo largo del siglo XX, pero a

pesar de su desarrollo paralelo, existen diferencias fundamentales en la forma en que evolucionaron. Por ejemplo, un cliente residencial típico en América del Norte recibe energía monofásica a 120 V/240 V a 60 Hz, mientras que un cliente residencial típico en Europa recibe energía trifásica a 400 V/240 V y 50 Hz. Se pueden encontrar diferencias sustanciales en el diseño y operación de los alimentadores dentro de un solo país, e incluso dentro de una sola empresa de servicios públicos. [7]

1.4.1 Modelos de alimentadores de prueba

Los primeros modelos desarrollados por el *TFWG* representaban el tipo de sistemas radiales desequilibrados que es común en América del Norte. Debido a que estos alimentadores fueron los primeros que se lanzaron, eran de tamaño relativamente pequeño y se centraban en representar las características del sistema desequilibrado. Como se dijo anteriormente, estos alimentadores fueron diseñados para probar algoritmos de flujo de potencia, que más tarde se utilizarían para probar la capacidad de modelar una amplia gama de elementos del sistema de distribución, especialmente transformadores. En la Tabla 1-3 se resumen las características principales de cada uno de ellos.

Tabla 1-3. Características principales de alimentadores de prueba publicados por IEEE [6]

ID	Longitud (km)	Voltaje primario (kV)	Número de clientes/cargas	Carga pico (MVA)	DG (MVA)
13 nodos	2.5	4.16	9	3.6	0
123 nodos	12	4.16	114	3.8	0
34 nodos	94	24.9	24	1.6	0
37 nodos	5.5	4.8	25	2.73	0
4 nodos	1.3	12.47	1	6.3	0
NEV	1.82	12.47	1	8.9	0
8500 nodos	170	12.47	1177	11.1	0
CRF	81.7	12.47/24.9	36	4.17	0.15
342 nodos	15.2	13.2	624	49.4	0

En el repositorio de *IEEE Test Feeder* [21] se encuentran disponibles estos alimentadores de prueba y también otros que no fueron desarrollados en el presente trabajo.

En las siguientes secciones se realizará un análisis más detallado de cada uno de estos alimentadores de prueba. [6] [12]

1.4.1.1 Alimentador de prueba de 4 nodos

Este alimentador no formaba parte del conjunto original de sistemas de prueba publicado en 1991. El propósito principal de este alimentador de prueba es proporcionar un sistema simple para probar todas las conexiones de transformadores trifásicos posibles. Las características del alimentador son:

1. Posee dos segmentos de línea con un banco de transformadores trifásico conectado entre los dos segmentos.
2. Los datos se especifican para conexiones de transformadores trifásicos "cerrados" y para conexiones "abiertas".
3. Los datos del transformador se especifican para pruebas de tomas *step-up* y *step-down*. El voltaje primario es de 12,47 kV, mientras que el voltaje secundario puede ser de 4,16 kV o 24,9 kV.
4. Los valores se especifican para carga equilibrada y desequilibrada en el nodo más remoto.

Los resultados de las pruebas para este alimentador incluyen las siguientes conexiones de los transformadores para operaciones *step-up* y *step-down* y para carga equilibrada y desequilibrada.

- i. Estrella con puesta a tierra – Estrella con puesta a tierra
- ii. Estrella con conexión a tierra – Triángulo
- iii. Estrella sin conexión a tierra – Triángulo
- iv. Triángulo - Estrella conectada a tierra
- v. Triángulo - Triángulo
- vi. Estrella abierta - Triángulo abierto

1.4.1.2 Alimentador de prueba de 13 nodos

Este alimentador es muy pequeño y, sin embargo, muestra algunas características muy interesantes.

1. Tiene un nivel de carga elevado si se lo compara con alimentadores típicos de 4,16 kV.
2. Tiene un regulador de voltaje de subestación que consta de tres unidades monofásicas conectadas en estrella.
3. Posee líneas aéreas y subterráneas con variedad de fases.
4. Cuenta con bancos de capacitores en derivación.
5. Posee un transformador en línea.
6. Presenta cargas desequilibradas puntuales y distribuidas.

Proporciona una buena prueba de la convergencia de un programa en el caso de un sistema con cargas desequilibradas.

1.4.1.3 Alimentador de prueba de 34 nodos

Este alimentador es un alimentador real ubicado en Arizona. La tensión nominal del alimentador es de 24,9 kV. Algunas de sus características principales son:

1. Es una red muy extensa y está ligeramente cargada si se la compara con alimentadores típicos de 24,9 kV.
2. Se requieren dos reguladores de tensión en línea para mantener un buen perfil de voltaje que satisfaga el estándar de la norma ANSI C84.1.
3. Cuenta con un transformador en línea que reduce el voltaje a 4,16 kV para una sección corta del alimentador.
4. La carga es desequilibrada, con cargas "puntuales" y "distribuidas". Se supone que las cargas distribuidas están conectadas en el centro del segmento de línea. Las cargas son del tipo PQ (potencia constante), corriente constante e impedancia constante.
5. Posee capacitores de derivación.

Debido a la longitud del alimentador y la carga desequilibrada, permite estudiar los problemas de convergencia de flujo de potencia. [6]

1.4.1.4 Alimentador de prueba de 37 nodos

Este alimentador es un alimentador real ubicado en California. Las características del alimentador son:

1. Cuenta con conexión trifásica del transformador en Dd que funciona a una tensión nominal de 4,8 kV.
2. Todos los tramos de línea son subterráneos.
3. Tiene un regulador de tensión de subestación compuesto por dos unidades monofásicas conectadas en triángulo abierto.
4. Todas las cargas son cargas "puntuales" y son del tipo PQ (potencia constante), corriente constante e impedancia constante.
5. La carga es muy desequilibrada.

1.4.1.5 Alimentador de prueba de 123 nodos

El alimentador de prueba de 123 nodos de IEEE opera a un voltaje nominal de 4.16 kV, es un sistema de tamaño mediano que consta de múltiples reguladores de voltaje y capacitores de derivación para solventar los problemas de caída de voltaje. Hay suficientes interruptores en el alimentador para que se puedan probar los procedimientos de configuración "óptimos".

Este es el alimentador más completo y se caracteriza por:

1. Tramos de líneas aéreas y subterráneas con varias fases.

2. Carga desequilibrada con todas las combinaciones de tipos de carga, PQ (potencia constante), I constante, Z constante.
3. Todas las cargas son "cargas puntuales" ubicadas en cada uno de los nodos.
4. Reguladores de voltaje de cuatro pasos.
5. Bancos de capacitores de derivación.
6. Conmutación para proporcionar rutas alternativas de flujo de potencia.

Este alimentador no tiene problemas de convergencia. Los cuatro reguladores de voltaje que posee proporcionan una buena prueba para asegurar que el cambio de las tomas de los reguladores individuales esté coordinado.

1.4.1.6 Alimentador de prueba de voltaje a neutro (NEV)

La proliferación de armónicos que generan las cargas residenciales está produciendo un aumento de los NEVs. Hoy en día, el modelado de NEV (voltaje a neutro) se está convirtiendo en un tema importante, sin embargo, requiere redes de prueba más detalladas. Este alimentador de prueba incluye explícitamente las resistencias de tierra de los polos y tiene como objetivo comprender el impacto en los niveles de voltaje de contacto y NEV bajo diferentes parámetros de sección del conductor neutro o resistencia de tierra. Se usa principalmente en estudios de modelado de carga y análisis de armónicos.

1.4.1.7 Alimentador de prueba de 8500 nodos

Este alimentador de prueba se obtuvo de una red de distribución real de EE.UU. con algunos cambios leves. El objetivo de este alimentador es proporcionar una red de referencia a gran escala que permita a los investigadores probar algoritmos en un sistema más realista y de mayor escala. Las coordenadas relativas del bus se incluyen en los datos del alimentador. Comprende varias características comunes de América del Norte, como una penetración profunda de los reguladores de voltaje, bancos de capacitores por fase, fases secundarias de bajo voltaje y transformadores de derivación centrada. Se publicó en dos versiones diferentes, una versión de carga secundaria equilibrada y otra más realista con cargas secundarias desequilibradas. El nivel de voltaje primario es de 12,47 kV y los secundarios se dividen entre 120 y 240 V. Este alimentador de prueba se usa ampliamente en estudios de integración de DER y redes inteligentes.

1.4.1.8 Alimentador de prueba de distribución Integral (CTF)

Este alimentador de prueba tiene como objetivo presentar una red diversa y detallada donde se incluyen la mayoría de las configuraciones y equipos disponibles. La inclusión de dispositivos de conmutación permite una variedad de posibilidades de reconfiguración de la

red. Esta capacidad, combinada con varios tipos de equipos diferentes como líneas aéreas y subterráneas, transformadores disímiles, reguladores de voltaje de paso, máquinas de inducción, bancos de capacitores y la combinación de cargas desequilibradas distribuidas y puntuales, da como resultado una red notablemente completa donde se pueden probar muchas posibilidades. A pesar de ser tan completo, este alimentador de prueba ha sido diseñado para problemas de pruebas de software y fuera de esta aplicación los resultados pueden ser poco realistas. Se usa a menudo en el análisis de redes inteligentes.

1.4.1.9 Alimentador de prueba de 342 nodos

Este sistema de prueba contiene un conjunto de ocho alimentadores primarios de 13,2 kV conectados mediante transformadores triángulo/estrella a tierra a una red mallada de baja tensión para alimentar 50 MVA de cargas desequilibradas. La red de baja tensión diferencia entre el sistema de red de 120/208 V (utilizado en áreas de alta densidad) y el sistema de red de 277/480 V (utilizado en grandes centros de carga).

El objetivo de este sistema de prueba es proporcionar una referencia para los investigadores cuyo objetivo es la evaluación de algoritmos en redes de distribución no radiales. Ha sido diseñado para tratar problemas como sistemas muy interconectados con varias líneas y transformadores de baja tensión en paralelo. Se utiliza principalmente en estudios de DER y análisis de planificación de comunicaciones.

1.4.2 Limitaciones de los Alimentadores de prueba

Las redes de prueba de distribución que representan grandes regiones y coinciden con características de servicios públicos realistas se vuelven necesarias para el desarrollo de herramientas de distribución avanzadas, como OPF, optimización Volt/Var o algoritmos de reconfiguración de red.

El conjunto de alimentadores de prueba descrito anteriormente presenta una serie de limitaciones que dificultan su uso para aplicaciones de sistemas de distribución avanzados. Entre estas limitaciones se han encontrado algunas comunes, como el tamaño pequeño del alimentador, la falta de datos de series temporales que representen la variabilidad de la demanda y la generación, la ausencia de representatividad, la falta de coordenadas geográficas para el trazado físico de la red. También, como se desarrolló anteriormente, los alimentadores de prueba fueron creados para tratar un solo tema técnico o económico y no cuentan con la información suficiente necesaria para otras aplicaciones. A continuación, se desarrolla brevemente los problemas que generan las limitaciones nombradas: [6]

- Alimentadores de tamaños pequeños: El tamaño de la red es un tema importante que se debe considerar para extraer conclusiones confiables de los estudios. El uso del

término “gran escala” se aplica sólo cuando se toman en cuenta múltiples generadores que están conectados a una subestación. En la mayoría de los casos, las redes a gran escala capturan más heterogeneidad en algunos factores como niveles de voltaje, variedad de equipos o configuraciones de red. Esto conduce a resultados y conclusiones más escalables y robustos. Sin embargo, el tiempo de cálculo aumenta drásticamente cuando aumenta el tamaño del problema. El tamaño de los alimentadores de prueba existentes es generalmente pequeño, siendo los más grandes el alimentador de prueba de 8500 nodos y los alimentadores EPRI. Sin embargo, estos sistemas de prueba de tamaño mediano no son suficientes para verificar el rendimiento de las soluciones a gran escala que proporcionan los nuevos algoritmos.

- Falta de datos de series temporales: Los datos de series temporales para demandas y DER permiten un análisis más completo de las operaciones de la red. Por ejemplo, la integración de DERs, como pueden ser los dispositivos de almacenamiento de baterías con limitaciones de tiempo para su óptima gestión (debido a su capacidad de almacenamiento), hace necesaria la utilización de datos de series temporales durante el periodo de estudio de interés. En estos casos, el OPF estándar de un solo período debe transformarse en una optimización de varios períodos. Los OPF de varios períodos permiten soluciones que consideran las restricciones temporales de los DER, como el almacenamiento de energía, la carga de vehículos eléctricos o la respuesta del sistema frente al incremento de la demanda.
- Falta de representatividad: La representatividad de una red de prueba de distribución está relacionada con las características zonales específicas de las redes reales. Por ejemplo, a diferencia de Europa, los alimentadores primarios de EE. UU. consisten en secciones alimentadoras monofásicas y trifásicas que suministran electricidad en las zonas de cobertura particulares. Además, el número de clientes alimentados por un único transformador de distribución de media tensión/baja tensión en los EE.UU es mucho menor que en Europa, así como el tamaño del propio transformador. Como consecuencia, en EE.UU, la longitud de las redes de baja tensión también es más corta que en Europa. Otra diferencia notable es la disposición de los generadores. En Europa, la gran mayoría de los generadores dentro de las áreas urbanas son subterráneos. Sin embargo, en EE.UU, los generadores subterráneos están limitados a algunas áreas residenciales y comerciales específicas. En general, las redes urbanas tienen mayor densidad de carga que las redes rurales. El tipo de topologías de red y el tipo de equipo de red pueden cambiar según las consideraciones geográficas y de desarrollo de una región en particular. Por ejemplo, las redes rurales son topológicamente mucho más radiales, y la existencia de bucles que conectan

diferentes generadores es mucho menos frecuente que en las redes urbanas. Por último, el tipo de infraestructura que utilizan las empresas de servicios públicos cambia de un país a otro o de una región a otra. Por ejemplo, el alimentador de prueba de nodo IEEE 8500 presentado incluye las características detalladas de los transformadores de distribución MV/LV con toma central utilizados en EE.UU.

- Coordenadas geográficas faltantes: las coordenadas del cliente no son relevantes para los cálculos eléctricos; sin embargo, brindan una imagen topológica útil del diseño de la red. Estos problemas topológicos juegan un papel importante en la planificación de la expansión y las posibles estrategias de reconfiguración en caso de fallas en la red. Por ejemplo, el alimentador de prueba de nodos IEEE 8500 incluye las coordenadas geográficas de los diferentes nodos de la red.
- Diseño y datos disponibles para un solo problema: Algunos de los alimentadores de prueba descritos fueron diseñados con el objetivo de modelar y resolver un problema operativo técnico o económico específico y, en general, se vuelven inadecuados para ser utilizados en otro tipo de problemas o aplicaciones, por falta de información relevante.
- Alimentadores aislados: Casi todos los sistemas de prueba de distribución existentes contienen un único alimentador aislado. Esto ignora las interacciones entre los generadores que comparten un transformador de subestación y que complican la reconfiguración del sistema frente a mantenimiento y recuperación de fallas.

1.5 Selección de la red a simular

Con el fin de generar una estadística de los alimentadores de prueba más utilizados en simulaciones en el ámbito de investigación y desarrollo se analizó una muestra de 100 papers publicados en los últimos diez años bajo la temática de *Node Test feeders* y en los cuales se hayan evaluado algoritmos y el desempeño de los alimentadores de prueba. Estos papers fueron extraídos de repositorios del *Institute of Electrical and Electronics Engineers (IEEE Xplore)* [23] y *Elsevier – Journals and Books (Science Direct)* [22], por su relevancia como repositorios relacionados a la ciencia. El criterio de selección de la muestra fue el de seleccionar aproximadamente diez papers por año y, a su vez, seleccionar aquellos con la mayor cantidad de citas. Una vez conformada la muestra, se analizó cada uno de los papers y se realizó un conteo de los alimentadores de prueba simulados, los softwares en donde fueron simuladas las redes, y las temáticas desarrolladas.

En primer lugar, se presenta un gráfico de tortas en el que se visualiza la distribución de los tipos de alimentadores de prueba simulados:

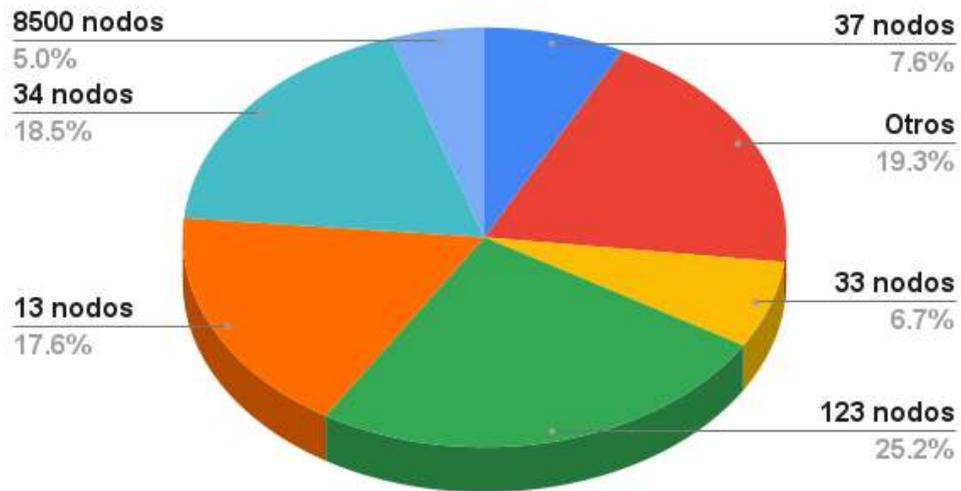


Figura 1-17. Porcentaje de las redes modeladas diferenciadas según la cantidad de nodos.

Nota: En la categoría “Otros” se agruparon las redes que no pertenecen a las desarrolladas por TFWG y que no fueron presentadas en la Sección 1.4.1 del presente trabajo. Algunos de estos alimentadores de prueba fueron creados por los autores a partir de redes reales o eran alimentadores de prueba modificados.

Se observa que los alimentadores de prueba más simulados son los de 123 nodos (25,2%), 34 nodos (18,5%) y 13 nodos (18,5%).

En cuanto a las temáticas tratadas en cada uno de los papers, se las agrupó en categorías con el fin de compactar la información sin perder representatividad:

- Control de Voltaje/VAR
- Detección y análisis de fallas
- Flujo óptimo de potencia (OPF)
- Generación distribuida
- Perfil de voltaje/Pérdidas de energía
- Restauración de servicio

Vale la pena aclarar que en lo que respecta a la categoría “Generación distribuida”, en la gran mayoría de los casos también se trataban otras temáticas como, por ejemplo, la manera en la que la generación distribuida afecta el control del voltaje, o el estudio de algoritmos para restaurar el servicio en redes con generación distribuida, etc. El resultado del porcentaje de temáticas se muestra en Figura 1-18:

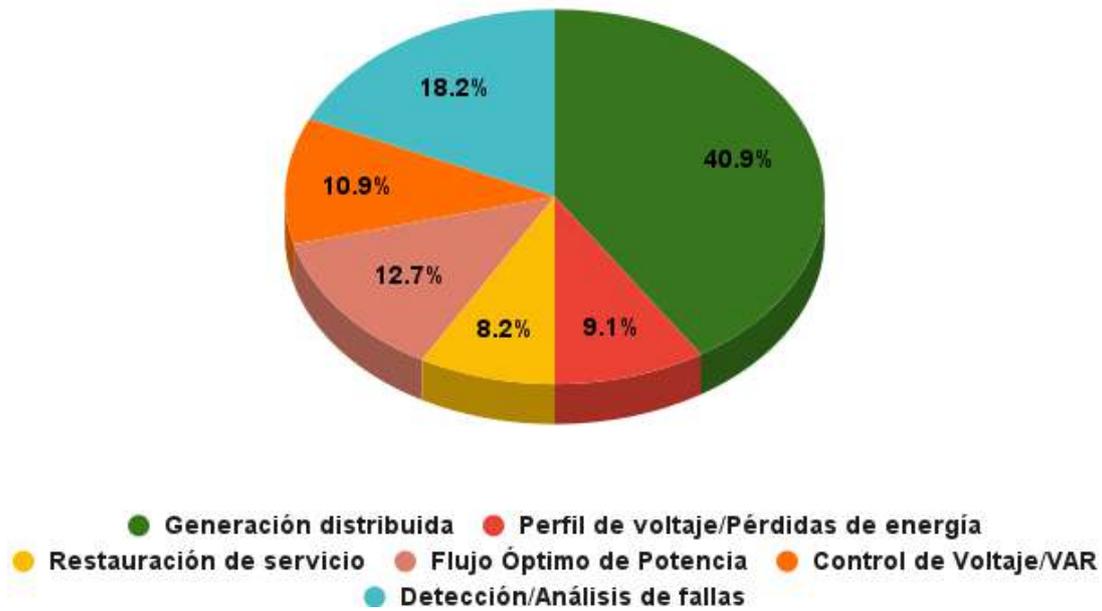


Figura 1-18. Porcentaje de las temáticas desarrolladas en investigaciones utilizando alimentadores de prueba.

Con el objetivo de detectar las tendencias en el uso de los alimentadores de prueba y las temáticas correspondientes en los últimos años, se construyó el siguiente histograma mostrado en la Figura 1-19:

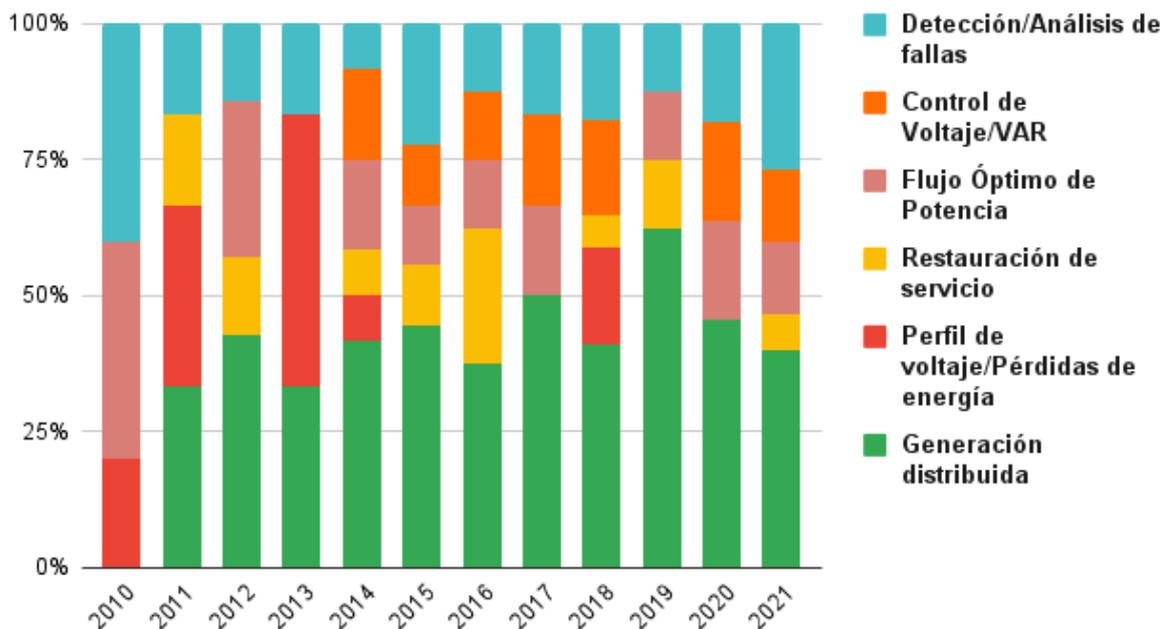


Figura 1-19. Distribución porcentual de temáticas desarrolladas en investigación utilizando alimentadores de prueba.

Se observa que la categoría predominante es la generación distribuida, ganando relevancia en la última década. Esto concuerda con los cambios actuales en la topología de las redes eléctricas, donde las fuentes de energía eólica y solar, por nombrar algunos ejemplos, están

ganando preponderancia en la totalidad de la energía eléctrica generada (Tabla 1-1). Sin embargo, las redes eléctricas convencionales deben adaptarse a estos cambios y solventar los problemas asociados, y es ahí donde surge la necesidad de simular cómo evolucionan las redes eléctricas, y cómo las nuevas tecnologías pueden insertarse de manera exitosa en las redes convencionales.

En lo que respecta a los softwares utilizados en las simulaciones llevadas a cabo en los papers analizados, se destacan: Matlab-Simulink, ATPDraw, DigSILENT, OpenDSS, PSCAD, GridLAB-D.

1.6 Conclusión

Se ha evidenciado la creciente relevancia de las redes de alimentaciones de prueba dentro del ámbito de investigación y desarrollo como una herramienta para analizar una gran variedad de temáticas como, por ejemplo, la detección de fallas en la red, restauraciones de servicio, flujos de potencia, etc. mediante la utilización de algoritmos con el fin de analizar el desempeño de los mismos.

En lo que respecta al análisis estadístico se evidencia que la generación distribuida en conjunto con los análisis de fallas eléctricas y su detección, control del voltaje, restauración de servicio frente a fallas o la búsqueda de flujos óptimos de potencia han ido ganando relevancia en el ámbito de desarrollo. Esto concuerda con la evolución que están sufriendo las redes eléctricas convencionales hacia las redes eléctricas inteligentes, teniendo que analizarse las problemáticas asociadas y donde entran en juego las tecnologías emergentes de los elementos que componen a la red.

Uno de los puntos débiles de los alimentadores de prueba es la falta de representatividad y la disponibilidad de datos reales. Si además se adiciona que no existe una gran cantidad de redes disponibles y que cada una de estos alimentadores fue creado para simular fenómenos puntuales, resulta un punto crítico el analizar cuidadosamente las opciones disponibles e intentar seleccionar aquellas redes que mejor se adapten al problema que se quiera modelar.

Para la selección del alimentador de prueba que se utilizará en las simulaciones de este trabajo, se partió de la base que en este proyecto final se analizará el desempeño del algoritmo de detección de eventos de tensión ASA y del algoritmo eASA.

También se analizará cómo los eventos de tensión se propagan a lo largo de la red resultando en diferencias en las mediciones de los parámetros como la tensión RMS, saltos de fase, etc. dependiendo del sitio en donde se realicen estas mediciones, es decir aguas arriba o aguas debajo de la falla, el nivel de tensión, si hay transformadores entre la falla y el sitio de medición, la impedancia de la red entre la falla y la medición, entre otros. Esta propagación provoca que dos eventos de tensión de diferente naturaleza generen una firma

muy similar, al punto de que los algoritmos de detección fallen en la clasificación de estos eventos.

Por tal motivo, se decidió seleccionar el alimentador de prueba de 34 nodos, ya que resulta ser una red extensa donde podrá analizarse cómo influye la impedancia de la red en la clasificación de los eventos. Esta red posee dos transformadores resultando en dos niveles de tensión, por lo que permitirá analizar el comportamiento de la propagación de eventos a través de transformadores. Por último, el análisis estadístico respalda la selección de este alimentador de prueba, siendo uno de los más utilizados en temáticas relacionadas a los eventos de tensión.

Respecto al software utilizado para realizar las simulaciones, se decidió seleccionar ATP-ATPDraw, siendo uno de los más utilizados en la muestra de papers analizados. Lo que se traduce en una gran cantidad de bibliografía y herramientas disponibles a la hora de llevar a cabo el modelado de los elementos de la red.

1.7 Bibliografía

- [1] American National Standard for Electric Power Systems and Equipment (2011). Voltage Ratings (60 Hz). *ANSI Std C84.1*, pp. 1-11.
- [2] I. Z. F. bin Hussien, A. A. Rahim, and N. Abdullah (2011). *Power Electronics Handbook*, tercera edición. Editorial Boston: Butterworth-Heinemann. <https://www.sciencedirect.com/science/article/pii/B9780123820365000471>.
- [3] Comisión Electrónica Internacional (2009). IEC 60038 - standard voltajes.
- [4] M. H. J. Bollen and I. Y.-H. Gu (2006). *Signal Processing of Power Quality Disturbances*. Wiley - IEEE Press.
- [5] G. Benysek and M. Pasko, (2012). *Power Theories for Improved Power Quality, ser. Power Systems*. Springer-Verlag London, vol. 1.
- [6] Postigo Marcos F., Domingo C. M., Gómez San Román T., Palmintier B., Hodge B., Krishnan V., Cuadra García F., Mather B. (2017). *A Review of Power Distribution Test Feeders in the United States and the Need for Synthetic Representative Networks*. *Energies*. <https://doi.org/10.3390/en10111896>
- [7] Schneider K. P., Mather B. A., Pal B. C., Ten C. -W. , Shirek G. J., Zhu H., Fuller J. C., Pereira J. L. R., Ochoa L. F., de Araujo L. R., Dugan R. C., Matthias S., Paudyal S., McDermott T. E., Kersting W. (2017) *Analytic Considerations and Design Basis for the IEEE Distribution Test Feeders*. *IEEE Transactions on Power Systems*, Volume: 33. Pg: 3181 - 3188. <https://doi.org/10.1109/TPWRS.2017.2760011>
- [8] Téllez E. (2007) *Programa de Ahorro de Energía: Calidad de la Energía*. Cuarto Seminario Nacional de Watergy México.
- [9] M. H. J. Bollen, (1999) *Understanding Power Quality Problems: Voltage Sags and Interruptions*. Wiley - IEEE Press.
- [10] Strack J. L. (2015) *Identificación y clasificación automática de eventos de tensión en redes eléctricas inteligentes*. Universidad Nacional de Mar del Plata.
- [11] Institute of Electrical and Electronics Engineers (2009). IEEE Standard 1159, Recommended Practice for Monitoring Electric Power Quality a Status Update.
- [12] Kersting W. H.(1991) *Radial distribution test feeders*. *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 975-985. <https://doi.org/10.1109/59.119237>

- [13] Leonard L. Grigsby (2007). *Electric Power Generation, Transmission, and Distribution*. The Electric Power Engineering Handbook, Second Edition. CRC Press.
- [14] Oscar Perpiñán Lamigueiro (2013). *Energía Solar Fotovoltaica*. Creative Commons. <https://oscarperpinan.github.io/esf/ESF.pdf>
- [15] Compañía Administradora del Mercado Eléctrico Mayorista S.A (CAMESSA) (2023). Resumen anual 2022. Recuperado de: <https://cammesaweb.cammesa.com/2023/01/13/variables-relevantes-del-mem-resumen-anual-2022/>
- [16] K. Prakash, A. Lallu, F. R. Islam, K. A. Mamun. (2016) *Review of Power System Distribution Network Architecture*. <http://dx.doi.org/10.1109/APWC-on-CSE.2016.030>
- [17] Xinghuo Yu, Carlo Cecati, Tharam Dillon, M. Godoy Simoes. (2011) *The New frontier of Smart Grids, An Industrial Electronics Perspective*. IEEE Industrial Electronics Magazine 5(3): pg. 49 – 63. <http://dx.doi.org/10.1109/MIE.2011.942176>
- [18] Cooper Power Systems (1993). *How Step-Voltage Regulators Operate*. Bulletin 77006.
- [19] T.A. Short (2004). *ELECTRIC POWER distribution handbook*. CRC Press LLC
- [20] Asociación Electrotécnica Argentina (2006). AEA 90364. Reglamentación para la ejecución de instalaciones eléctricas en inmuebles.
- [21] Institute of Electrical and Electronics Engineers. (2017) *IEEE PES Test Feeder*. Recuperado de: <https://cmte.ieee.org/pes-testfeeders/resources/>
- [22] Elsevier B.V. (2023). Science Direct, Journals and Books. Recuperado de: <https://www.sciencedirect.com/>
- [23] Institute of Electrical and Electronics Engineers (2023). IEEE Xplore. Recuperado de: <https://ieeexplore.ieee.org/Xplore/home.jsp>

CAPÍTULO 2

ELEMENTOS DE LA RED

MODELADO COMPUTACIONAL



2 Modelado computacional

Resumen

En este capítulo se desarrollarán las bases del modelado computacional de algunos de los elementos típicos presentes en las Redes Eléctricas, como transformadores, las líneas eléctricas, máquinas eléctricas, entre otros.

A partir de lo desarrollado en el Capítulo 1 de este trabajo, se determinó que la red eléctrica a simular será el alimentador de pruebas de 34 nodos y el software utilizado será ATPDraw. Por este motivo, se describirá brevemente el funcionamiento de ATP-ATPDraw y se presentarán las opciones disponibles del software para los elementos presentes en la red. Con esto se busca facilitar la selección de los modelos disponibles en el software con el fin de obtener resultados confiables y simulaciones eficientes en cuanto a recursos computacionales, siempre teniendo en mente que las simulaciones tendrán el objetivo de recrear eventos de tensión.

2.1 Software ATP - ATPDraw

El uso de softwares de simulación está cobrando cada vez más protagonismo en ámbitos de investigación y desarrollo, convirtiéndose en una herramienta imprescindible a la hora de diseñar y desarrollar modelos en todas las áreas de ingeniería. Particularmente en el área de la Ingeniería Eléctrica, el *Alternative Transients Program* (ATP por sus siglas en inglés) es considerado uno de los programas más utilizados para la simulación computacional de fenómenos transitorios de naturaleza electromagnética en sistemas eléctricos de potencia. Con este software, se pueden simular redes complejas, sistemas de control, perturbaciones en la red eléctrica, entre otros, teniendo amplias capacidades de modelado. Otro punto a favor de ATP es que es un software gratis, aunque no es de dominio público y se requiere una licencia antes de que se otorgue el acceso a los materiales patentados. Sin embargo, la licencia está disponible de forma gratuita para cualquier persona en el mundo que no haya participado voluntariamente en la venta o intento de venta de cualquier programa de transitorios electromagnéticos. A continuación, se presentarán las bases de los cálculos numéricos de ATP, junto a los diferentes modelos de los elementos eléctricos presentes en la red eléctrica seleccionada. Con esto se busca poder seleccionar aquellos modelos que mejor se adapten y den mejores resultados frente a las perturbaciones eléctricas simuladas, que para este trabajo serán los eventos de tensión.

2.1.1 Cálculo numérico

El análisis de transitorios en grandes sistemas de potencia suele realizarse mediante técnicas numéricas en el dominio del tiempo, y las cuales se basan en la integración de las ecuaciones diferenciales que representan el comportamiento del sistema. La integración numérica se utiliza para transformar las ecuaciones diferenciales de los elementos del circuito en ecuaciones algebraicas que involucran voltajes y corrientes, y que utilizan los valores almacenados de tiempos pasados de la simulación para poder realizar las integraciones discretizadas. Estas ecuaciones algebraicas se pueden interpretar como la representación de un circuito equivalente del elemento que se esté simulando (ver Tabla 2-1). Las ecuaciones de toda la red son ensambladas utilizando una técnica de red, por ejemplo, la técnica que involucra ecuaciones de admitancia nodal, y se resuelven como una función del tiempo en instantes discretos. [25]

ATP utiliza la regla trapezoidal de integración para resolver las ecuaciones diferenciales de los componentes del sistema en el dominio del tiempo. Suponiendo que la ecuación diferencial que representa el comportamiento de un elemento de circuito sea la siguiente:

$$\frac{dy(t)}{dt} = x(t)$$

Integrando esta ecuación, la misma puede escribirse de la siguiente manera:

$$y(t) = y(0) + \int_0^t x(z) dz$$

La Figura 2-1 muestra el principio de la integración numérica. El área correspondiente a un intervalo dado (t_{n-1} , t_n) se aproxima mediante un trapecoide, es decir, se supone que $x(t)$ varía linealmente dentro del intervalo definido. Si el valor de y se ha calculado en el tiempo t_{n-1} , el valor en el tiempo t_n se aproximará mediante la siguiente expresión:

$$y_n = y_{n-1} + \frac{x_{n-1} + x_n}{2} * (t_n - t_{n-1})$$

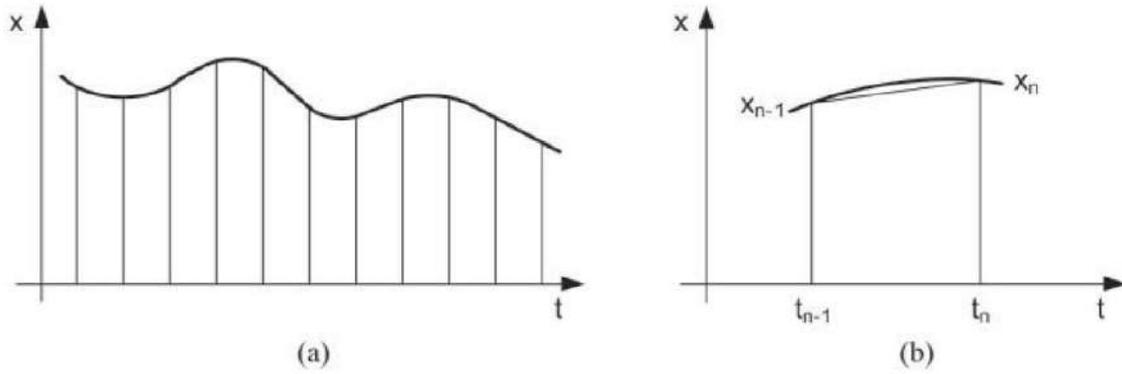


Figura 2-1. Aplicación de la regla trapezoidal: a) Función primitiva. B) Integración trapezoidal [25]

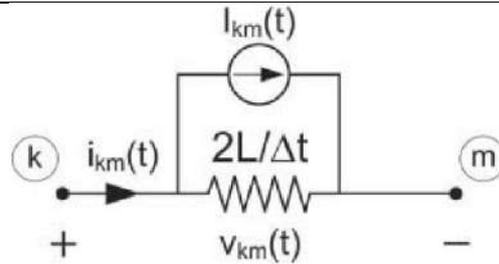
Si el cálculo se realiza utilizando un intervalo de tiempo constante Δt , también conocido como paso de tiempo de integración, la ecuación discretizada se puede expresar de la siguiente manera:

$$y(t) = y(t - \Delta t) + \frac{\Delta t}{2} [x(t) + x(t - \Delta t)]$$

Siguiendo esta metodología es que se obtienen los circuitos equivalentes y sus ecuaciones discretizadas mostrado en la Tabla 2-1 de los siguientes elementos: una resistencia, un inductor, un capacitor y la una línea monofásica sin pérdidas. El desarrollo matemático detallado para obtener las siguientes expresiones puede consultarse en la Sección 3.3 de “Técnicas de solución para el análisis de transitorios electromagnéticos” [25]:

Tabla 2-1. Ecuaciones discretizadas de una resistencia, un inductor, un capacitor y una línea monofásica sin pérdidas [25]

Circuito en el dominio temporal	Circuito discretizado
Resistencia	
$v_{km}(t) = R * i_{km}(t)$	$v_{km}(t) = R * i_{km}(t)$
Inductor	

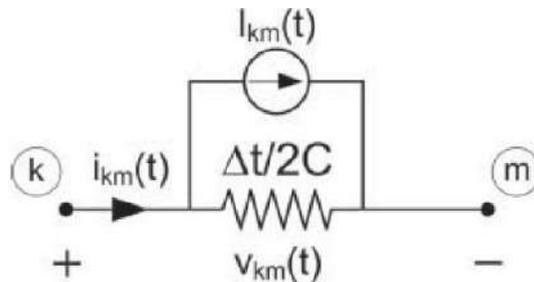
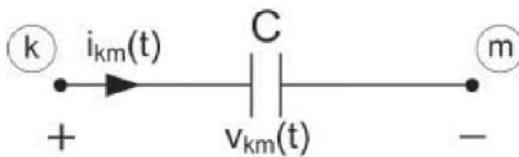


$$v_{km}(t) = L \frac{di_{km}(t)}{dt}$$

$$i_{km}(t) = \frac{\Delta t}{2L} * v_{km}(t) + I_{km}(t)$$

$$I_{km}(t) = \left[\frac{\Delta t}{2L} * v_{km}(t - \Delta t) + i_{km}(t - \Delta t) \right]$$

Capacitor

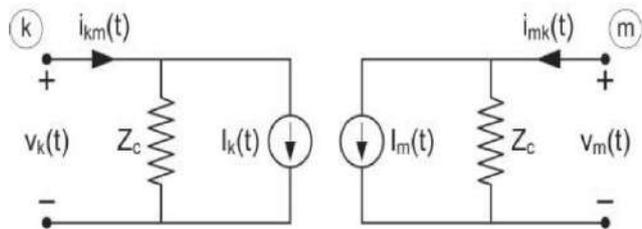
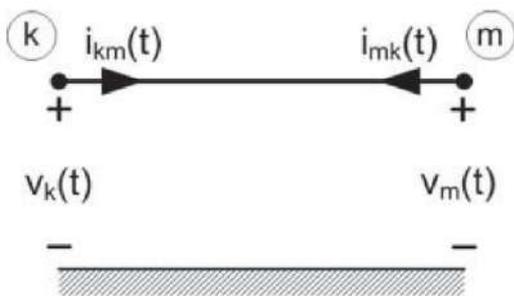


$$i_{km}(t) = C * \frac{dv_{km}(t)}{dt}$$

$$i_{km}(t) = \frac{2C}{\Delta t} * v_{km}(t) + I_{km}(t)$$

$$I_{km}(t) = - \left[\frac{2C}{\Delta t} * v_{km}(t - \Delta t) + i_{km}(t - \Delta t) \right]$$

Línea sin pérdidas



$$\frac{\partial v(x,t)}{\partial x} = -L * \frac{\partial i(x,t)}{\partial t}$$

$$\frac{\partial i(x,t)}{\partial x} = -C * \frac{\partial v(x,t)}{\partial t}$$

$$i_{km}(t) = \frac{v_k(t)}{Z_c} + I_k$$

$$I_k(t) = - \left[\frac{v_m(t - \tau)}{Z_c} + i_{mk}(t - \tau) \right]$$

$$i_{mk}(t) = \frac{v_m(t)}{Z_c} + I_m(t)$$

$$I_m(t) = -\left[\frac{v_k(t - \tau)}{Z_c} + i_{km}(t - \tau)\right]$$

Las condiciones iniciales distintas de cero necesarias para la integración de estas ecuaciones pueden determinarse mediante una solución fasorial de estado estable realizada por el software o pueden ser ingresadas por el usuario.

ATP tiene disponible diferentes modelos de máquinas rotativas, transformadores, líneas de transmisión y cables. También posee una interfaz con los módulos de programa TACS (Análisis transitorio de sistemas de control) y el lenguaje de simulación MODELS [36] [37] que permite modelar sistemas de control y componentes con características no lineales como arcos de tensión y efecto corona, etc. Por nombrar algunos ejemplos, estos módulos permiten modelar perturbaciones simétricas o asimétricas, como fallas, sobretensiones por rayos, varios tipos de operaciones de conmutación, también llevar a cabo análisis de armónicos en el dominio de la frecuencia (*Harmonic Frequency Scan*) y el cálculo de la respuesta de frecuencias mediante la función *Frequency Scan*. En este proyecto se utilizará el módulo MODELS para calcular, a partir de mediciones de la red, los valores RMS y los fasores de las tensiones y corrientes. Estos datos serán almacenados en un formato determinado con el fin de facilitar su posterior lectura para llevar a cabo los cálculos necesarios por el algoritmo ASA y el algoritmo eASA.

2.1.2 Biblioteca de modelos de ATP

En esta sección se resumirán brevemente las bibliotecas de modelos disponibles en ATP. Posteriormente se desarrollarán las bibliotecas de aquellos modelos que serán utilizados en las simulaciones de este proyecto:

- Elementos R,L,C lineales acoplados y desacoplados.
- Líneas y cables de transmisión con parámetros distribuidos y dependientes de la frecuencia.
- Resistencias e inductancias no lineales, inductor con histéresis, resistencia variable en el tiempo, resistencia controlada por TACS/MODELS.
- Componentes no lineales: transformadores incluyendo saturación e histéresis, pararrayos, arcos.
- Interruptores ordinarios, interruptores dependientes del tiempo y dependientes de la tensión, conmutación estadística (estudios de Montecarlo).
- Diodos, tiristores, triacs, interruptores controlados por TACS/MODELS.
- Fuentes del tipo: escalón, rampa, sinusoidal, funciones de sobretensión exponencial, fuentes definidas TACS/MODELS.
- Máquinas rotativas: máquina síncrona trifásica, modelo de máquina universal.

- Componentes eléctricos definidos por el usuario que incluyen la interacción con MODELS.
[24]

2.1.3 Rutinas de soporte

Las rutinas de soporte son utilidades integradas dentro de ATP que ayudan a los usuarios a realizar conversiones de datos disponibles del fabricante del elemento en cuestión y el requerido por el programa, o para calcular parámetros eléctricos de líneas y cables a partir de datos geométricos y sus materiales. Las rutinas de soporte disponibles en ATP son nombradas a continuación:

- Cálculo de parámetros eléctricos de líneas aéreas y cables utilizando los módulos de programa "LINE CONSTANTS", "CABLE CONSTANTS" y "CABLE PARAMETERS".
- Generación de datos de entrada del modelo de línea dependiente de la frecuencia (modelos de línea Semlyen, JMarti, Noda).
- Cálculo de datos modelo para transformadores (XFORMER, BCTRAN).
- Conversión de curvas de saturación e histéresis.
- Modularización de la Base de Datos (para uso de \$INCLUDE).

En la Figura 2-2 se esquematiza el flujo de datos entre el "bloque central", lugar donde se encuentra la representación de la red eléctrica simulada, con las rutinas de soporte y con los módulos de programas TACS y MODELS.

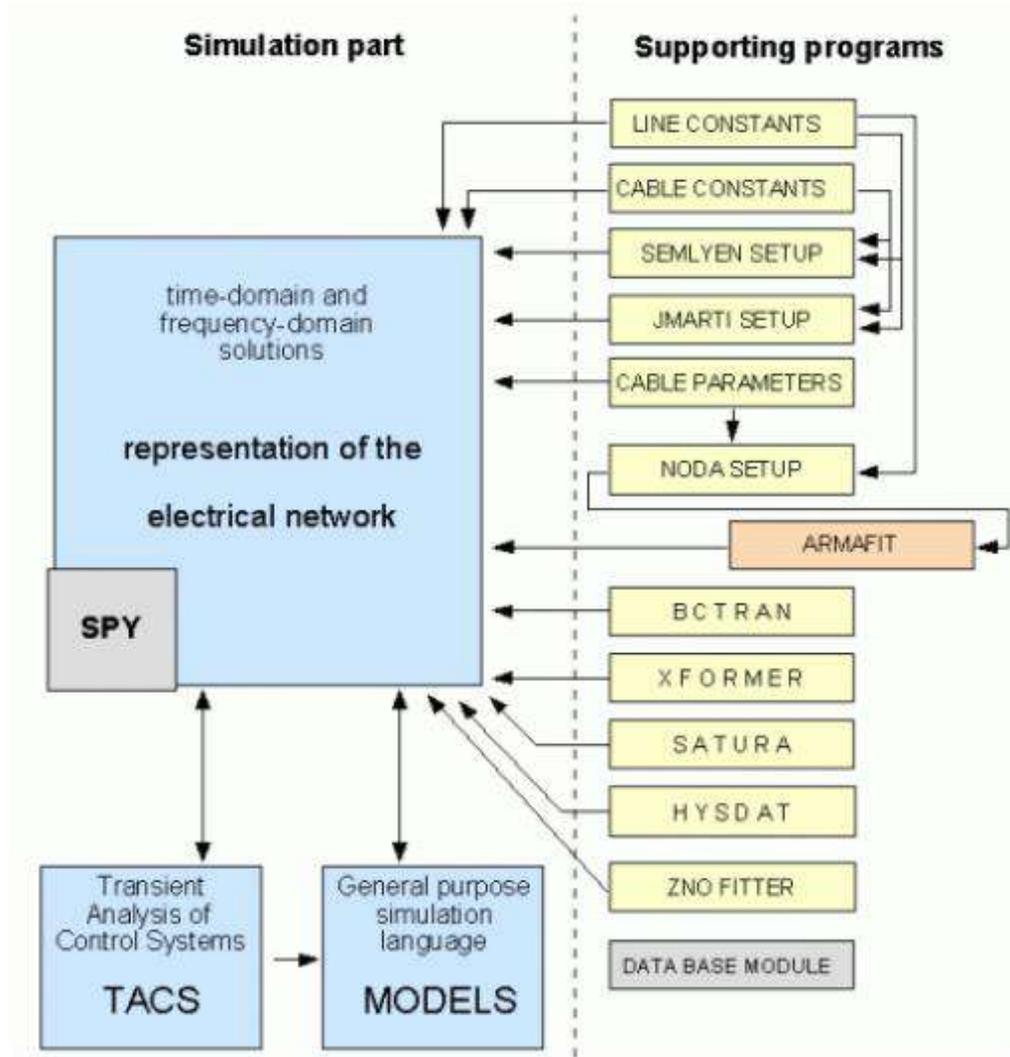


Figura 2-2. Rutinas de soporte en ATP. [24]

2.1.4 Preprocesador ATPDraw

Dentro del Software ATP, se encuentra el preprocesador gráfico llamado ATPDraw, el cual fue programado en *CodeGear Delphi 2007* y se ejecuta en múltiples versiones de Windows. En ATPDraw, el usuario puede construir un circuito eléctrico mediante la selección de los componentes en los diferentes menús disponibles e interconectando cada uno de estos, formando la red eléctrica a simular. Una vez armado el circuito, se guarda el proyecto generándose automáticamente un archivo de extensión “*.acp”. Al ejecutar el programa, se crea automáticamente un archivo de extensión “*.atp” con toda la información de los elementos que pertenecen al circuito eléctrico simulado que posteriormente son utilizados por el motor de cálculo, junto a otros archivos correspondientes a las rutinas de soporte que estén en uso, archivos utilizados para gráficos (.PL4), entre otros. ATPDraw se puede integrar con ATP y otros programas, como por ejemplo ATP Analyzer, GTPPLOT, PlotXY uno de los más utilizados para graficar las soluciones, etc.

Todos los modelos de línea se incluyen mediante un módulo especialmente diseñado denominado LCC. Los diferentes modelos de transformadores y máquinas también poseen sus correspondientes módulos. Adicionalmente, se incluyen ciertos elementos propios de ATPDraw (no presentes en ATP), como el modelo híbrido de transformador o el programa auxiliar Windsyn utilizado para el modelado de máquinas eléctricas. En la Figura 2-3 se muestra un esquema de la Interfaz ATPDraw con los archivos generados una vez simulado el circuito, las rutinas de soporte y con otros programas del paquete ATP-EMTP:

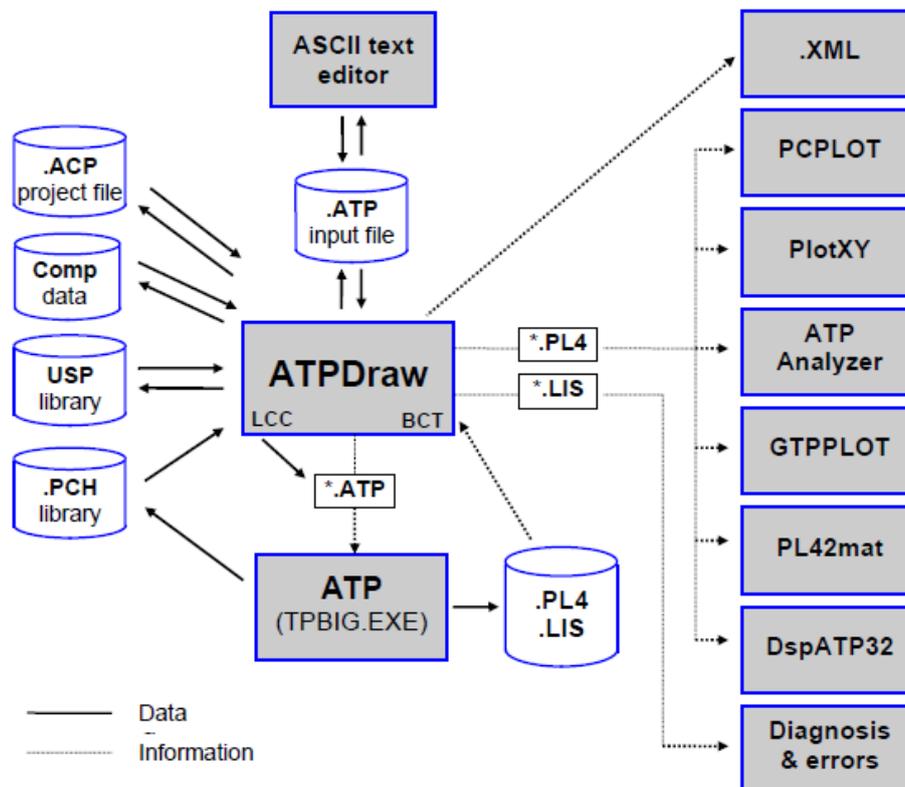


Figura 2-3. Interfaz ATPDraw con otros programas del paquete ATP-EMTP. [24]

2.2 Modelos de elementos de la red

Se desarrollarán las distintas variantes existentes en ATP para modelar cada uno de los elementos utilizados en la red eléctrica. Con esta información, en el Capítulo 4, se procederá a la selección del modelo más adecuado para cada elemento de la red. El fin es obtener resultados fiables y además que la simulación sea eficiente computacionalmente hablando. Es decir, no se utilizarán modelos que tengan buen desempeño en fenómenos electromagnéticos que no sean parte del propósito de la simulación de este proyecto.

2.2.1 Transformadores

En esta sección se abordarán los cuatro tipos de modelos de transformadores que existen en el software ATPDraw:

- Transformador ideal monofásico y trifásico.
- Transformador saturable: monofásico, y trifásico de dos o tres devanados.
- BCTRAN. Modelo de transformador monofásico, y trifásico de 2 o 3 devanados. Autotransformadores, conexiones Y- y D- con todos los desfases posibles. Este modelo admite inductancias magnetizantes no lineales externas.
- Transformador Híbrido (*XFMR – Hybrid Model*). Transformador topológicamente correcto, con informe de pruebas, datos de diseño o ingreso por el usuario de valores típicos. Es trifásico, de 2-4 devanados, con conexiones Y, D y acoplamiento en zigzag.

Tabla 2-2. Transformadores disponibles en el entorno de simulación de ATPDraw. [26]

Selección	Nombre del objeto	Icono	tarjeta ATP	Descripción
Ideal 1 fase	TRAFO_I		FUENTE tipo 18	Transformador ideal monofásico.
Trifásico ideal	TRAFO_I3		FUENTE tipo 18	Transformador ideal trifásico.
Saturable 1 fase	TRAFO_S		RAMA TRANSFORMADOR	Transformador monofásico saturable.
Saturable 3 fases	SATTRAFO		RAMA TRANSFORMADOR	Transformador general saturable. 3 fases. 2 o 3 bobinados.
BCTRAN	BCTRAN		RAMA Tipo 1...9	Soporte directo del modelado de matrices de transformadores BCTRAN.
modelo híbrido	XFMR		RAMA	Resistencia de devanado, inductancia de fuga, núcleo topológicamente correcto, capacitancia. Informe de prueba, datos de diseño o típico.

2.2.1.1 Transformadores Ideales

En el transformador ideal se asume que la magnetización en el núcleo es lineal, es decir, la curva de campo magnético en función de la intensidad de corriente es lineal. De esta manera, la permeabilidad magnética del núcleo es lineal y la conductividad eléctrica es cero, y por lo tanto no hay pérdidas en el núcleo, y como no se representan las pérdidas del bobinado, su resistividad es cero. En general, el transformador ideal no tiene impedancias y simplemente eleva o disminuye la tensión y la corriente del lado primario i_1 al lado secundario i_2 de acuerdo a la relación de espiras entre los devanados. [27]

$$\frac{v_2}{v_1} = \frac{N_2}{N_1} = n$$

$$\frac{i_2}{i_1} = \frac{1}{n}$$

Donde:

- N1: número de espiras en el arrollamiento 1 (en el primario).
- N2: número de espiras en el arrollamiento 2 (en el secundario).
- v1: tensión en el lado primario.
- v2: tensión en el secundario.
- i1: intensidad en el primario.
- i2: intensidad en el secundario.
- n: relación de espiras.

Este modelo tiene un formato de entrada muy simple. Una de sus principales ventajas es que se puede utilizar junto a otros componentes lineales y no lineales de ATP-EMTP, con el fin de representar transformadores de potencia más complejos no disponibles en la versión base de ATP. Sin embargo, en ATP existen de base transformadores de mayor complejidad, por lo tanto, hay que evaluar si modelar externamente los comportamientos no lineales es conveniente o si resulta más práctico seleccionar un modelo ya existente.

2.2.1.2 Transformadores Saturables

Este modelo se basa en la representación del circuito en estrella que se muestra en la Figura 2-7. La rama primaria se trata como una rama R-L desacoplada y cada uno de los otros devanados se modela como un transformador ideal de dos devanados para contemplar las relaciones de transformación. Por otro lado, los efectos de saturación e histéresis se modelan agregando un inductor no lineal adicional en el punto de estrella, y en paralelo una resistencia, obteniendo el modelo del núcleo del transformador.

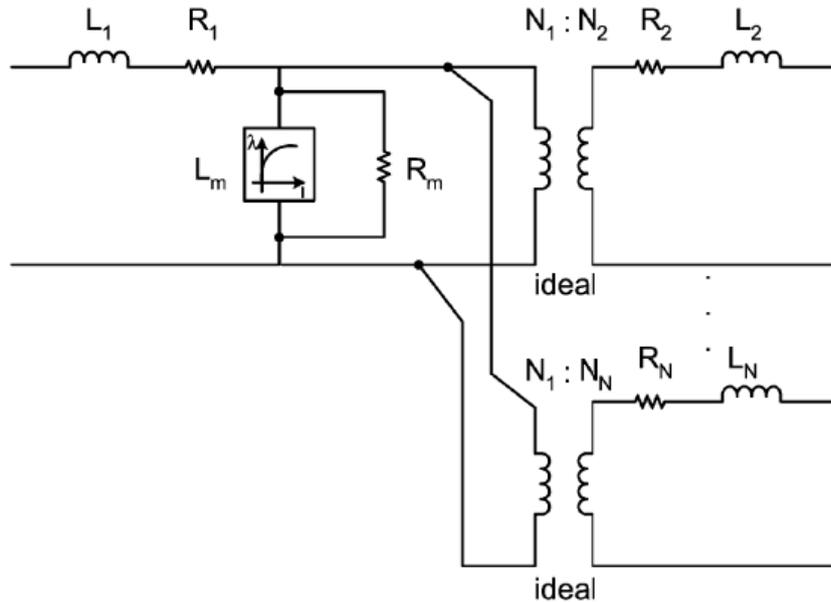


Figura 2-4. Modelo de transformador saturable.[30]

El caso particular de un transformador monofásico de dos devanados en régimen permanente se muestra en la Figura 2-5. [28]

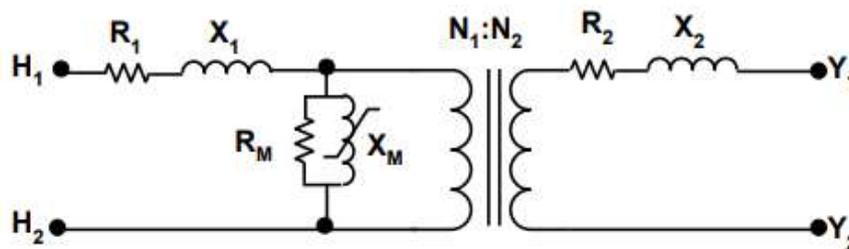


Figura 2-5. Modelo de transformador saturable. [28]

Donde H_1 y H_2 son los terminales del devanado de lado de alta e Y_1 e Y_2 los del lado de baja. R_1 y R_2 son las resistencias de los arrollamientos de alta y baja respectivamente. La reactancia de dispersión del transformador X_{12} se divide entre la reactancia del lado de alta y del lado de baja X_1 y X_2 , respectivamente. R_M representa las pérdidas en vacío y X_M la reactancia de magnetización no lineal. La relación de transformación es $N_1:N_2$, donde N_1 es el número de espiras en el arrollamiento de alta y N_2 en el de baja. La relación de espiras no se puede cambiar dinámicamente durante la simulación para reflejar el cambio de tomas. Las resistencias dependen de la frecuencia y deben ser modificadas para reflejar el amortiguamiento propio a frecuencias más altas.

Este modelo suele utilizarse para casos monofásicos porque no representa adecuadamente ciertos comportamientos presentes en los transformadores trifásicos como acoplamientos mutuos entre fases, ni tampoco fenómenos presentes al estudiar las componentes simétricas.

2.2.1.3 Modelo BCTRAN

El modelo de transformador BCTRAN utiliza una representación matricial de los devanados del transformador sin considerar los efectos de saturación del núcleo, de esta manera tiene en cuenta el acoplamiento de cada devanado de una fase con las demás. La generación de los parámetros de la matriz se realiza mediante la rutina auxiliar BCTRAN incluida en ATP, la cual requiere los datos de los ensayos de cortocircuito y de vacío a frecuencia nominal. La rutina admite transformadores de dos o tres devanados, en conexión estrella o triángulo.

Las ecuaciones de estado estacionario de un transformador multifásico con múltiples devanados se pueden expresar utilizando la matriz de impedancia:

$$[V] = [Z][I]$$

[V]: matriz de tensión

[I]: matriz de corriente.

[Z]: matriz de impedancia, que se separa en sus componentes resistiva [R] e inductiva [L].

Y en calculo transitorio puede reescribirse como:

$$[v] = [R][i] + [L] \left[\frac{di}{dt} \right]$$

Donde la matriz R y L son la parte real e imaginaria respectivamente de la matriz Z. Los valores de estas matrices pueden ser obtenidos de ensayos de cortocircuito y ensayo en vacío. Este enfoque incluye el acoplamiento de fase a fase, modela las características de los terminales, pero no considera las diferencias en la topología del núcleo o del devanado, ya que todos los diseños de núcleo reciben el mismo tratamiento matemático.

Para el estudio de transitorios, es necesario incluir efectos de saturación e histéresis mediante el modelado externo al transformador de estos efectos no lineales. Esta conexión externa no siempre es topológicamente correcta, pero es lo suficientemente buena en muchos casos. Aunque estos modelos son teóricamente válidos solo para la frecuencia a la que se obtuvieron los datos de la placa de identificación, son razonablemente precisos para frecuencias por debajo de 1 kHz. [30]

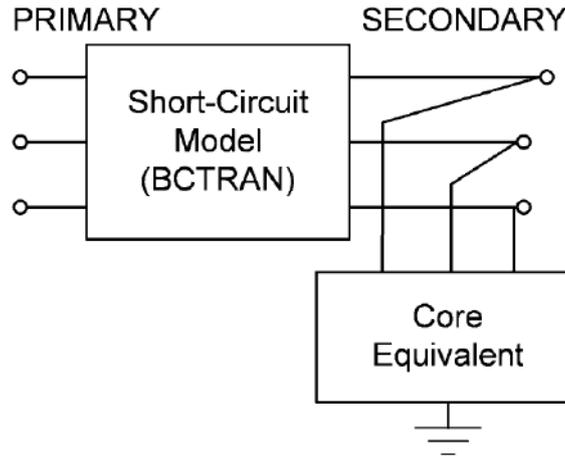


Figura 2-6. Modelo BCTRAN con representación externa del núcleo para considerar no linealidades. [30]

2.2.1.4 Modelo Híbrido

El modelo del transformador Híbrido puede aplicarse para transformadores trifásicos de dos o tres devanados, conexiones estrella, triángulo, zigzag o autotransformador; núcleo acorazado, de tres columnas, de cinco columnas o tipo banco de unidades monofásicas.[26].

Este modelo se basa en el circuito magnético transformado a su dual eléctrico. El flujo principal se separa en un modelo del núcleo, y el de fuga en una matriz de inductancia inversa. Las pérdidas de cobre y las capacidades de la bobina se suman en los terminales del transformador [30]. En la Figura 2-7 se muestra el modelo de un transformador trifásico de dos devanados:

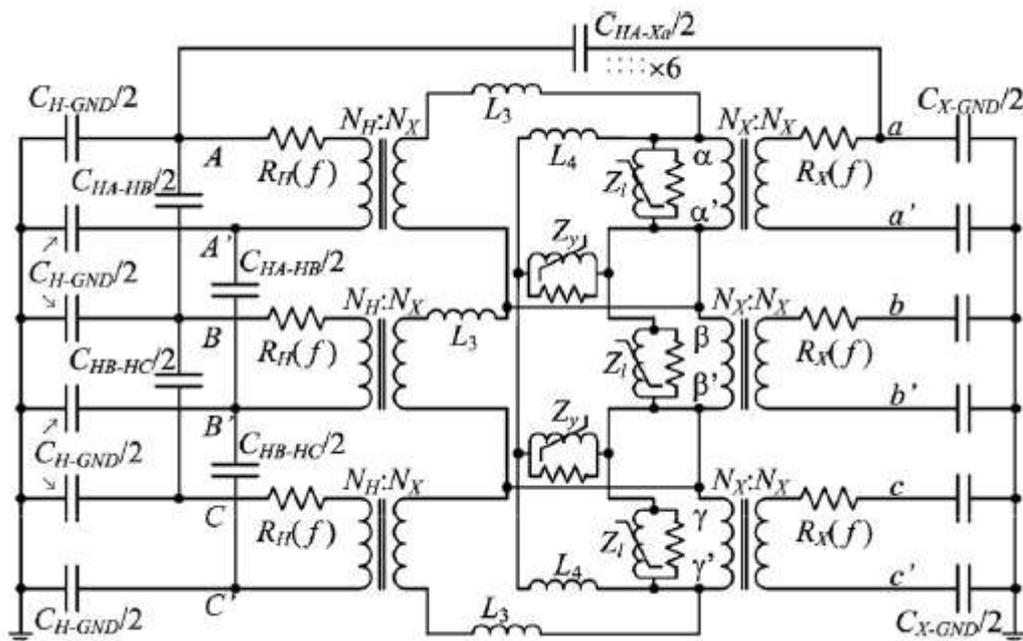


Figura 2-7. Modelo eléctrico del transformador Híbrido de 2 devanados, 3 fases y núcleo de 3 patas. [31]

Los parámetros del transformador se pueden generar a partir de tres fuentes de datos diferentes: valores típicos, informes de pruebas e información de diseño. Las tres fuentes se pueden seleccionar de forma independiente para la resistencia, inductancia, capacitancia y núcleo.

El ingreso de datos de informes de pruebas se basa en pruebas estándar de circuitos abiertos y cortocircuitos, con mediciones de capacitancia como opción adicional.

Los datos de diseño requieren la geometría y los parámetros del material de los devanados y el núcleo.

La opción de valor típico utiliza valores tabulados disponibles en la literatura como por ejemplo de inductancias de fuga, pérdidas de cobre y núcleo y corriente de magnetización, para luego estimar los parámetros del modelo. [31]

2.2.1.4.1 Inductancia de fuga

La inductancia de fuga se modela con una matriz de inductancia inversa (matriz A). La matriz tiene dimensión $(n_w+1)n_p$ donde n_w es el número de devanados físicos, el núcleo está conectado al devanado n_w+1 y n_p es el número de fases. El acoplamiento (automático, Y, D), la relación de espiras y el cambio de fase se producen directamente en la matriz A. Se admiten todos los cambios de fase posibles. La matriz A tiene la siguiente estructura para un transformador trifásico de tres devanados:

$$A = \begin{matrix} & \begin{matrix} A & B & C \end{matrix} \\ \begin{bmatrix} A_w & 0 & 0 \\ 0 & A_w & 0 \\ 0 & 0 & A_w \end{bmatrix} \end{matrix}$$

Donde:

$$A_w = \begin{matrix} & \begin{matrix} P & S & T & C \end{matrix} \\ \begin{bmatrix} a_{11} & \cdots & a_{14} \\ \vdots & \ddots & \vdots \\ a_{41} & \cdots & a_{44} \end{bmatrix} \end{matrix}$$

donde ABC son las tres fases y PSTC significa devanado primario, secundario, terciario y central (n_w+1) . Se supone que la matriz A no tiene acoplamiento mutuo entre las fases. Todo el efecto de secuencia cero se modela en el núcleo adjunto.[31]

2.2.1.4.2 Resistencias de los devanados

Las resistencias de los devanados se suman externamente en el terminal del transformador (matriz A). Opcionalmente, las resistencias pueden depender de la frecuencia. [31]

2.2.1.4.3 Capacidad

La matriz de capacitancia C se divide en dos y se conecta a cada extremo de los devanados físicos:

$$C_w = \begin{bmatrix} C_{11} & \cdots & C_{13} \\ \vdots & \ddots & \vdots \\ C_{31} & \cdots & C_{33} \end{bmatrix}$$
$$C_p = \begin{bmatrix} C_{AA} & \cdots & C_{AC} \\ \vdots & \ddots & \vdots \\ C_{CA} & \cdots & C_{CC} \end{bmatrix}$$

La matriz C_w contiene las capacitancias entre los devanados 1–3 siendo iguales en todas las fases, y se construye como una matriz de admitancia nodal. La matriz C_p contiene capacitancias que son específicas de la fase A, B o C. Por lo general, están conectadas a los devanados externos. La matriz C total se construye sobre estas dos matrices simétricas dependiendo del tipo de devanado. [31]

2.2.1.4.4 Núcleo

El modelo del núcleo está conectado a los terminales de "bobinado central" de la matriz A, admitiéndose núcleos apilados con tres y cinco patas, y las partes del núcleo inductivo y resistivo se tratan de forma independiente. Las pérdidas del núcleo están representadas por una resistencia lineal y las inductancias no lineales están modeladas por la ecuación de *Frolich*:

$$B = \frac{H}{a + B|H|}$$

Donde las constantes a y b se representan como:

$$a = \frac{a_m l_L}{N^2 A_L}$$

$$b = \frac{b_m}{N A_L}$$

se basan en la longitud absoluta (l_L) y el área de la sección transversal (A_L) del núcleo. El parámetro "a" controla la pendiente a bajas excitaciones, por lo que a_m es la inversa de la permeabilidad inicial. El parámetro "b" controla la saturación y b_m es la inversa de la densidad de flujo de saturación completa, que normalmente es de alrededor de 2 T. [30]

Cada parte del núcleo se modela con su propia resistencia de pérdida de núcleo e inductancia no lineal utilizando información sobre su sección transversal y longitudes relativas

de manera de escalar los valores. La Figura 2-8 muestra el modelo del núcleo de un transformador de 5 patas.

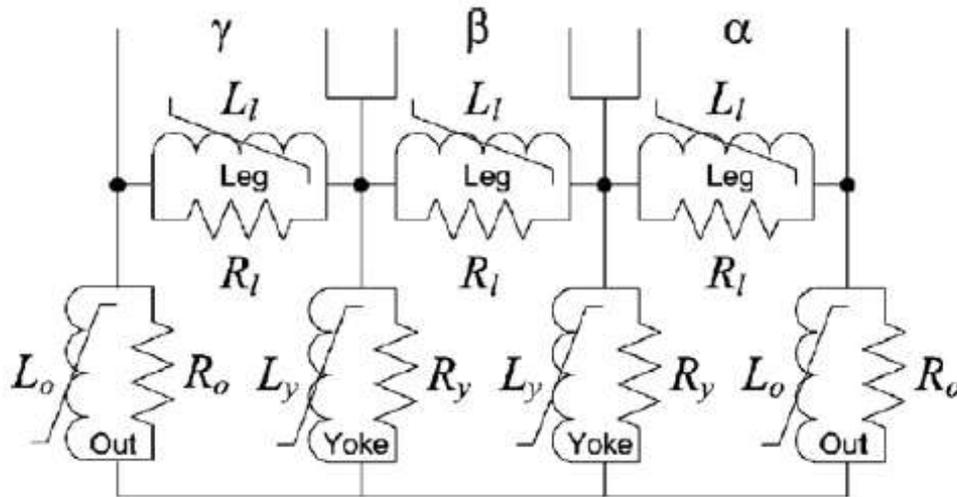


Figura 2-8. Modelo de núcleo apilado de 5 patas. [31]

Se desarrolla una lista de materiales de acero típicos basada en el ajuste de los datos del fabricante de catálogos de última generación. Las pérdidas y las no linealidades del núcleo son desarrolladas en el paper *Hoidalen et al.* [31] y en el *paper Mork et al.* [32]

2.2.2 Líneas eléctricas

El correcto modelado de las líneas eléctricas en los sistemas de potencia es una de las partes sensibles al estudiar los fenómenos transitorios, como por ejemplo sobretensiones por maniobras en la red, por descargas atmosféricas, entre otros. El estudio de los modelos de línea permite entender y conocer cómo será el comportamiento frente a este tipo de fenómenos, a partir de modelos matemáticos que son suficientemente exactos para un rango específico de frecuencia, existiendo diferentes modelos para diferentes aplicaciones que dependerán del fenómeno que se quiera simular. [33] En la Figura 2-9 se presenta una clasificación de los modelos de línea de transmisión para transitorios electromagnéticos.

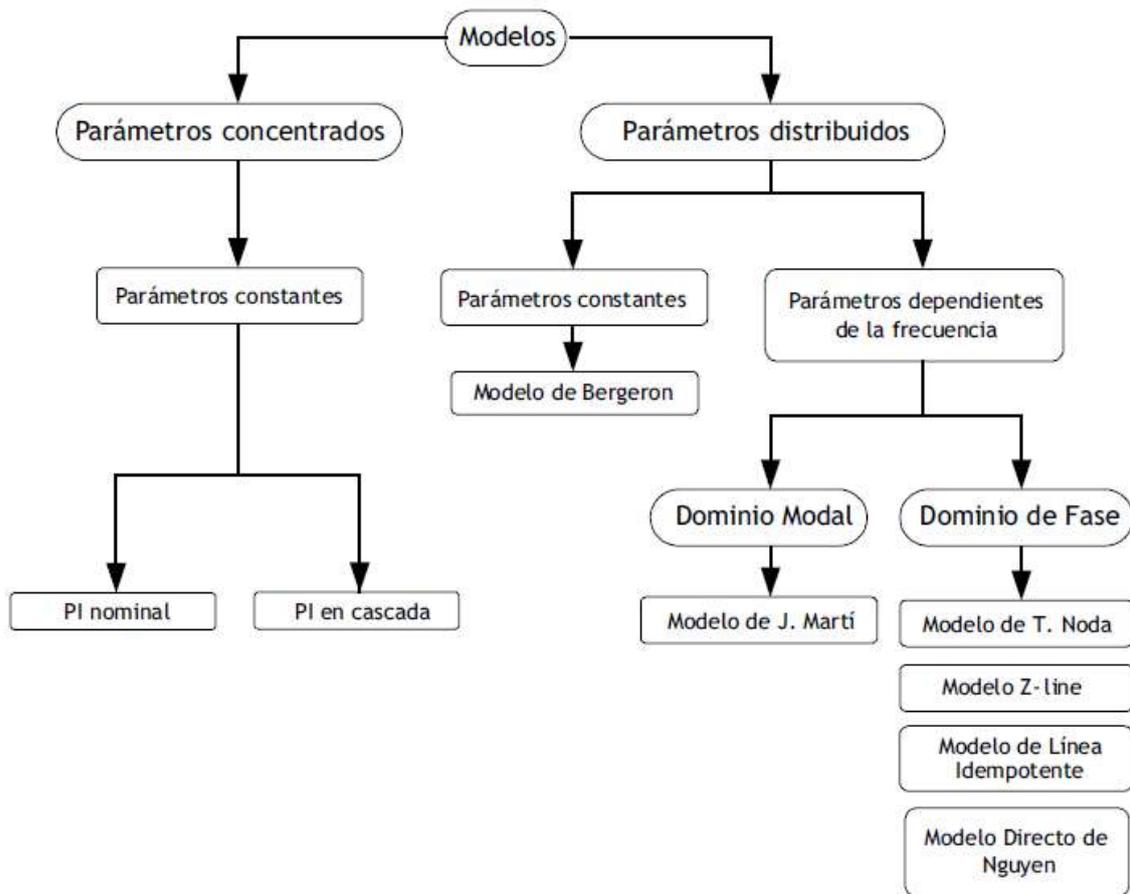


Figura 2-9. Clasificación de los modelos de línea de transmisión para transitorios electromagnéticos. [33]

En el software ATP los modelos de líneas eléctricas pueden ser agrupados en los modelos con parámetros concentrados y aquellos con parámetros distribuidos. Los modelos de parámetros concentrados consisten en la combinación de elementos R, L, G y C, cuyos valores son calculados a una única frecuencia. Por otro lado, los modelos de parámetros distribuidos tienen en cuenta la naturaleza distribuida de los elementos de la línea, y a su vez se subdividen en aquellos donde sus parámetros son constantes y en aquellos donde sus parámetros son dependientes con la frecuencia.

2.2.2.1 Ecuaciones de línea de transmisión y distribución

Las ecuaciones de los modelos de líneas de transmisión serán desarrolladas partiendo del circuito equivalente mostrado en la Figura 2-10. Este circuito representa una sección diferencial de una línea aérea multiconductora, y se representan los acoplamientos entre inductancias en serie y entre capacitancias en derivación.

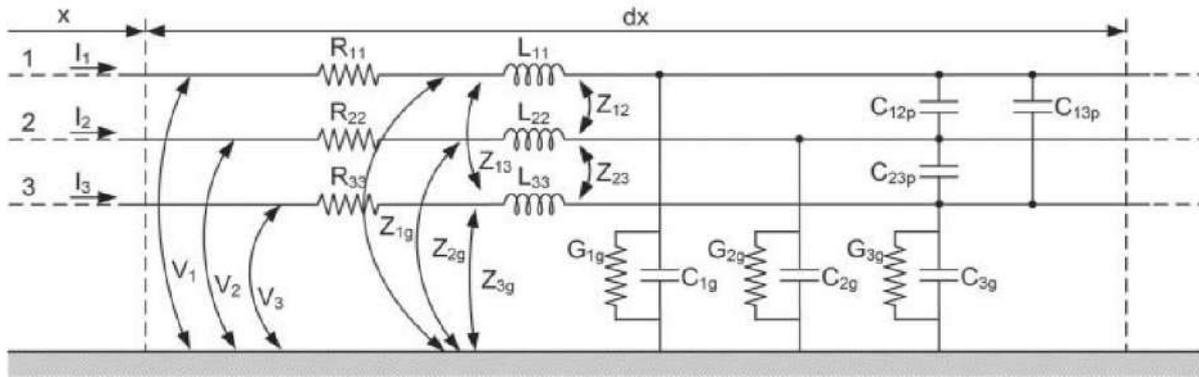


Figura 2-10. Sección diferencial de una línea aérea trifásica. [25]

El comportamiento de esta línea se describe en el dominio de la frecuencia mediante dos ecuaciones matriciales:

$$-\frac{dV_x(\omega)}{dx} = Z(\omega)I_x(\omega)$$

$$-\frac{dI_x(\omega)}{dx} = Y(\omega)V_x(\omega)$$

Ecuación 2-1

donde Z es una matriz compleja y simétrica, cuyos elementos son dependientes de la frecuencia. Para el análisis de transitorios, los elementos de R y L deben calcularse teniendo en cuenta el efecto pelicular en conductores y tierra, y puede llevarse a cabo utilizando la impedancia de tierra de Carson o las fórmulas de impedancia de superficie de Schelkunoff para conductores cilíndricos.[25]

La admitancia de derivación se puede expresar como:

$$Y(\omega) = G + j\omega C$$

donde Y es también una matriz compleja y simétrica, con elementos dependientes de la frecuencia. En general, los elementos de G pueden estar asociados con corrientes que se escapan a tierra a través de cadenas de aisladores. Por lo general, sus valores pueden despreciarse en la mayoría de los estudios, sin embargo, en casos como el efecto corona, los valores de conductancia pueden ser significativos. Es decir, si en la simulación se parte de la base de que los efectos corona son despreciables, los aisladores limpios y clima seco, las conductancias pueden despreciarse.

En cuanto a los elementos C , su dependencia de la frecuencia se puede despreciar dentro del rango de frecuencia que es de interés para el diseño de líneas aéreas. Si las matrices R , L , G y C se suponen constantes, es decir, independientes de la frecuencia, la Ecuación 2-1 se puede expresar como:

$$-\frac{\partial v(x, t)}{\partial x} = Ri(x, t) + L \frac{\partial i(x, t)}{\partial t}$$

$$-\frac{\partial i(x, t)}{\partial x} = Gv(x, t) + C \frac{\partial v(x, t)}{\partial t}$$

donde $v(x, t)$ e $i(x, t)$ son los vectores de tensión y corriente respectivamente. Los modelos avanzados pueden considerar una dependencia adicional de la distancia de los parámetros de la línea (línea no uniforme), el efecto de los voltajes inducidos debido a fuentes distribuidas causadas por rayos cercanos, y la dependencia de la capacitancia de la línea con respecto al voltaje. Dada la dependencia frecuencial de los parámetros de la serie, la aproximación a la solución de las ecuaciones lineales, incluso en cálculos transitorios, se realiza en el dominio de la frecuencia.

La solución general de las ecuaciones de línea con parámetros uniformemente distribuidos en el dominio de la frecuencia se puede expresar de la siguiente manera:

$$I_x(\omega) = e^{-r(\omega)x}I_f(\omega) + e^{r(\omega)x}I_b(\omega)$$

$$V_x(\omega) = Y_c^{-1}(\omega)[e^{-r(\omega)x}I_f(\omega) - e^{r(\omega)x}I_b(\omega)]$$

donde $I_f(\omega)$ e $I_b(\omega)$ son los vectores de las corrientes de onda viajera hacia adelante y hacia atrás en $x = 0$, $r(\omega)$ es la matriz constante de propagación e $Y_c(\omega)$ es la matriz de admitancia característica:

$$r(\omega) = \sqrt{YZ}$$

$$Y_c(\omega) = \sqrt{Z^{-1}Y}$$

$I_f(\omega)$ e $I_b(\omega)$ se pueden deducir de las condiciones de contorno de la línea. Para un caso como el de la Figura 2-11:

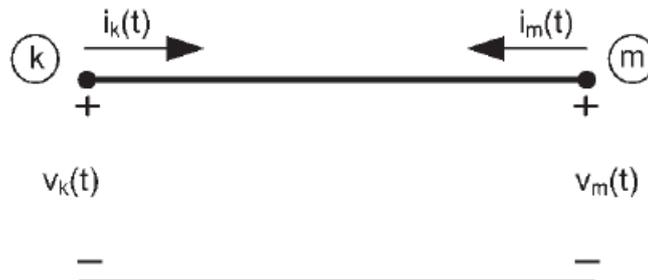


Figura 2-11. Modelo de línea. [25]

la solución en los extremos de la línea se puede formular de la siguiente manera:

$$I_k(\omega) = Y_c(\omega)V_k(\omega) - H(\omega)[Y_c(\omega)V_m(\omega) + I_m(\omega)]$$

$$I_m(\omega) = Y_c(\omega)V_m(\omega) - H(\omega) * [Y_c(\omega)V_k(\omega) + I_k(\omega)]$$

$$H = e^{(-rl)}$$

Ecuación 2-2

siendo l la longitud total. Si se transforma la Ecuación 2-2 al dominio del tiempo resulta:

$$i_k(t) = y_c(t) * v_k(t) - h(t) * [y_c(t) * v_m(t) + i_m(t)]$$

$$i_m(t) = y_c(t) * v_m(t) - h(t) * [y_c(t) * v_k(t) + i_k(t)]$$

donde el símbolo $*$ indica convolución y $x(t) = F^{-1}\{X(\omega)\}$ es la transformada inversa de Fourier. Estas ecuaciones sugieren que una línea aérea puede representarse en cada extremo mediante una admitancia de múltiples terminales en paralelo a una fuente de corriente también de múltiples terminales, como se esquematiza en la Figura 2-12. [25]

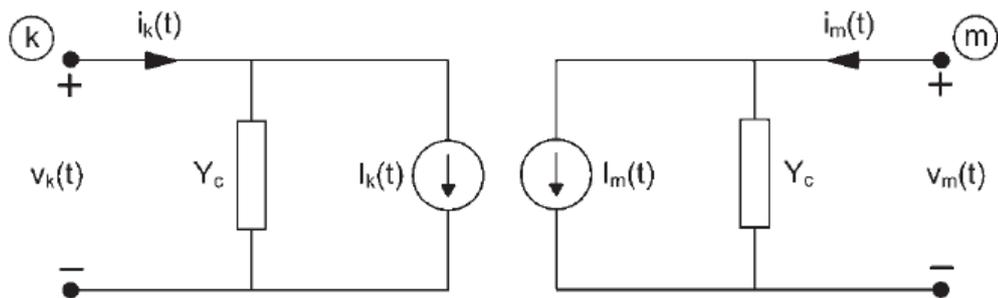


Figura 2-12. Circuito equivalente para simulaciones en el dominio temporal. [25]

A continuación se presentarán brevemente las ecuaciones de las inductancias y capacitancias presentes en las redes eléctricas aéreas. Para ver el desarrollo de estas ecuaciones consultar *Haginomori et al* [34].

2.2.2.1.1 Inductancia entre un solo conductor y tierra

El esquema de un conductor aéreo ubicado sobre la superficie de la tierra, junto a los parámetros necesarios para calcular la autoinductancia presente, se muestra en la siguiente figura:

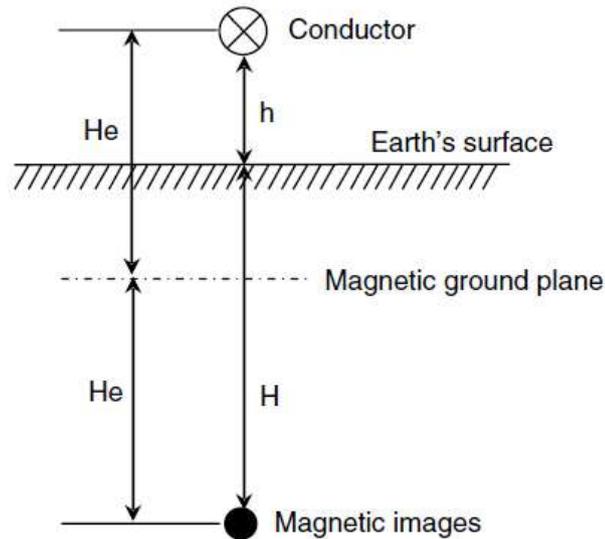


Figura 2-13. Esquema de un conductor aéreo sobre la tierra, junto a los parámetros necesarios para expresa la inductancia generada. [34]

La autoinductancia se expresa como:

$$L = \frac{\mu_0}{2\pi} \left(1 + 2 \ln \frac{2H_e}{r} \right)$$

La variación de la autoinductancia en función de H_e se presenta en tablas tabuladas en la Sección 2.1 de Haginomori et al [34]:

2.2.2.1.2 Capacitancia entre un solo conductor y tierra

En la figura 2.2 se muestra el caso de un conductor de radio r (m) ubicado a h (m) de altura sobre tierra y con una carga eléctrica $+q$ (c) por unidad de longitud:

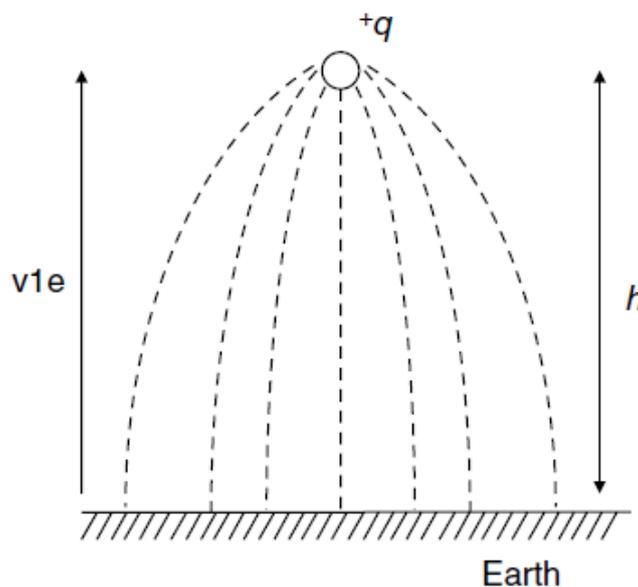


Figura 2-14. Líneas del campo eléctrico a tierra desde un conductor. [34]

Se muestran las líneas del campo eléctrico hacia tierra, y la distribución del voltaje en tierra se simplifica en un perfil plano. Partiendo de la ecuación de tensión entre dos conductores en función de la distancia entre ellos, el radio del conductor y la permitividad del aire que los rodea:

$$v_{11} = \frac{q}{\pi\epsilon_0} \ln \frac{2h}{r}$$

Luego al tratar a la tierra como un conductor:

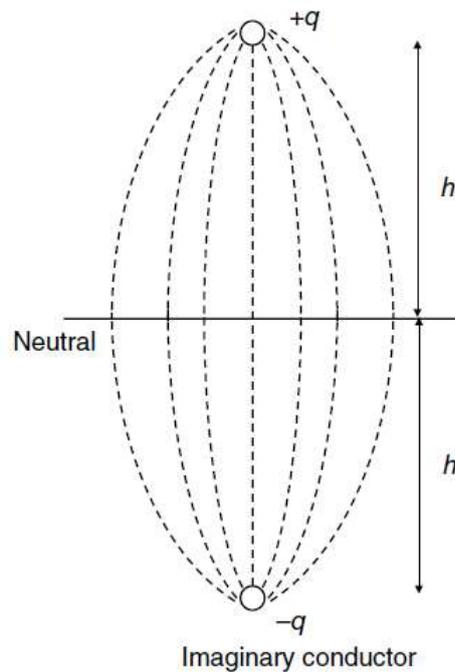


Figura 2-15. Líneas de campo eléctrico entre el conductor y un conductor imaginario presente en la tierra. [34]

Donde generalmente la permitividad de la Tierra, ϵ_e , es bastante grande en comparación con la permitividad del aire, ϵ_a , el voltaje entre el conductor y la tierra se expresa como:

$$v_{1e} = \frac{v_{11}}{2} = \frac{q}{2\pi\epsilon_0} \ln \frac{2h}{r}$$

Luego la capacitancia entre el conductor y la tierra se expresa mediante la siguiente ecuación:

$$C_{1e} = \frac{q}{v_{1e}} = \frac{2\pi\epsilon_0}{\ln \frac{2h}{r}}$$

En el caso de una línea transpuesta, siendo un caso típico en líneas eléctricas, la capacitancia de secuencia positiva se expresa mediante la siguiente ecuación:

$$C_1 = \frac{2\pi\epsilon_0}{\ln \frac{D}{R'}}$$

Esa capacitancia se define en función de la distancia equivalente D entre las fases y el radio equivalente del conductor R', y donde la altura de la línea no afecta la capacitancia.

La capacitancia de secuencia cero se expresa como:

$$C_0 = \frac{2\pi\epsilon_0}{\ln \frac{8h^3}{R'D^2}}$$

La relación entre la capacitancia de secuencia positiva y cero es:

$$C_1 = C_0 + 3C_m$$

Siendo C_m la capacitancia entre fases, y que despejando se obtiene:

$$C_m = \frac{1}{3}(C_1 - C_0)$$

Finalmente se muestran las impedancias entre las fases y tierra y entre fases en una terna de conductores:

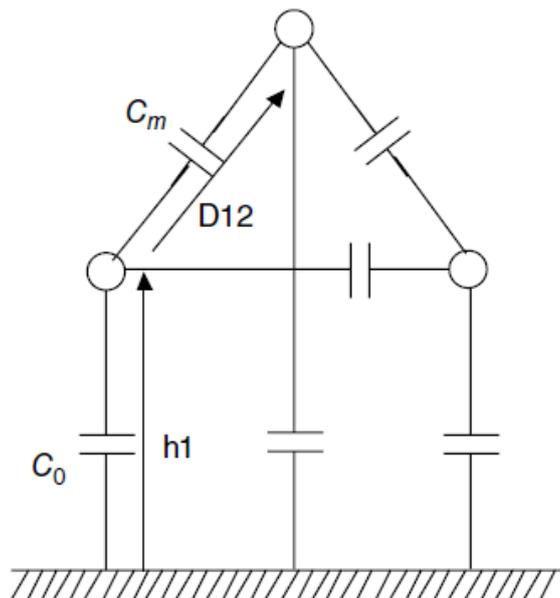


Figura 2-16. Capacitancias entre tres conductores, entre fases y entre fase y tierra. [34]

A partir del desarrollo previo se deducen las matrices de capacitancia para tres fases conductoras situado sobre tierra. Por la extensión en el desarrollo y por el tamaño de las matrices se decidió no agregarlos a este capítulo, pero el lector puede consultar en la *Sección 2.1.1.1.2.1 Matriz de capacitancia de Haginomori et al* [34].

2.2.2.2 Modelo Line Constants

En ATP pueden obtenerse los parámetros y modelos de líneas aéreas por medio de una rutina de soporte conocida como *Line Constants*. Para utilizar esta rutina de soporte se deben ingresar los datos de la geometría de las torres de línea que se desea modelar, es decir las coordenadas (x,y) de los conductores instalados en la torre, la flecha, las dimensiones físicas, la resistividad DC de los conductores y la resistividad del terreno. Por lo general, no se requiere una gran precisión al especificar valores de entrada si el objetivo es duplicar transitorios de baja frecuencia y de frente lento, pero se debe tener un cuidado mayor principalmente con el valor de resistividad del suelo si el objetivo es simular transitorios rápidos.

Una vez ingresados los datos de la geometría se debe seleccionar el tipo de modelo de línea que se desea utilizar, los cuales se nombran a continuación:

- Circuitos PI nominales o equivalentes de parámetros agrupados a la frecuencia especificada
- Modelo de parámetros distribuidos constantes a la frecuencia especificada (Bergeron)
- Modelo de parámetros distribuidos dependiente de la frecuencia ajustado para un rango de frecuencia dado. (Jmarti, Semlyen, Noda)

Estos modelos se desarrollan en la próxima sección del presente proyecto.

Como información proporcionada por la rutina se destacan: la capacitancia o la matriz de susceptancia; la matriz de impedancias en serie; resistencia, inductancia y capacitancia por unidad de longitud para secuencias cero y positivas, a una frecuencia dada o para un rango de frecuencia especificado, la impedancia de sobretensión, atenuación, velocidad de propagación y longitud de onda para secuencias cero y positivas, a una frecuencia dada o para un rango de frecuencia específico. Estos datos en formato matricial pueden estar definidos en el dominio de la fase, en el dominio modal o en el dominio de la secuencia, según lo defina el usuario. [25]

La rutina *Line Constants* también es la encargada de construir la arquitectura de los archivos de línea a implementar en las simulaciones, interactúa con otras rutinas para la construcción de los modelos dependientes de la frecuencia, etc. Exceptuando a los modelos PI y Noda, el resto de modelos de línea de transmisión en ATP hacen uso de la técnica de la transformación modal para manipular líneas polifásicas.

2.2.2.3 Modelo Pi

El modelo PI nominal se basa en la aproximación discreta de los parámetros distribuidos constantes, cuya impedancia serie total se modela en una única impedancia concentrada, y

la admitancia en derivación se divide en dos, ubicándose al inicio y al final de la línea, como se muestra en Figura 2-17:

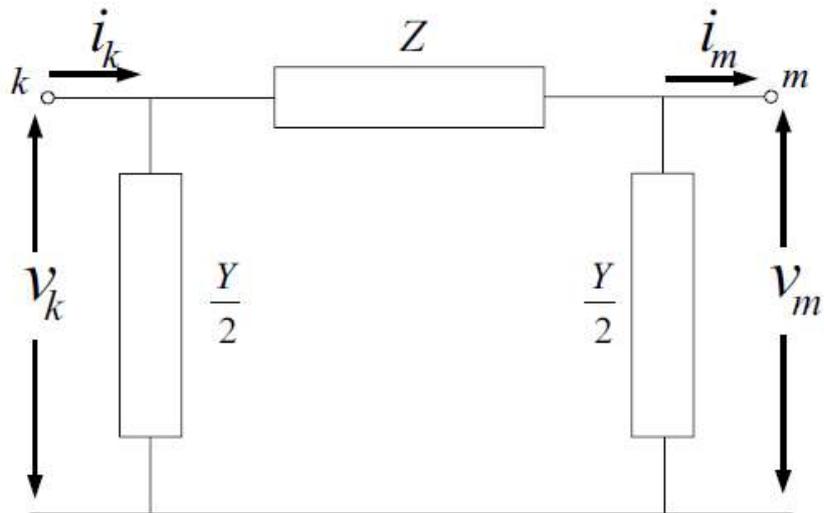


Figura 2-17. Circuito equivalente del Modelo Pi nominal. [32]

Las ecuaciones que relacionan los voltajes y corrientes en los extremos son:

$$v_k = \left(\frac{ZY}{2} + 1 \right) v_m + Zi_m$$

$$i_k = Y \left(1 + \frac{ZY}{4} \right) v_m + \left(\frac{ZY}{2} + 1 \right) i_m$$

y:

$$Z = l(R' + j\omega L')$$

$$\frac{Y}{2} = \frac{l}{2}(G' + j\omega C')$$

Donde:

V_k = Voltaje en el nodo k

i_k = Corriente en el nodo k

v_m = Voltaje en el nodo m

i_m = Corriente en el nodo m

R' = Resistencia por unidad de longitud [Ω / km]

L' = Inductancia por unidad de longitud [H / km]

C' = Capacitancia por unidad de longitud [F / km]

G' = Conductancia por unidad de longitud [S / km]

l = Longitud total de la línea [km]

Z= Impedancia serie de la línea de transmisión

Y=Admitancia en derivación de la línea de transmisión

Este modelo suele utilizarse en simulaciones en régimen estacionario, en líneas cortas de longitudes menores a 15 km [35], y a la frecuencia de la red. Como defecto del modelo, no se puede representar la dependencia con la frecuencia de los parámetros de la línea y deben aceptarse oscilaciones espurias provocadas por los elementos concentrados, y que pueden amortiguarse mediante el uso de resistencias en paralelo con las ramas R-L. Por tal razón, los modelos PI nominal no son generalmente el mejor modelo para analizar soluciones transitorias, fundamentalmente porque las ondas viajeras no se reproducen en forma precisa. Una solución para aproximar la naturaleza distribuida de una línea larga es el modelo Pi nominal en cascada, Figura 2-18, sin embargo, este modelo requiere un mayor tiempo de simulación y tiene una menor precisión si se lo compara con otros modelos específicos utilizados para soluciones transitorias. [33]

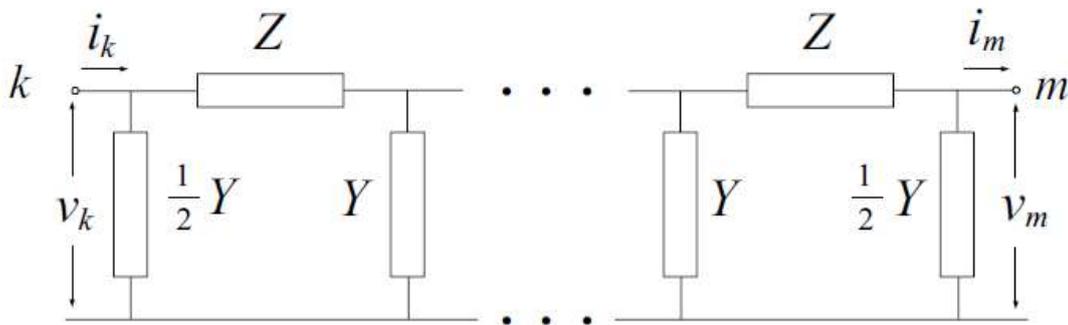


Figura 2-18. Circuito equivalente del modelo Pi nominal en cascada. [33]

Para obtener las variables de entrada con respecto a las variables de salida, el sistema se representa como una red de parámetros de dos entradas:

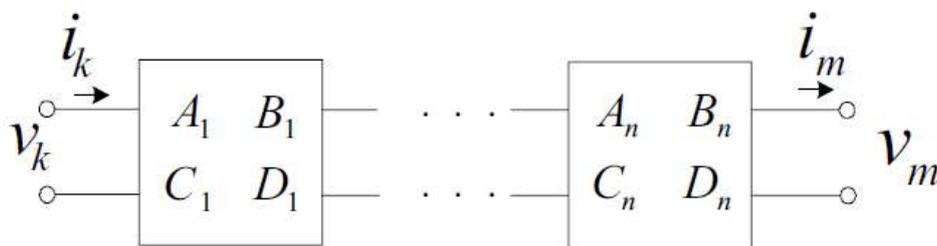


Figura 2-19. Red n-ésima de dos puertos para una línea de transmisión completa. [33]

Si todos los circuitos corresponden a segmentos de igual longitud, entonces el modelo de la línea se representa mediante la ecuación, siendo n la cantidad de segmentos:

$$\begin{bmatrix} v_k \\ i_k \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^n \begin{bmatrix} v_m \\ i_m \end{bmatrix}$$

Restrepo et al [33] desarrolla el cálculo para determinar la cantidad de secciones Pi nominales en cascada para una correcta representación de la línea, y que depende principalmente de la frecuencia del fenómeno transitorio a estudiar.

Una ventaja del modelo Pi nominal es que no condicionan el paso de tiempo de la simulación. En cambio, otros parámetros si afectan el paso del tiempo para aumentar la precisión en los cálculos, con lo cual el modelo Pi nominal es adecuado para analizar transitorios donde se requiera ahorrar tiempo de cálculo.

Como desventaja, este modelo no es muy exacto para fenómenos basados en ondas viajeras (sobrevoltajes por maniobra y por descargas atmosféricas), y en el caso del modelo Pi Nominal en cascada no es eficiente debido a la cantidad de secciones de circuitos Pi utilizadas en el momento de representar este tipo de fenómenos.

2.2.2.4 Modelo Bergeron

Este modelo de parámetros distribuidos constantes fue desarrollado por Hermann Dommel y fue el primero de esta clase que se implementó en ATP-EMTP. El modelo Bergeron está basado en la propagación de las ondas en una línea de transmisión sin pérdidas, y con parámetros L' y C' constantes y distribuidos a través de la línea de transmisión y cuyo circuito equivalente se muestra en la Figura 2-20:

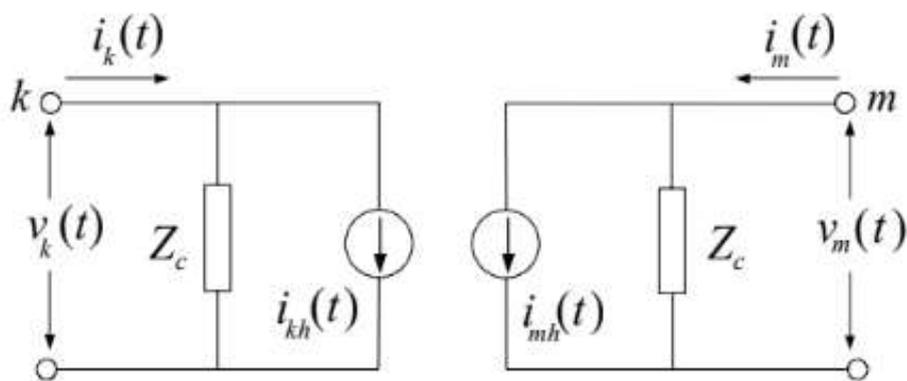


Figura 2-20. Circuito equivalente del modelo Bergeron. [33]

Las ecuaciones que relacionan los voltajes y corrientes en los extremos de la línea son:

$$i_k(t) = \frac{1}{Z_c} v_k(t) + i_{kh}(t)$$

$$i_m(t) = \frac{1}{Z_c} v_m(t) + i_{mh}(t)$$

$$i_{kh}(t) = -\frac{1}{Z_c} V_m(t - \tau) - i_m(t - \tau)$$

$$i_{mh}(t) = -\frac{1}{Z_c} v_k(t - \tau) - i_k(t - \tau)$$

Donde:

τ : tiempo de desplazamiento de la onda viajera.

Y la impedancia característica Z_c se representa como:

$$Z_c = \sqrt{\frac{L'}{C'}}$$

Estas ecuaciones fueron deducidas en la *Sección 4 del libro Neville Watson, Jos Arrillaga* [35].

Para sistemas M-fásicos, las líneas se desacoplan por medio del Método de Descomposición Modal, con el fin de estudiar cada modo como una línea monofásica. Esta descomposición se realiza por medio de matrices de transformación constantes con componentes reales, dando resultados exactos para líneas balanceadas. Para líneas desbalanceadas el modelo no es útil, ya que la suposición de que la matriz de transformación sea constante no es válida, porque las matrices son dependientes de la frecuencia. Si se quieren incluir en el modelado fenómenos como el *efecto skin*, y el efecto de retorno a tierra a medida que aumenta la frecuencia, es necesario considerar la dependencia en frecuencia de los parámetros, teniendo que utilizar modelos que consideren parámetros dependientes de la frecuencia. [33]

La limitación principal del modelo de Bergeron es que el paso de tiempo de la simulación Δt debe ser menor que el tiempo de propagación τ , de forma tal que entre los extremos de la línea k y m exista el efecto de desacople durante la simulación. Como conclusión, el modelo Bergeron resulta ser una buena elección para estudios alrededor de la frecuencia fundamental, que coincide con la frecuencia sobre la cual se calcularon los parámetros distribuidos. Este método no resulta adecuado para representar líneas en un amplio espectro de frecuencias durante condiciones transitorias.

2.2.2.5 Modelo JMarti

El modelo JMarti utiliza la técnica de la transformación modal para la resolución de las ecuaciones de la línea en el dominio del tiempo. La constante de propagación y la impedancia característica modales se calculan en un determinado rango de frecuencias y que se

establece por el usuario. El ajuste de la constante de propagación y la impedancia característica se realiza con aproximaciones asintóticas mediante una función racional de alto orden con polos reales. Luego, el cálculo en el dominio del tiempo de los transitorios es realizado mediante convoluciones numéricas recursivas.

Debido a la naturaleza distribuida de las pérdidas y a la dependencia en frecuencia de los parámetros, es más conveniente desarrollar las ecuaciones de línea en el dominio de la frecuencia, como se ilustra en las ecuaciones presentadas a continuación [33]:

$$V_k(\omega) = Z_c(\omega) * I_k(\omega) + E_{mh}(\omega)$$

Ecuación 2-3

$$V_m(\omega) = Z_c(\omega) * I_m(\omega) + E_{kh}(\omega)$$

Ecuación 2-4

Donde las fuentes de voltaje en los nodos k y m se representan como:

$$E_{mh} = A(\omega) * F_{pk} = [V_k(\omega) + Z_c(\omega) * I_k(\omega)] * e^{-\gamma(\omega)*l}$$

$$E_{kh} = A(\omega) * F_{pm} = [V_m(\omega) + Z_c(\omega) * I_m(\omega)] * e^{-\gamma(\omega)*l}$$

La impedancia característica Z_c y la constante de propagación γ de las ecuaciones, se desarrollan a continuación:

$$Z_c(\omega) = \sqrt{\frac{R'(\omega) + j\omega L'(\omega)}{G'(\omega) + j\omega C'(\omega)}}$$

$$\gamma(\omega) = \sqrt{(R'(\omega) + j\omega L'(\omega)) * (G'(\omega) + j\omega C'(\omega))}$$

El circuito equivalente de la Ecuación 2-3 y Ecuación 2-4 en el dominio de la frecuencia, se ilustra en la Figura 2-21.

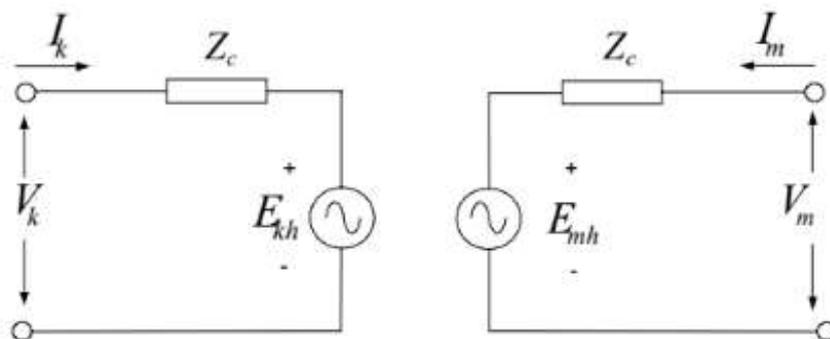


Figura 2-21. Circuito equivalente del modelo Jmarti. [33]

Transformando la Ecuación 2-3 y Ecuación 2-4 al dominio del tiempo, se obtienen las ecuaciones:

$$v_k(t) = Z_c i_k(t) + [v_m(t) + Z_c i_m(t)] a(t)$$

$$v_m(t) = Z_c i_m(t) + [v_k(t) + Z_c i_k(t)] a(t)$$

Este modelo no es muy exacto en bajas frecuencias y para secciones de línea muy cortas, debido a las imprecisiones presentes para los cálculos en el dominio del tiempo.

2.2.2.6 Modelo Semlyen

Al igual que el modelo anterior, el modelo Semlyen utiliza la técnica de la transformación modal para la resolución de las ecuaciones de la línea en el dominio del tiempo. La constante de propagación y la impedancia característica modal son ajustadas a una función racional de segundo orden con polos reales, y el cálculo en el dominio del tiempo de los transitorios se realiza mediante convoluciones numéricas recursivas. Debido a las simplificaciones sobre las cuales se construye este modelo, dejó de utilizarse por ser una aproximación insuficiente y con resultados no tan fiables si se lo compara con el modelo Jmartí. [39]

2.2.2.7 Modelo Noda

El modelo de NODA representa un nuevo enfoque, y la diferencia con los modelos JMartí y Semlyen es que el cálculo se hace directamente en el dominio de la frecuencia, evitándose el inconveniente de la matriz de transformación. En general es más dificultoso obtener un modelo adecuado para una determinada línea y su uso más común es el de modelar cables de potencia.

En este modelo, las ecuaciones que relacionan los voltajes y corrientes en los extremos de la línea son: [33]

$$I_k(\omega) = Y_c(\omega) * V_k(\omega) - e^{-i\omega\tau} H^T(\omega) \{Y_c(\omega) * V_m(\omega) + I_m(\omega)\}$$

$$I_m(\omega) = Y_c(\omega) * V_m(\omega) - e^{-i\omega\tau} H^T(\omega) \{Y_c(\omega) * V_k(\omega) + I_k(\omega)\}$$

Transformando las ecuaciones en el dominio del tiempo, se obtiene:

$$i_k(t) = y_c(t) * v_k(t) - i'_k(t)$$

$$i_m(t) = y_c(t) * v_m(t) - i'_m(t)$$

Donde:

$$i'_k = h^T(t) * \{y_c(t) * v_m(t - \tau) + i_m(t - \tau)\}$$

$$i'_m = h^T(t) * \{y_c(t) * v_k(t - \tau) + i_k(t - \tau)\}$$

El circuito equivalente de las ecuaciones en el dominio temporal se ilustra en la siguiente figura:

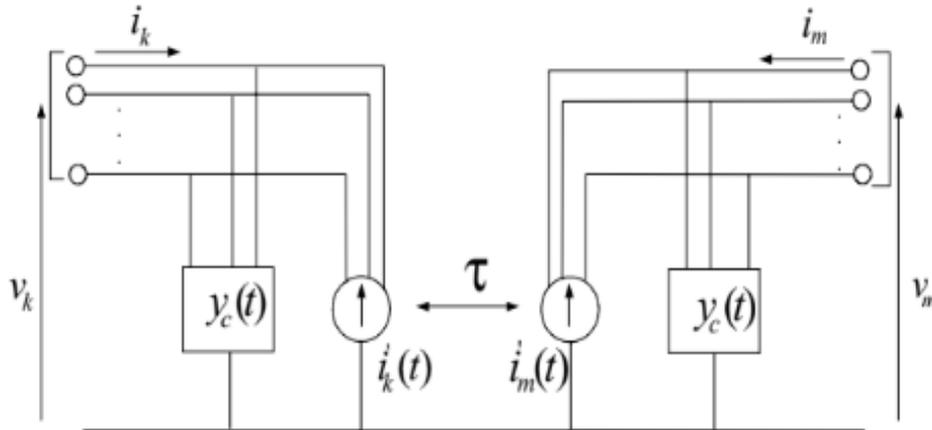


Figura 2-22. Circuito equivalente del modelo T. Noda. [33]

La eficiencia de este modelo se basa en la utilización de un modelo matemático llamado *ARMA (Autoregressive Moving Average)* que reemplaza las convoluciones en el dominio del tiempo, para reducir los tiempos de cálculo.

Para modelar la admitancia $Y_c(\omega)$ con los modelos *ARMA*, la operación de convolución $y_c(t)*v(t)$ se representa por medio de la ecuación [33]:

$$y_c(t) * v(t) = y_{c0} * v(t) + y_{c1}(t) * v(t - \Delta t)$$

Donde y_{c0} es una Matriz constante. Debido a que en el modelo desarrollado por Taku Noda se aproxima la admitancia característica y la matriz de propagación por medio de la transformada Z, el modelo depende del paso de tiempo de la simulación ($\Delta t=1/10.f_{\text{máx}}$), y por consecuencia no se puede aplicar directamente un paso de tiempo arbitrario.

2.2.2.8 Máquinas eléctricas

Hay dos categorías de máquinas eléctricas disponibles en ATPDraw: las máquinas síncronas y las máquinas universales. Dentro de estas categorías, las opciones disponibles se muestran a continuación:

Selección	Nombre del objeto	Icono	tarjeta ATP	Descripción
SM 59/59 + No control	SM		MÁQUINA tipo 59 o 58	Máquina síncrona. Máx. 5 salidas TACS. armadura trifásica.
MI 56	IM56A		MÁQUINA tipo 56	Máquina de inducción con múltiples controles. armadura trifásica.
WI de inducción	UMENTE		UNA MÁQUINA Tipo 3, 4	Máquina universal con entrada de datos de fabricantes.
WI síncrono	UMSYN		UNA MÁQUINA Tipo 1	Máquina universal con entrada de datos de fabricantes.
UM1 síncrono	UM_1		UNA MÁQUINA tipo 1	Sincrónico. Establecer la inicialización en ATP Configuración/Cambiar/UM.
Inducción UM3	UM_3		UNA MÁQUINA tipo 3	Inducción. Establecer la inicialización en ATP Configuración/Cambiar/UM.
Inducción UM4	UM_4		UNA MÁQUINA tipo 4	Inducción. Establecer la inicialización en ATP Configuración/Cambiar/UM.
UM6 Monofásico	UM_6		UNA MÁQUINA tipo 6	Fase única. Establecer la inicialización en ATP Configuración/Cambiar/UM.
UM8 CC	UM_8		UNA MÁQUINA tipo 8	máquina de corriente continua. Establecer la inicialización en ATP Configuración/Cambiar/UM.

Figura 2-23. Modelos disponibles de máquinas eléctricas en ATP. [24]

Máquinas sincrónicas

Los modelos *Synchronous Machine* en ATPDraw tienen las siguientes características/limitaciones:

- Pueden utilizarse con o sin control por medio de TACS.
- Utilizan datos de fabricantes.
- Son modelos sin saturación.
- El modelo no contempla corrientes de Foucault ni bobinas amortiguadoras.
- Tienen acoplada una masa única.

Puede ser representada de dos maneras distintas dependiendo del tipo de estudio que se desea realizar [40]:

1 - Mediante fuentes estáticas AC trifásicas o monofásicas e impedancias RL acopladas. Esta representación es muy sencilla pero no permite representar en detalle el comportamiento de la máquina.

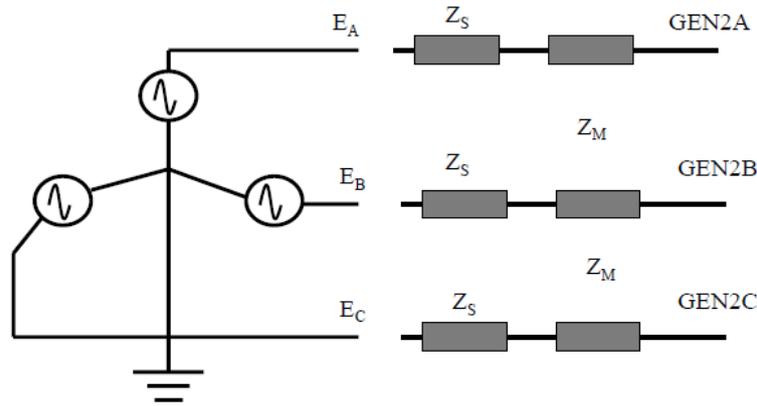


Figura 2-24. Máquina simplificada mediante fuentes estáticas Tipo 14. [40]

2 - Mediante fuentes dinámicas “tipo 59” (es el número del modelo de máquina sincrónica utilizado por ATP para distinguirlo de otros modelos). Esta representación es más compleja y permite estudiar en detalle el comportamiento de la máquina. Consta de una parte eléctrica y otra mecánica mostrada en la Figura 2-25 y Figura 2-26:

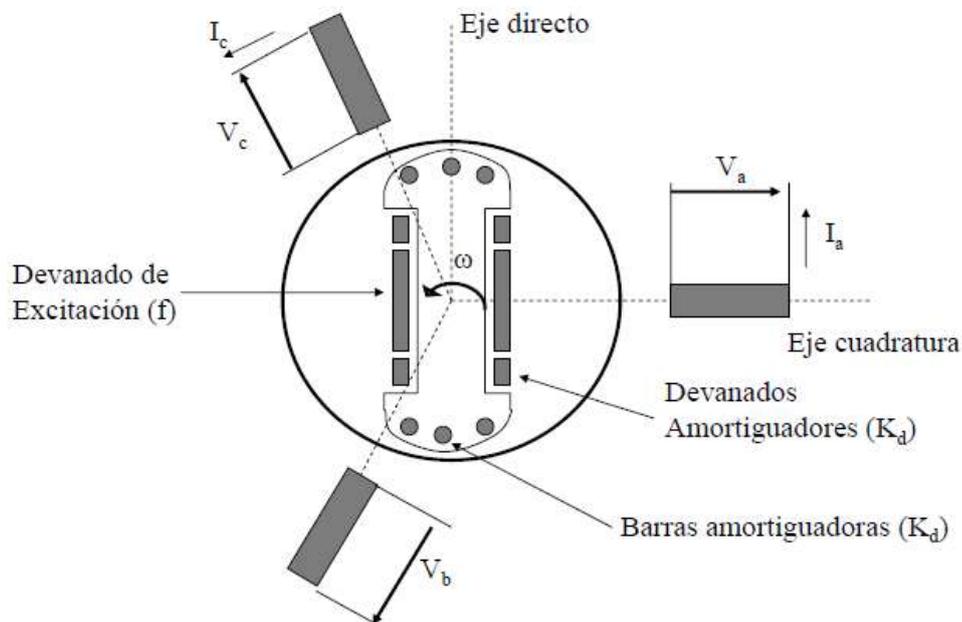


Figura 2-25. Máquina tipo 59: Parte Eléctrica. [40]

Permite contemplar:

- Tres bobinados estáticos para las fases A, B y C.
- Dos devanados en el eje directo, uno de excitación “f” y otro real o hipotético de amortiguación “kd”.
- Dos devanados hipotéticos en el eje de cuadratura, uno para representar los efectos de amortiguamiento producidos por las barras “g”, y otro para representar los efectos de las corrientes parásitas “kg”.

- El bobinado de armadura se asume conectado en estrella, con la posibilidad de conectar una impedancia de neutro.

La parte mecánica del modelo se representa de la siguiente manera:

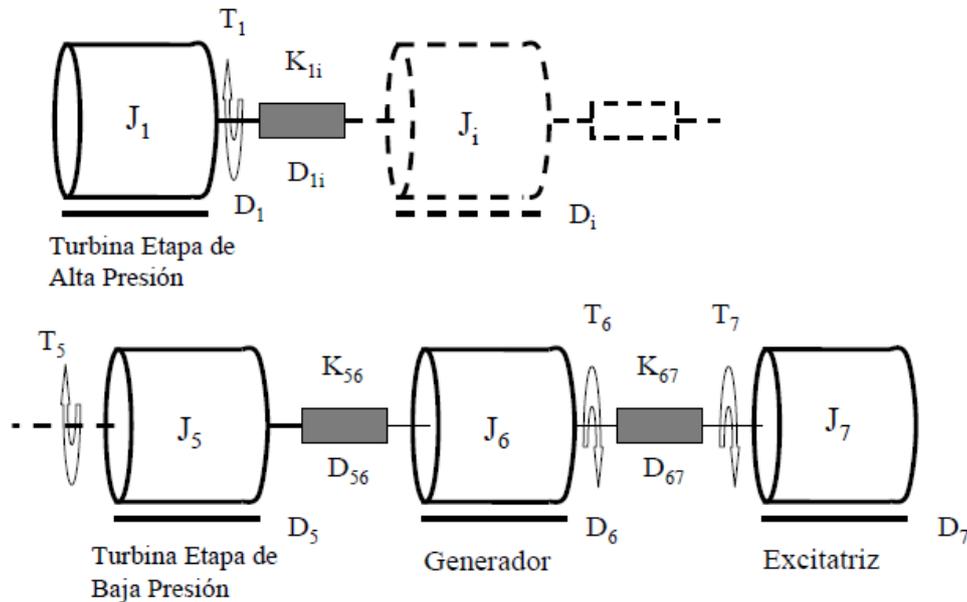


Figura 2-26. Máquina tipo 59: Parte mecánica [40]

Permite contemplar hasta 7 masas para representar las distintas masas de la turbina impulsora, el generador y la excitatriz. Los datos necesarios para configurar la máquina son clasificados en 6 clases:

Datos Clase 1: Datos de estado estable, como la tensión en bornes del motor, la frecuencia en estado estacionario, entre otros. Los datos de inicialización son calculados automáticamente por ATP partir de los datos de estado estable.

Datos Clase 2: Solicitudes especiales. Son datos extras, como la tolerancia, pero que no son obligatorios ingresar.

Datos Clase 3: Valores nominales y parámetros eléctricos. Incluye datos como el número de masas conectadas al eje del generador, números de polos de la máquina, valor de la corriente de campo que produce los valores nominales de tensión de armadura, entre otros. En parámetros eléctricos se ingresan los valores de resistencia de armadura, reactancia de dispersión de armadura, reactancia sincrónica de eje directo, reactancia sincrónica de eje cuadratura, reactancia transitoria de eje directo, reactancia de secuencia cero, entre otros.

Datos Clase 4: Parámetros Mecánicos. Algunos de los datos a ingresar son la cantidad de masas, la fracción de cupla mecánica asociada a cada masa, coeficientes de amortiguamiento, etc.

Datos Clase 5: Solicitud de almacenamiento de parámetros. Con este se configura que datos de salida se guardarán, como la velocidad mecánica sobre las masas respecto a la velocidad sincrónica, la cupla, etc.

Datos Clase 6: Conexión con el módulo TACS. Opcional.

Máquinas universales

El modelo *Universal Machine* de ATPDraw se divide a su vez en los siguientes submodelos diferenciados por “tipos”:

Tipo 1: Máquina Sincrónica, armadura trifásica.

Tipo 2: Máquina Sincrónica, armadura bifásica.

Tipo 3: Máquina de Inducción, armadura trifásica y rotor jaula.

Tipo 4: Máquina de Inducción, armadura y rotor bobinados trifásico.

Tipo 5: Máquina de Inducción, armadura bifásica.

Tipo 6: Máquina de Inducción o Sincrónica, monofásica.

Tipo 7: Máquina de Inducción o Sincrónica bifásica.

Tipo 8, 9, 10, 11 y 12: Máquinas de Corriente Continua.

Algunas de las características y limitaciones del modelo *Universal Machine* son:

- Se permite una inicialización manual y una automática.
- Pueden ingresarse datos brutos de la bobina (parámetros internos) o ingresarse datos de fabricantes en la rutina Windsyn, la cual calcula automáticamente los valores de parámetros necesarios para el modelo.
- La saturación se admite en d, q o en ambos ejes.
- Máximo cinco bobinas de excitación, suma d y eje q.
- Opción de red solo para par mecánico.
- Fuente de par única.

Este modelo está conformado por [40]:

Una parte eléctrica: Se modelan las distintas máquinas a partir de coordenadas d-q-0 (resistencias e inductancias en transformada Park). ATP calcula estos valores y lo convierte a coordenadas a-b-c:

$$[V_{dq0}] = [T]^{-1} \cdot [V_{abc}]$$

$$[I_{dq0}] = [T]^{-1} \cdot [I_{abc}]$$

con T una matriz de transformación. Partiendo de estas expresiones, se obtienen ecuaciones de tensiones, corrientes, flujos y par electromagnético. Y sobre esta base se construyen los circuitos equivalentes de cada eje.

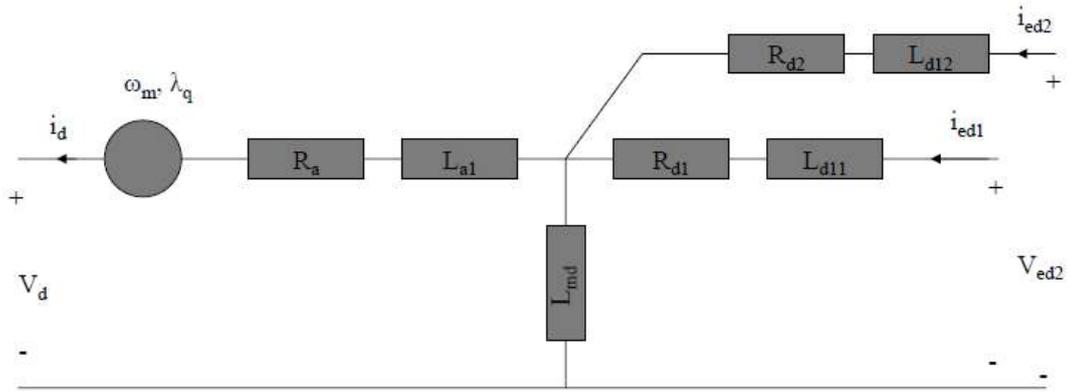


Figura 2-27. Circuito equivalente sobre el eje-d. [40]

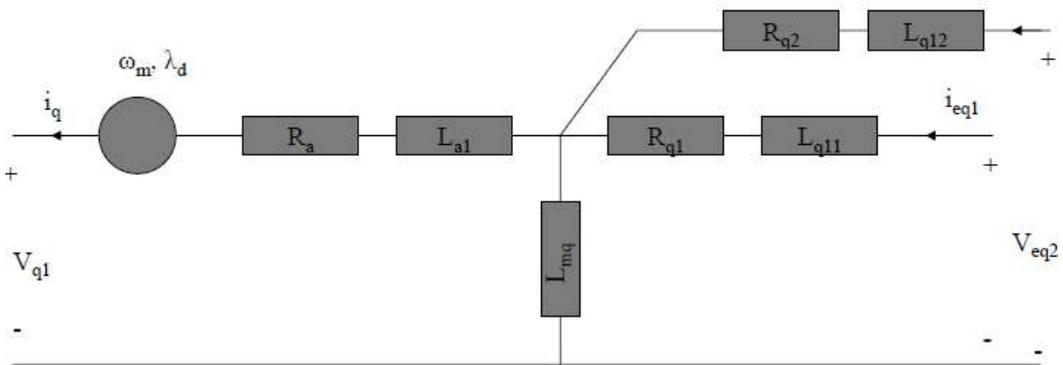


Figura 2-28. Circuito equivalente sobre el eje-q. [40]

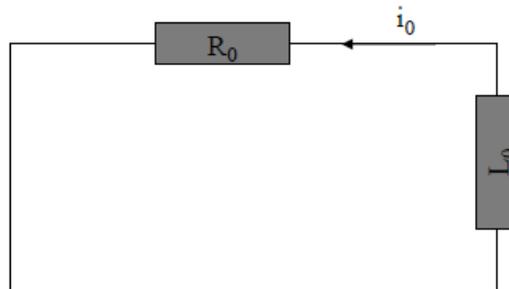


Figura 2-29. Circuito equivalente sobre el eje-0. [40]

Y una parte mecánica: Representada en la Figura 2-30.

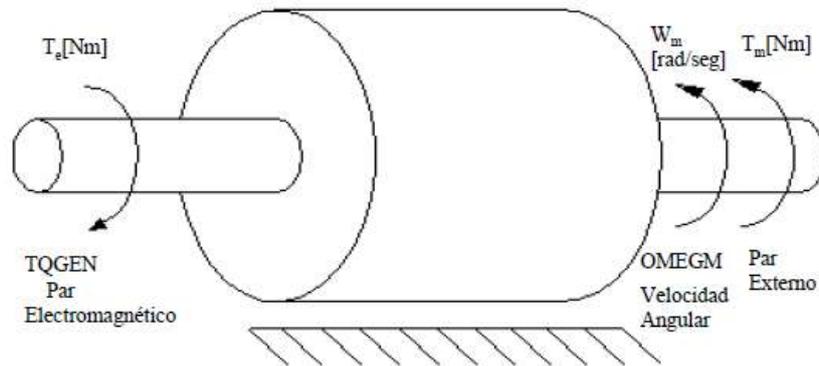


Figura 2-30. Parte mecánica, donde se destaca el par electromagnético, el externo, y la velocidad angular de la máquina. [40]

Y cuya ecuación es:

$$T_m = J \frac{d\omega_m}{dt} + D \cdot \omega_m + T_e$$

Esta ecuación puede ser modelada como una red eléctrica análoga:

$$i_m = C \cdot \frac{dv_m}{dt} + \frac{1}{R} \cdot v_m + i_e$$

Obteniendo las siguientes equivalencias entre los parámetros mecánicos y eléctricos:

$$i_m = T_m$$

$$i_e = T_e$$

$$V_m = \omega_m$$

$$C = J \cdot 10^6 \mu F$$

$$R = \frac{1}{D} \Omega$$

Esto permite controlar los parámetros de la máquina mediante elementos eléctricos disponibles en ATP. El par electromagnético y el externo puede controlarse mediante fuentes de corriente, la velocidad angular puede ser controlada con una fuente de tensión, y los amortiguamientos (fricción) se modelan como una resistencia en serie, y por cada masa acoplada al sistema se agrega un capacitor a tierra. Las equivalencias entre los valores mecánicos y eléctricos se adjuntan a continuación [26]:

- Masa del eje (momento de inercia) - Capacitancia: (1kg/m² <-> 1 Farad)
- Sección del eje (constante de resorte) - Inductancia inversa: (1 Nm/rad <-> 1/Henry)

- Fricción del eje (amortiguación viscosa) - Conductancia: (1 Nm/rad/s \leftrightarrow 1/ohm)
- Velocidad angular - Voltaje: (1 rad/s \leftrightarrow 1 Voltio)
- Par - Corriente: (1 Nm \leftrightarrow 1 Amp)
- Ángulo - Carga:(1 rad \leftrightarrow 1 Coulomb)

La inicialización manual o automática: Para el caso de inicialización automática debe ingresarse el deslizamiento “s”. La inicialización manual es utilizada solo para arranques en vacío, especificando las corrientes en los bobinados para tiempo cero.

Rutina Windsyn: Esta rutina propia de ATP se utiliza para obtener los valores de los parámetros del circuito equivalente del motor de inducción.

Los parámetros de entrada (datos del fabricante) son: Frecuencia f [Hz], tensión línea a línea VL [kV], potencia P [hp], velocidad rpm [rpm], factor de potencia $\cos \phi$ [pu], rendimiento [pu], deslizamiento [pu], corriente de arranque I_{st} [pu], par de arranque T_{st} [pu], par de carga T [pu], par máximo (opcional) T_{max} [pu] y el factor de jaula m [pu]. Además, se supone que la corriente de carga nominal es de 1 pu. La Figura 2-31 muestra el circuito equivalente en el eje “d” de la máquina de inducción soportada por Windsyn:

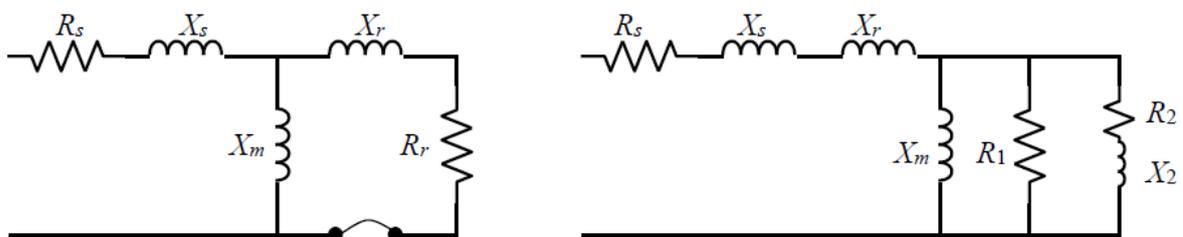


Figura 2-31. Modelos de máquinas de inducción Windsyn en la puesta en marcha. A la izquierda el modelo del motor para un rotor bobinado y jaula simple y a la derecha para un rotor de doble jaula y barra profunda [24]

Para el régimen nominal, las resistencias del rotor se dividen por el deslizamiento “s”. Se adjunta una captura de pantalla del cuadro de dialogo de la rutina, donde se muestran los datos que debe ingresar el usuario y los resultados obtenidos:

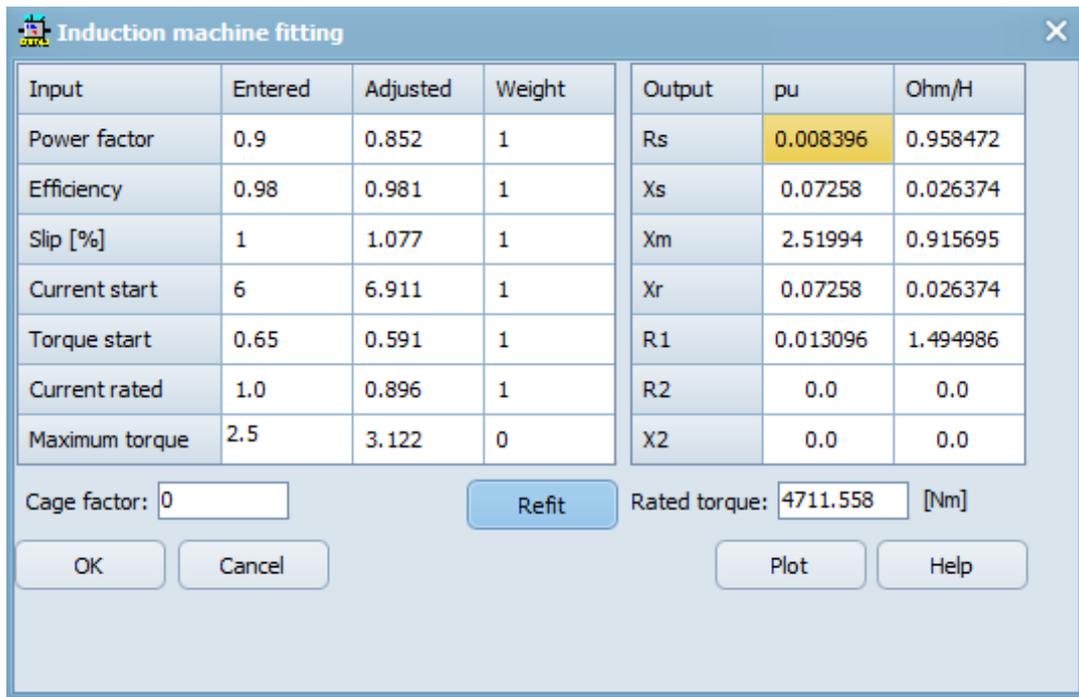


Figura 2-32. Rutina Windsyn. A la izquierda se muestra el “input” que el usuario debe completar con la información del motor de inducción a simular. A la derecha se observa el “output”, siendo los valores de los parámetros del circuito equivalente del motor.

2.2.3 Lenguaje MODELS

En esta sección se presentará brevemente al lenguaje MODELS y como se utiliza en ATP. La idea es mostrar algunos conceptos claves para entender el funcionamiento básico, y en el Capítulo 3 se desarrollará su uso en las mediciones de los valores RMS y en el cálculo de los fasores, para las corrientes y tensiones en nodos seleccionados de la red.

MODELS es una herramienta de ATP disponible en el menú de herramientas del programa, y que permite la inserción del “bloque models” de forma análoga a como se insertan los elementos disponibles para generar el circuito eléctrico. Este bloque le permite al usuario generar código dentro del mismo, y crear un elemento de la red totalmente personalizado.

Para codificar en bloque MODELS se debe insertar el mismo en el circuito (Figura 2-33) y luego clicar en el botón “Edit”. Al hacer esto, se abrirá el editor de texto mostrado en la Figura 2-34.

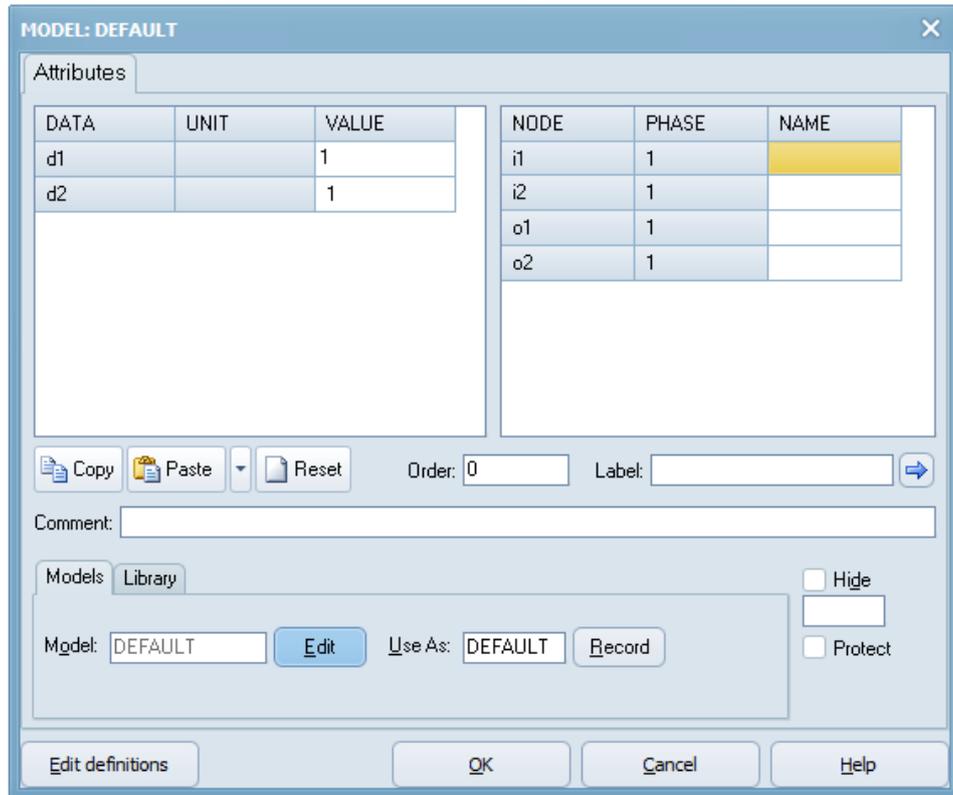


Figura 2-33. Bloque MODELS por default.

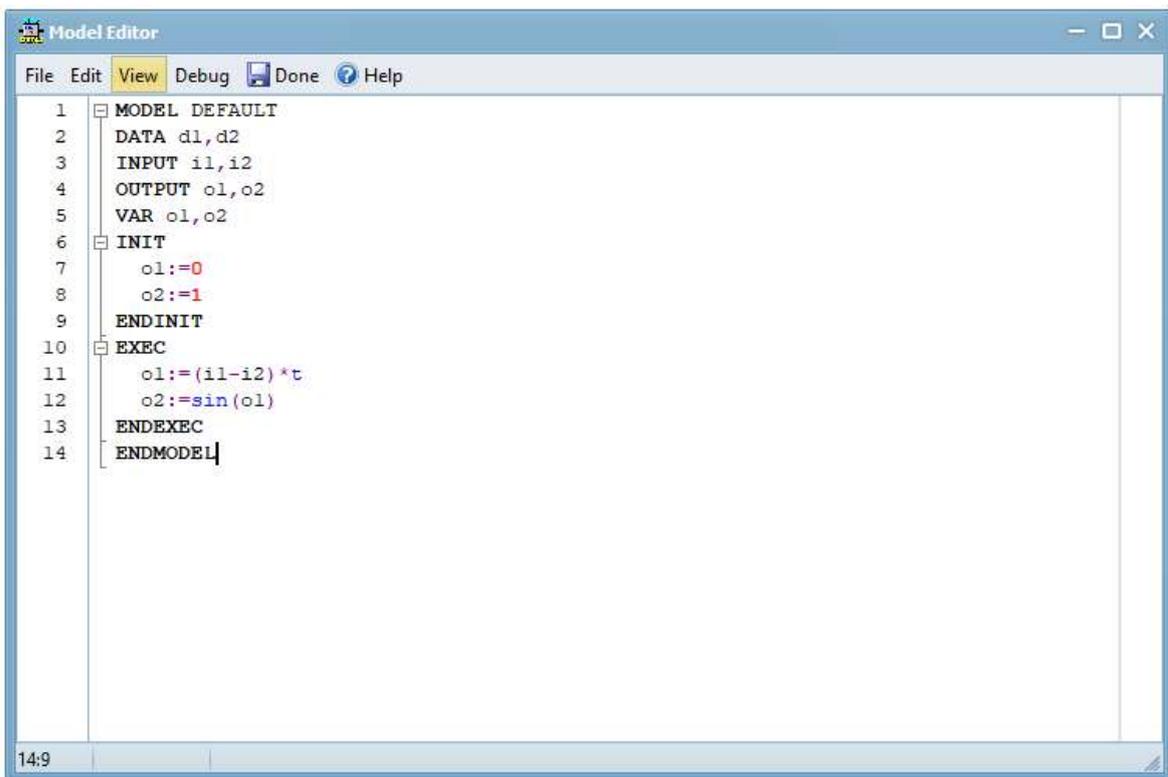


Figura 2-34. Editor de texto en MODELS disponible en ATPdraw.

MODELS permite llevar a cabo un conjunto de operaciones y crear funciones necesarias para la representación de sistemas que varían en el tiempo. Todas aquellas variables que

sean función del tiempo dentro de MODELS se actualizarán al mismo tiempo que en la simulación, es decir en estos casos el tiempo de MODELS y el de la simulación es absoluto. Sin embargo, también puede programarse pasos de tiempo definidos localmente en MODELS con la capacidad de actualizarse varias veces en lo que transcurre un paso de tiempo en la simulación, debiendo definirse alguno de los métodos para extrapolar los valores correspondientes.

MODELS gestiona automáticamente el almacenamiento y la recuperación de los valores históricos de las variables de un modelo según sea necesario durante la ejecución de la simulación. Todas las funciones del lenguaje basadas en el tiempo tienen acceso directo a los valores pasados de una variable y son utilizados para los cálculos, sin imponer al usuario la tarea de especificar cómo se almacena y se accede a esa información.

Al codificar en MODELS deben definirse los siguientes valores/variables:

CONST: son valores constantes y que el usuario define para utilizarlos en los cálculos. Hay un conjunto de constantes que tienen nombres y valores predefinidos y se pueden utilizar sin necesidad de declarar, por ejemplo, el número π .

DATA: Los datos se utilizan para mantener valores que se definen fuera del modelo y que no varían durante la simulación. Los nombres de los elementos DATA son visibles dentro del modelo donde se declaran, y se asignan en las sentencias USE. Un ejemplo típico es definir a la frecuencia del sistema como un dato, de esta manera el MODELS leerá la frecuencia de la simulación y la utilizará en los cálculos, manteniéndose invariante.

INPUT: Su valor se asigna en las sentencias USE. Al declarar un input lo que se hace es leer valores de la simulación para usarlo en MODELS. Por ejemplo, se puede declarar como INPUT la tensión en un nodo del sistema.

VAR: Se utiliza para asignar variables. Su valor se puede asignar en cualquier declaración dentro del modelo. Una vez asignado un valor, una variable mantiene ese valor durante la simulación, hasta que se le asigne uno nuevo.

OUTPUT: Son aquellas variables que se declararan como salida del modelo. Estas salidas pueden ser leídas por otros elementos del circuito.

Una vez definidos estos valores, se prosigue con la definición de los *Procedures* (procedimientos) que contienen las sentencias en las que se utilizan y asignan estos valores. También se pueden usar *Statements* (declaraciones) que permiten representar cómo se calcula y asigna un valor en el modelo. Se pueden usar varios tipos de declaraciones en un procedimiento para asignar valores a las variables del modelo, también para controlar el flujo algorítmico de un procedimiento, para definir y usar instancias de otros modelos y para llamar a los procedimientos de un modelo.

Si el usuario necesita calcular a partir de las variables de su modelo derivadas, integrales, o incluso trabajar en el dominio de Laplace, lo puede hacer definiendo las respectivas

expresiones (Ver *Dubé* [37] para profundizar en estos conceptos). Por ejemplo para realizar una integral en MODELS se utiliza la expresión:

integral(x):=expresión_a_integrar {dmin:expresion, dmax:expresion}

donde d_{\min} y d_{\max} representan los límites de la integral. Para el caso de derivaciones se utiliza la expresión *deriv()*. Otra expresión interesante es la de *delay()*. Esta es una función de retraso, es decir permite asignar un espacio de almacenamiento para los valores pasados de una variable. Para hacer uso de esta expresión debe designarse cuantos pasos de tiempo previos serán almacenados. Resulta útil en casos como al realizar una derivada, o integrar en un período pasado. Además, se pueden definir *Submodels* (submodelos) y *functions* (funciones) para realizar parte del cálculo de estos valores. El uso de cada modelo y submodelo se puede personalizar para cada instancia de uso, mediante directivas de simulación especificadas en la definición del modelo y en las declaraciones USE. [36] [37]

2.3 Conclusión

En este capítulo se han mostrado las bases del modelado computacional en ATP de algunos de los elementos típicos presentes en las redes eléctricas. A partir de esto, se seleccionaron modelos que tienen un buen desempeño en las perturbaciones que serán simuladas en este trabajo, es decir en recrear eventos de tensión a partir de fallas eléctricas. La selección de los elementos de la red se desarrolla a continuación.

Transformador. La red eléctrica seleccionada para utilizar en este trabajo, es decir el alimentador de prueba de 34 nodos, posee dos transformadores trifásicos de potencia. La única información disponible son las Potencias, tensiones nominales, conexionado y R% y X%. En base a la bibliografía citada, la opción seleccionada es el modelo de Transformador Híbrido. Este modelo tiene disponible los conexionados Dy e YY (conexiones de los transformadores a simular) y cuenta con la posibilidad de realizar las simulaciones a partir de valores típicos de transformadores, información disponible en ATP, con la posibilidad de realizarle las modificaciones que el usuario considere necesarias. La idea no es realizar un modelado extremadamente detallado del transformador ya que el fin de la simulación no es representar fenómenos transitorios de frecuencias elevadas, sino analizar como los transformadores afectan a la firma del evento de tensión generado en otros lugares de la red.

Líneas eléctricas. Los documentos disponibles en IEEE del alimentador de prueba de 34 nodos proporcionan las matrices de Z y B de cada una de las líneas de la red. Además, cada segmento de la línea individualmente no supera los 15 km de longitud, y finalmente no se busca analizar perturbaciones en donde las altas frecuencias, o el análisis de ondas viajes tengan especial relevancia. Por este motivo se decidió utilizar el modelo PI nominal.

Las *cargas eléctricas*, incluyendo los dos bancos de capacitores serán modelados con los elementos disponibles en la librería de ATP (R-L-C).

Los reguladores de tensión serán modelados con transformadores monofásicos cuya relación de transformación corresponda con el nivel del paso del regulador, y cuyo dato es proporcionado por los documentos de la IEEE. Con este modelado se buscará analizar únicamente si los perfiles de tensión se encuentran en valores aceptables comparados con los indicados por la IEEE.

En el capítulo 4 se analizará con más detalle la simulación de cada uno de estos elementos, y también se desarrollará la metodología para el modelado de las fallas eléctricas en la red.

2.4 Bibliografía

- [24] ATPDRAW version 7.3, User Manual, 2021.
- [25] Juan A. Martinez-Velasco (2020). *Transient Analysis of Power Systems, A Practical Approach*. JohnWiley & Sons Ltd
- [26] Página web de ATPDraw – sección Help. Recuperado de: <https://www.atpdraw.net/help7/>
- [27] Meyer, W. S., & Liu, T.-h. (1994). *Electro-Magnetic Transients Program (EMTP) Theory Book*. Portland: American EMTP user Group.
- [28] C. Q. Su, (2013). *Electromagnetic Transients in Transformer and Rotating Machine Windings*. USA: IGI Global.
- [29] Can/Am EMTP Users Group, ATP Rule Book, 1987.
- [30] J.A. Martinez, B.A. Mork, (2005). *Transformer modeling for low- and mid-frequency transients- a review*, IEEE Trans. Power Deliv. 20 (2) (2005) 1625–1632.
- [31] Hans K. Høidalena, Bruce A. Morkb, Francisco Gonzalezb, Dmitry Ishchenkob, Nicola Chiesaa (2009). *Implementation and verification of the Hybrid Transformer model in ATPDraw*. Electric Power Systems Research 79 pg. 454-459.
- [32] B.A. Mork, F. Gonzalez, D. Ishchenko, D.L. Stuehm, J. Mitra (2007). *Hybrid transformer model for transient simulation*. Part I. Development and parameters, IEEE Trans. Power Deliv. 22, pg. 248–255.
- [33] Luis H-Restrepo, Gladys Caicedo Delgado, Ferley Castro-Aranda, (2008). *Transmission line models for electromagnetic transients in power systems*. Revista Energía y Computación Vol. 16 No. 1 Junio de 2008 p. 21- 32.
- [34] Eiichi Haginomori, Tadashi Koshiduka, Junichi Arai, Hisatoshi Ikeda, (2016). *Power System Transient Analysis, Theory and Practice using Simulation Programs (ATP-EMTP)*. JohnWiley & Sons Ltd.
- [35] Neville Watson, Jos Arrillaga, (2018). *Power Systems Electromagnetic Transients Simulation-Institution of Engineering and Technology*. Published by The Institution of Engineering and Technology, London, United Kingdom.
- [36] Laurent Dubé (1996). *User guide to Models in ATP*.

[37] Laurent Dubé (1996). *Language Manual to Models in ATP*.

[38] A. Ametani. (2015). *Numerical Analysis of Power System Transients and Dynamics*. London, UK: The Institution of Engineering and Technology.

[39] Swee Peng Ang. (2010). *Ferroresonance simulation studies of transmission systems*. Ph.D. dissertation, School of Elect. and Electron. Eng., Univ. of Manchester, Manchester, UK.

[40] Alejandro Jurado, (2012). ATP/EMTP, Aplicaciones del ATPDraw a la calidad de la Energía Eléctrica. Curso dictado en la Universidad Nacional de Mar del Plata en el año 2012.

CAPÍTULO 3

ALGORITMOS

CÁLCULOS Y AUTOMATIZACIÓN



3 Cálculos y automatización

Resumen

En este capítulo se describirán los algoritmos desarrollados y utilizados durante el presente trabajo final con el fin de automatizar las simulaciones, generar registros y por último analizar los resultados obtenidos. Los algoritmos fueron implementados en lenguaje Python, y también se hizo uso del lenguaje y herramienta *MODELS* propio de *ATPDraw*. Para automatizar las ordenes de ejecución de los algoritmos y de simulación se utilizaron archivos de procesamiento de lotes *Batch*.

Los códigos desarrollados en este trabajo se pueden categorizar según su aplicación:

Como una herramienta para parametrizar la red eléctrica simulada, permitiendo cambiar sus valores desde un archivo de texto plano externo al software *ATP*. Para ello se desarrollaron los *Algoritmos de Inicialización y de Convergencia*.

Para agilizar el proceso de simulaciones de fallas y obtener un muestreo de datos de tamaño considerable de una manera automatizada, se desarrolló el *Algoritmo de Fallas*.

Para implementar el Algoritmo *ASA* y *eASA* se desarrolló un código que permita realizar los cálculos pertinentes a partir de las muestras obtenidas de las simulaciones. Esto fue posible a partir del *Algoritmo de clasificación de eventos*.

Como códigos complementarios, se encuentran aquellos utilizados para realizar cálculos de valores RMS y obtener los fasores de tensiones y corrientes desde el software *ATP* mediante el uso de *MODELS*; generar diferentes tipos de gráficos desde Python, y ejecutar automáticamente los algoritmos previamente nombrados utilizando archivos *Batch*.

3.1 Introducción

En una primera instancia, se desarrollaron algoritmos en lenguaje de programación Python con el objetivo de automatizar la carga de datos de la red, y modificarlos de una manera rápida, Algoritmo de inicialización (Ver Sección 3.2) y Algoritmo de convergencia (Ver Sección 3.3).

Una vez creada la red eléctrica, para este trabajo el alimentador de pruebas de 34 nodos, con la totalidad de los elementos de red ya definidos, se desarrolló un algoritmo que permitiera simular de manera iterativa y continua fallas en distintos puntos de la red con valores de impedancia de falla variables, y que almacene los resultados de cada falla en un archivo de texto delimitado por comas (*csv*). Una de las ventajas de utilizar archivos de formato “*csv*” es que pueden ser leídos desde cualquier editor de texto plano, y el formato resulta práctico para ser leído desde Python (Ver Sección 3.4).

Finalmente, se desarrolló un código de programación para evaluar el desempeño de algoritmos de detección de eventos de tensión, realizando los diversos cálculos necesarios como el cálculo de componentes simétricas de las señales medidas, y posteriormente aplicar los algoritmos ASA [42] y eASA para la clasificación de los eventos de tensión (Ver Sección 3.5).

Para el cálculo de los valores RMS y obtención de fasores de las tensiones y corrientes, se utilizó la herramienta MODELS de ATP (Ver 3.4.1). Estos valores se guardan con un formato particular en los archivos “*.lis” que genera ATP, y posteriormente son leídos y almacenados por el Algoritmo de Fallas (Sección 3.4).

Para poder visualizar en gráficos los resultados obtenidos, se creó un código que lee los valores de RMS y sus fasores, y genera gráficos fasoriales, de Tensión versus Z_{falla} y Salto de fase versus Z_{falla} .

En todos los casos, se utilizaron como herramientas el software *ATP-ATPDraw* y códigos de elaboración propia desarrollados en lenguaje de programación Python. Para los cálculos en complejos se utilizó el módulo propio de Python *cmath* [44] y para generar los gráficos se utilizó la librería compatible con Python y Open Source *Matplotlib* [43]

En total, fueron requeridas más de 120 horas de simulación, teniendo en cuenta las horas requeridas para simular en *ATPDraw* y el tiempo de ejecución de los códigos para los cálculos en cada instancia.

Las especificaciones del hardware de la computadora utilizada para la ejecución de todos los algoritmos se adjuntan a continuación:

Processor	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
Installed RAM	8.00 GB (7.88 GB usable)
System type	64-bit operating system, x64-based processor

En el anexo se adjuntan los códigos desarrollados y mencionados a lo largo del presente capítulo.

3.2 Algoritmo de inicialización

Este algoritmo se ejecuta una vez creado el circuito eléctrico en ATPDraw con la totalidad de los elementos eléctricos correspondientes, y guardado el proyecto en su extensión nativa “*.acp” (ver Capítulo 2, sección 1.1.4). El fin del desarrollo de este algoritmo es que el usuario pueda generar el circuito eléctrico, pero no tenga que ingresar los valores cargas manualmente (valores de R, C, L), brindando la posibilidad de completar esta información con una carga masiva de datos de una forma externa a ATP. Si bien en la red simulada de 34

nodos no hay una cantidad excesiva de cargas eléctricas, y por lo tanto el proceso puede realizarse manualmente, este algoritmo fue pensado para su uso en redes eléctricas más extensas, por ejemplo, los alimentadores de prueba de 123 nodos o el de 8500 nodos, por citar algunos de ellos.

A continuación se desarrolla la información necesaria para entender el funcionamiento del algoritmo y los pasos que debe seguir el usuario para el correcto funcionamiento.

Al crear el circuito eléctrico en ATP insertando los elementos eléctricos que lo componen (Ver Figura 3-1), por ejemplo, al insertar una carga eléctrica trifásica se abre automáticamente un cuadro de diálogo como el mostrado en la Figura 3-2. Para el correcto funcionamiento del algoritmo se debe agregar en el cuadro de dialogo “*Comment*” un nombre (*ID*) para referenciar y distinguir esa carga de las demás (ver Figura 3-2). Este ID es utilizado por el algoritmo para identificar a la carga en el archivo “*.atp” generado por el software *ATP*, y reemplazar sus valores que figuran en la columna *attributes* en (Figura 3-2) por aquellos que el usuario defina en un archivo “csv” externo al software *ATP* (Tabla 3-1).

Esto permite tener la libertad de insertar una gran cantidad de cargas sin tener que completar los atributos de cada uno de ellas, ya que estos datos serán cargados posteriormente por el algoritmo. Con esto se busca ahorrar tiempo y evitar errores en la carga de datos. También permite manipular los valores del circuito de una manera externa, sin tener que recurrir a la interface gráfica *ATPDraw*.

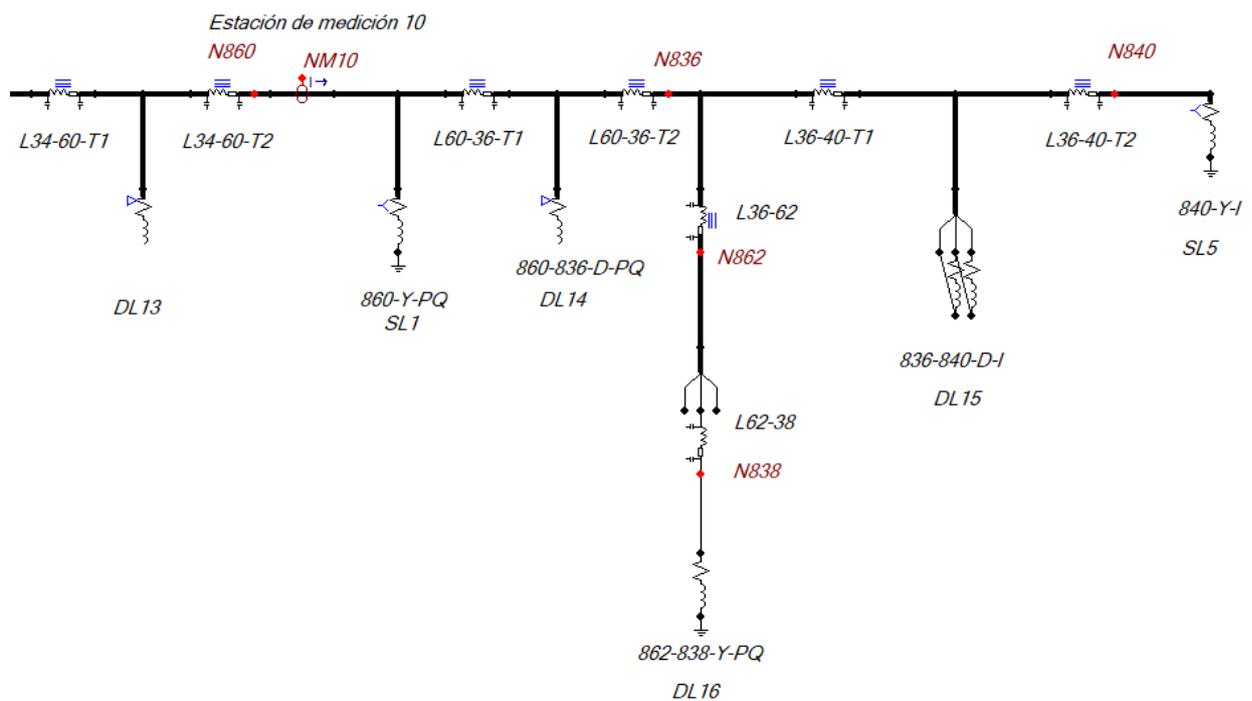


Figura 3-1. Circuito eléctrico generado en ATPDraw.

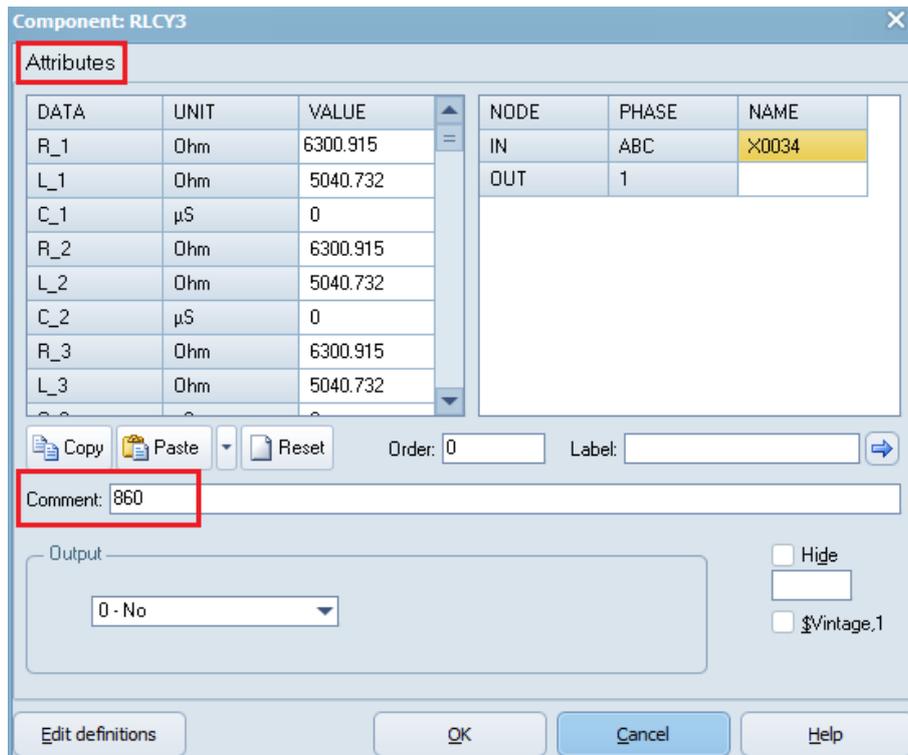


Figura 3-2. Información de la carga trifásica RLCY disponible en ATPDraw. Se recuadra la sección *Atributes*, y cuyos valores de la tercer columna serán reemplazados por el algoritmo de inicialización. También se recuadra *Comment* donde debe comentarse el nombre para identificar a la carga.

Una vez creado y simulado el circuito eléctrico, en la carpeta donde está guardado el proyecto, se crea automáticamente por *ATP* un archivo “*.atp” (Figura 3-3) que contiene toda la información del circuito eléctrico, incluidos los valores de las cargas que se desean modificar. Este archivo “*.atp” es el que utiliza el *solver* de *ATP* para realizar los cálculos durante la simulación. Una ventaja de trabajar con este archivo es que se puede abrir y modificar desde editores de texto plano, resultando ser más práctico y ágil que al trabajar con la interfaz gráfica de *ATP*.

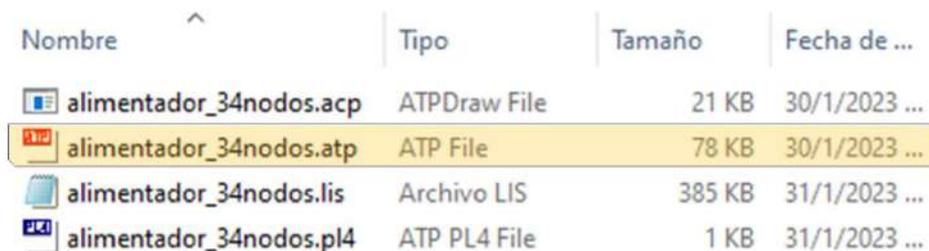


Figura 3-3. Archivos creados por ATPDraw una vez simulado el circuito eléctrico.

El algoritmo fue desarrollado para que abra el archivo “*.atp”, y ubique la carga indicada para editar sus datos. En la Figura 3-4 se observa la arquitectura que utiliza *ATP* para guardar la información de la carga ejemplificada en el archivo “*.atp”, cuyo comentario ingresado

previamente al insertar la carga en ATPDraw (Figura 3-2) figura como “C 860”, siendo “C” la letra reservada por ATP para ingresar comentarios en el código.

Una vez ubicada la carga, se reemplazan los valores de R, L, C por los valores ingresados en el archivo csv externo. Es importante notar que la información que reemplaza el algoritmo debe respetar un cierto formato, incluyendo separaciones y espacios, el “.” como separador decimal, entre otros.

alimentador_34nodos: Bloc de notas

Archivo Edición Formato Ver Ayuda

X0006C		7933.589251	4363.474088	0
C 846848_b				
X0007B		7312.938462	3497.492308	0
C 834860				
X0033AX0033B		31000.5	15500.25	0
X0033BX0033C		24800.4	12400.2	0
X0033CX0033A		4509.163636	2254.581818	0
C 860	Comentario ingresado en Comment	Valores de R [ohm]	e Impedancia Inductiva XL [ohm]	Valores de capacitancia [μ S]
X0034A		6300.915	5040.732	0
X0034B		6300.915	5040.732	0
X0034C		6300.915	5040.732	0
C 860836				
X0035AX0035B		16533.6	8266.8	0
X0035BX0035C		45588.97059	27353.38235	0
X0035CX0035A		11583.81673	6067.713523	0

Figura 3-4. Formato de tarjeta “.atp” para el caso de una carga trifásica. Se observa “C 860” que es el comentario ingresado previamente en la carga desde ATPDraw, junto a los valores de resistencia, inductancia y capacitancia de la carga.

ATP almacena la totalidad de los datos de los elementos eléctricos del circuito con un formato similar. Como, por ejemplo, las matrices de impedancias de transformadores, líneas eléctricas, máquinas eléctricas, información de fuentes, interruptores, etc. Se debe tener en cuenta que las rutinas de ATP como, por ejemplo, la LCC para calcular las impedancias de las líneas eléctricas crean otros archivos además del archivo “*.atp” que luego son llamadas por el *so/ver* de ATP. Estos archivos no son leídos por el algoritmo.

Para la importación de datos de las cargas eléctricas, el usuario debe guardar la información de aquellas cargas que desee modificar en un archivo “*.csv” que respete el formato mostrado en la Tabla 3-1. El diagrama de flujo correspondiente a este algoritmo se puede observar en la Figura 3-5.

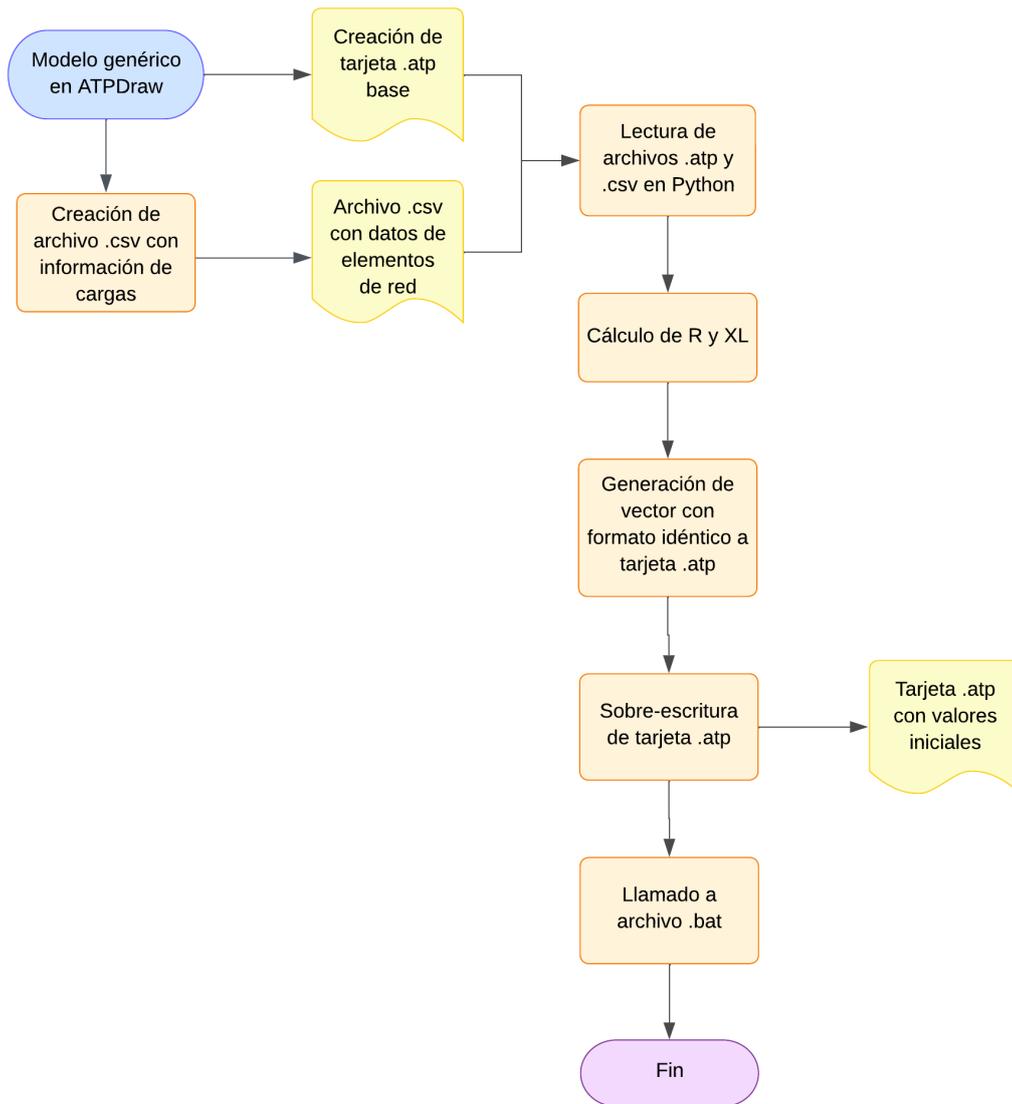


Figura 3-5. Diagrama de flujo de algoritmo de inicialización.

Para el caso particular de la red de 34 nodos seleccionada, se calcularon los valores de resistencia y reactancia a partir de los datos brindados por la IEEE [41]: tensión nominal RMS, potencia activa $P_{nominal}$ y reactiva $Q_{nominal}$ en [kW] y [kVAr] respectivamente.

$$\cos \varphi = \cos\left(\text{atan}\left(\frac{Q_{nominal}}{P_{nominal}}\right)\right)$$

Ecuación 3-1

$$\sin \varphi = \sin\left(\text{atan}\left(\frac{Q_{nominal}}{P_{nominal}}\right)\right)$$

Ecuación 3-2

$$Z_{nominal} = \frac{(V_{f\ nominal})^2 * \cos \varphi}{P_{nominal}}$$

Ecuación 3-3

Donde $V_{f-nominal}$ será igual a la tensión de línea (V_{RMS}) en el caso de carga trifásica en triángulo o, en caso de ser la carga trifásica en estrella, igual a $V_{rms}/\sqrt{3}$. Esto se asigna automáticamente al definir la naturaleza de la carga en el “.csv” de la Tabla 3-1.

Los valores de resistencia y reactancia son:

$$R_{nominal} = Z_{nominal} * \cos \varphi$$

Ecuación 3-4

$$X_{nominal} = Z_{nominal} * \sin \varphi$$

Ecuación 3-5

El archivo de texto delimitado por comas utilizado como base de datos para calcular los valores de las cargas eléctricas de la red (“formato_importacion.csv”) contiene la información listada a continuación y sigue la estructura que se ejemplifica en la Tabla 3-1:

- ID: ID de la carga, que coincide con el ID ingresado en ATP Draw. Como se indicó previamente, el ID se define en el cuadro de dialogo “*Comment*” al insertar la carga (Figura 3-2)
- Tipo: “m” o “t” si es monofásica o trifásica.
- Característica (Caract.): Esta línea debe dejarse como “puntual”.
- Conexión (Conex.): si es estrella “y” o triángulo “d”. Esto influye en los cálculos.
- Modelado: si es una carga de potencia constante “p”, corriente constante “i” o impedancia constante “z”. Hasta el momento, esta información no afecta a la importación de los datos, pero sí se utiliza en el algoritmo de convergencia desarrollado posteriormente (Ver Capítulo 3.3).
- Valores de potencia activa y reactiva nominales en cada una de las tres fases.
- Valor de tensión nominal de línea V_{RMS} .
- Convergencia: Si es una carga que luego será modificada por el algoritmo de convergencia. (s/n) (Ver Capítulo 3.3)

Tabla 3-1. Ejemplo de formato de archivo delimitado por comas utilizado en el algoritmo de inicialización. Los valores mostrados no son los utilizados por las simulaciones.

ID	Tipo	Caract.	Conex	Mod	P[kW] Fase A	Q[KVar] Fase A	P[kW] Fase B	Q[KVar] Fase B	P[kW] Fase C	Q[KVar] Fase C	Vrms [V] línea	Convergencia
1	m	puntual	y	p	10	7					24900	s
2	t	puntual	y	p	9	7	7	4	3	2	13200	n
3	m	puntual	y	p			5	4			380	s
4	t	puntual	d	p	7	3	5	3	9	4	380	s

El código de programación utilizado se encuentra en el apartado 6.1.1 del anexo. En las últimas líneas del código se llama al archivo de procesamiento por lotes de extensión “*.bat” con la finalidad de dar la orden para ejecutar el archivo “*.atp” y simular la red con los datos cargados. El archivo es el siguiente:

```
“inicio.bat”
start "C:/ATP_soft/" circuito.atp
wscript "C:\ATP_soft\comando.vbs"
pause
```

El comando de extensión “*.vbs” utilizado en el archivo “*.bat” es una herramienta auxiliar que permite dar órdenes a la computadora para ejecutar, cerrar o actualizar una pestaña, entre otras utilidades sin la necesidad de utilizar el teclado. En este caso, el contenido del archivo de texto mencionado en el batch es el siguiente:

```
“commando.vbs”
Set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.AppActivate "circuito.atp"
WshShell.SendKeys "{F2}"
```

La finalidad del comando anterior es la de dar la orden de iniciar la simulación en *ATP* ejecutando el archivo “circuito.atp” apretando la tecla “F2”. En este ejemplo resulta trivial, pero para el algoritmo de fallas donde se deben abrir y cerrar reiteradas veces archivos de Python y ejecutar el circuito eléctrico, y esto repetirse para cada una de las fallas, la aplicación del archivo *batch* resulta clave para automatizar la aplicación del algoritmo.

3.3 Algoritmo de convergencia

Este algoritmo surge a partir de notar que las redes eléctricas de la IEEE disponibles están divididas en tres grupos: Las de potencia constante (P), corriente constante (I), e impedancia constante (Z), informando como es el comportamiento y naturaleza de la carga [41]

Los datos de las cargas eléctricas entregados por la IEEE están definidos a partir de sus valores de potencia activa P y su potencia reactiva Q, y la tensión nominal RMS. Si la carga es de impedancia constante, los valores de resistencia y reactancia son calculados a partir de las potencias P y Q y el valor de tensión nominal RMS.

Sin embargo, para los casos de cargas con naturaleza de potencia constante y corriente constante, el caso cambia debido a que, si el valor de tensión que alimenta a la carga no es el nominal, lo cual es común en estas redes de distribución donde los perfiles de voltaje varían, la potencia consumida por la carga dista del valor entregado por la IEEE.

Para solventar estas diferencias se propuso desarrollar un algoritmo en Python que permita un cálculo iterativo de los parámetros de las cargas involucradas, R y X_L en este caso, hasta

alcanzar los valores de potencia o corriente deseados, con un margen de error que el usuario puede definir. Para este trabajo, al analizar los resultados de las simulaciones se concluyó que sin tener que modificar los parámetros de las cargas, los resultados obtenidos eran aceptables, con márgenes de error menores al 5% en prácticamente la totalidad de los sitios de medición. Debido a la gran similitud de los valores obtenidos por la simulación y los de referencia, la aplicación de este algoritmo solo fue probado en la carga correspondiente al nodo 890 por ser la que mayor distancia tenía del valor de referencia. Se puede observar el código detallado en el apartado 6.1.2 en el anexo.

La iteración del algoritmo hasta alcanzar el margen de error establecido, de 5% en este caso, se realizó manualmente, dado que sólo requirió de cinco iteraciones para alcanzar la convergencia. Para ingresar los datos de la carga que se quieran modificar, se utiliza la misma plantilla que para el Algoritmo de Inicialización (Tabla 3-1). En el Capítulo 4 se analizarán los resultados obtenidos en régimen estacionario.

Se quiere dejar en claro que este algoritmo no está basado en algoritmos específicos para resolver flujos de potencia. Simplemente lee las tensiones RMS de las cargas seleccionadas del archivo “*.lis” generado por ATP (Figura 3-3), y con la información de la impedancia de la carga, calcula la potencia/corriente consumida. En caso de no estar dentro del error definido, vuelve a recalcular los valores de impedancia necesarios. Tampoco es un algoritmo que provoque cambios durante la simulación, por el contrario, el algoritmo propuesto utiliza los resultados de una primera simulación para realizar los cálculos, y luego modifica los valores correspondientes para volver a simular, repitiendo esto n veces.

Carga de potencia constante: Luego de la primera simulación del circuito eléctrico, el algoritmo lee las tensiones RMS “ $V_{f-iteración\ previa}$ ” resultantes de la simulación, y utilizando el valor de impedancia de la simulación previa ($Z_{iteración\ previa}$), calcula los valores de potencia P y Q . El coseno ϕ se mantiene invariable en estos cálculos y se calcula a partir de los datos entregados por la IEEE (Ecuación 3-1).

$$P = \frac{(V_{f-iteración\ previa})^2 * \cos \phi}{Z_{iteración\ previa}}$$

Ecuación 3-6.

$$Q = \frac{(V_{f-iteración\ previa})^2 * \sin \phi}{Z_{iteración\ previa}}$$

Ecuación 3-7.

Si la potencia obtenida tiene un error mayor al admisible, se recalculan los valores de impedancia aplicando la Ecuación 3-8, y luego se aplica la Ecuación 3-4 y la Ecuación 3-5.

$$Z = \frac{(V_{f-iteración\ previa})^2 * \cos \varphi}{P_{nominal}}$$

Ecuación 3-8

Finalmente, se reemplazan en el archivo “*.atp” las impedancias calculadas y se vuelve a simular el circuito. Esto se repite hasta alcanzar una convergencia en función de los porcentajes de error admisibles definidos.

Carga de corriente constante: Para el caso de cargas de corriente constante, el primer paso es calcular la corriente nominal de cada carga que se desea obtener. Para ello, el algoritmo utiliza la Ecuación 3-9 y Ecuación 3-10. La potencia $S_{nominal}$ y la tensión nominal son los datos entregados por la IEEE.

$$S_{nominal} = \sqrt{(P_{nominal})^2 + (Q_{nominal})^2}$$

Ecuación 3-9.

$$I_{nominal} = \frac{S_{nominal}}{V_{nominal}}$$

Ecuación 3-10.

Luego de la primera simulación, se leen los valores de tensión de fase RMS obtenidos ($V_{f-iteración\ previa}$), y se calcula la corriente de fase obtenida (Ecuación 3-11), donde $Z_{iteración\ previa}$ es la impedancia utilizada en la simulación previa.

$$I_{fase} = \frac{V_{f-iteración\ previa}}{Z_{iteración\ previa}}$$

Ecuación 3-11.

Si el valor de corriente de fase obtenido tiene un error mayor al admisible, se recalcula Z utilizando la Ecuación 3-12:

$$Z = \frac{V_{f-iteración\ previa}}{I_{nominal}}$$

Ecuación 3-12.

Al igual que para la carga de potencia constante, se considera que el coseno ϕ se mantiene invariable y se calcula a partir de los datos entregados por la IEEE. Luego se aplica la Ecuación 3-4 y la Ecuación 3-5, donde Z es la obtenida en la Ecuación 3-12 para obtener los valores resistencia y reactancia. Estos valores obtenidos son reemplazados en el archivo

“*.atp” y se ejecuta nuevamente la simulación del circuito eléctrico. Esto se repite hasta alcanzar una convergencia en función de los porcentajes de error admisibles definidos.

Se debe tener en cuenta que este algoritmo no fue probado con varias cargas a converger en simultáneo, es decir no se analizó como es su desempeño cuando quiere obtenerse una convergencia en varios puntos de la red a la vez. Esto es importante de remarcar porque al ser un proceso iterativo y la red ser un conjunto dinámico, es posible que, al variar las impedancias de varias cargas eléctricas en simultáneo, el circuito evolucione satisfactoriamente en algunas zonas, pero se aleje de la solución buscada en otras, sin poder lograr una convergencia. Esto podría mejorarse limitando la tasa de variación de las impedancias, obteniendo una evolución del sistema más lenta y estable, con un posible aumento en cantidad de iteraciones. Este análisis en profundidad excede a este trabajo.

El esquema del algoritmo se adjunta a continuación:

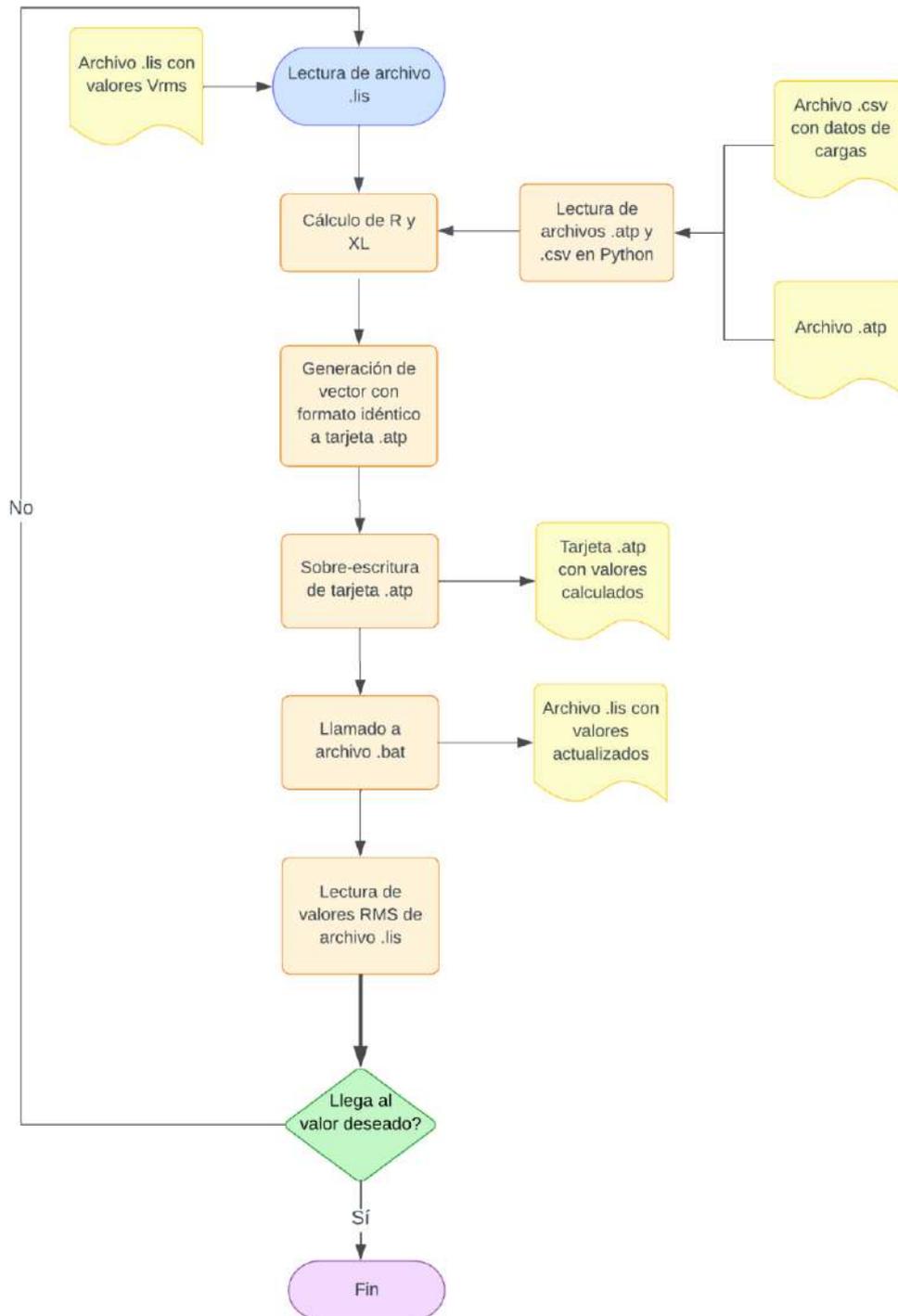


Figura 3-6. Diagrama de flujo del algoritmo de convergencia.

3.4 Algoritmo de fallas

Para estudiar el desempeño de los algoritmos ASA [42] y eASA en el alimentador de pruebas de 34 nodos, se simularon fallas eléctricas de distintos tipos (monofásicas y bifásicas con y sin impedancia a tierra) en todos los nodos de la red. En el presente trabajo se propuso

muestra en la Ecuación 3-7. En el caso de la estación 7, se la ubico en ese lugar para analizar cómo se propaga un evento de tensión a través de un transformador.

El diagrama de flujo del algoritmo es el siguiente:

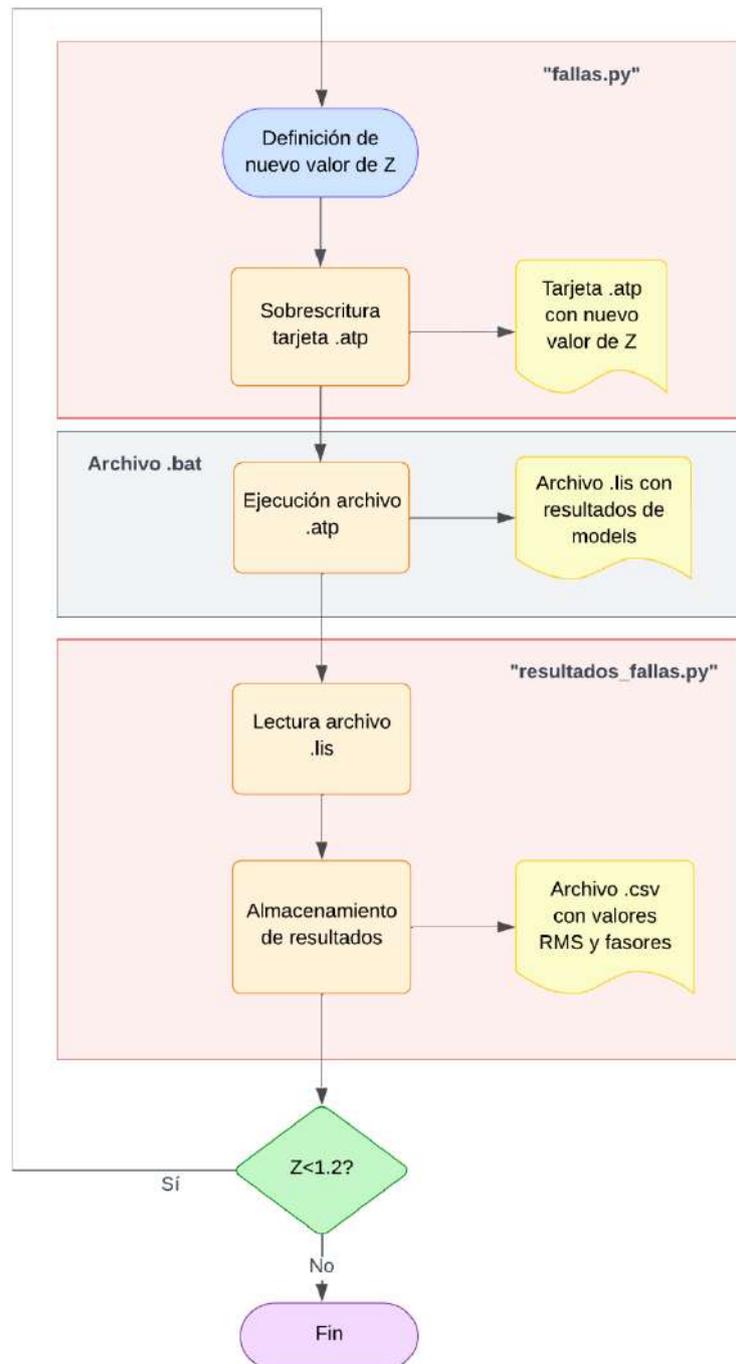


Figura 3-8. Diagrama de flujo del algoritmo de fallas. Z representa a la impedancia de falla.

Para lograr la iteración del algoritmo se utilizó un archivo de procesamiento por lotes de extensión “*.bat”, cuyo contenido se observa a continuación.

Archivo batch

```
for /l %%x in (1,1,31) do ( # esta línea da la orden de ejecutar secuencialmente los siguientes archivos de uno en uno, un total de 31 veces.
start "C:\SimulacionesATP\FALLAS\simulacion/" fallas.py
TIMEOUT /T 14
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando.vbs"
TIMEOUT /T 9
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando_cierre.vbs"
TIMEOUT /T 4
start "C:\SimulacionesATP\FALLAS\simulacion/" falla.atp
TIMEOUT /T 5
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando2.vbs"
TIMEOUT /T 65
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando2_cierre.vbs"
TIMEOUT /T 4
start "C:\SimulacionesATP\FALLAS\simulacion/" resultados_fallas.py
TIMEOUT /T 13
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando3.vbs"
TIMEOUT /T 30
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando3_cierre.vbs"
TIMEOUT /T 4
)
```

En la Figura 3-9 se muestran los archivos que el archivo *Batch* ejecuta en cada iteración.

Nombre	Fecha de ...	Tipo	Tamaño
fallas	24/2/2023 ...	Archivo de origen Python	2 KB
resultados_fallas	19/1/2023 ...	Archivo de origen Python	5 KB
comando	18/1/2023 ...	Archivo de secuencia de comandos de VBScript	1 KB
comando2	13/11/202...	Archivo de secuencia de comandos de VBScript	1 KB
comando2_cierre	17/11/202...	Archivo de secuencia de comandos de VBScript	1 KB
comando3	17/11/202...	Archivo de secuencia de comandos de VBScript	1 KB
comando3_cierre	17/11/202...	Archivo de secuencia de comandos de VBScript	1 KB
comando4_cierre	18/1/2023 ...	Archivo de secuencia de comandos de VBScript	1 KB
lista_nodos_fallas	19/1/2023 ...	Archivo de valores separados por comas de Microsoft Excel	1 KB
resultados_fallas	26/2/2023 ...	Archivo de valores separados por comas de Microsoft Excel	82 KB
falla	26/2/2023 ...	Archivo LIS	385 KB
inicio	26/1/2023 ...	Archivo por lotes de Windows	1 KB
falla	26/2/2023 ...	ATP File	78 KB
falla	26/2/2023 ...	ATP PL4 File	1 KB

Figura 3-9. Carpeta con la totalidad de los archivos utilizados para simular las fallas con una variación de impedancia automática.

Los tiempos de espera luego de cada instrucción “*TIMEOUT*” que figuran en el *Batch* fueron definidos a prueba y error considerando los tiempos de ejecución requeridos por cada sentencia.

Nótese que hay dos archivos de extensión “*.py”. Uno de estos archivos es el encargado de leer el archivo “*.atp” y variar la impedancia de falla (6.1.3.1). Luego se ejecuta el archivo ATP

para simular el circuito, y el segundo archivo “*.py” es el encargado de leer el archivo “*.lis” y almacenar los resultados obtenidos en un csv.(6.1.3.2).

La tasa de cambio del valor de impedancia de falla se definió de manera de obtener 31 iteraciones por nodo, comenzando en una falla franca y finalizando con un valor de 1,35 p.u. Esto se hizo con el objetivo de obtener la mínima cantidad de datos necesarios para realizar los análisis posteriores, pero logrando generar una muestra representativa sin tener que incurrir a un excesivo esfuerzo computacional por aumentar la cantidad de fallas por nodo. Este criterio es totalmente editable desde el código (6.1.3.1), es decir puede cambiarse la tasa de cambio de impedancia de falla según el usuario vea conveniente.

Una vez reemplazado el valor de impedancia de falla en la tarjeta “*.atp”, se ejecuta la simulación para obtener los nuevos valores de tensión y corriente RMS y sus fasores. Esto se realiza mediante las siguientes líneas del código en el archivo batch:

```
[...]  
start "C:/SimulacionesATP/FALLAS/simulacion/" falla.atp  
TIMEOUT /T 5  
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando2.vbs"  
TIMEOUT /T 65  
wscript "C:\SimulacionesATP\FALLAS\simulacion\comando2_cierre.vbs"  
TIMEOUT /T 4  
[...]
```

La línea destacada en negrita es la que corresponde a la apertura del archivo “*.atp”. Luego, es necesario ejecutar dos comandos para dar la orden de ejecución y de cierre del archivo mencionado. A continuación, se presentan como ejemplo, dos comandos en “*.vbs”: uno de ejecución y otro de cierre. La tecla “F2” inicia la simulación de ATP, y con “F4” se cierra el programa.

```
“comando2.vbs”  
Set WshShell = WScript.CreateObject("WScript.Shell")  
WshShell.AppActivate "falla.atp"  
WshShell.SendKeys "{F2}"  
-----  
“comando2_cierre.vbs”  
Set WshShell = WScript.CreateObject("WScript.Shell")  
WshShell.AppActivate "falla.atp"  
WshShell.SendKeys "%{F4}"
```

Una vez finalizada la simulación del circuito con el nuevo valor de impedancia de falla, se almacenan los valores RMS y fasores de tensión y corriente en un archivo csv luego de la lectura de los resultados contenidos en el archivo “*.lis” que genera automáticamente ATP. Este procedimiento se realiza a través de la ejecución del segundo código de programación “resultados_fallas.py”.

Para el correcto funcionamiento de este código, es necesario indicar mediante un archivo “*.csv” los datos de las estaciones de medición:

- Estación: número de la estación de medición, en este trabajo enumeradas del 1 al 10.
- Tipo: hace referencia a si la estación de medición está en un línea monofásica o trifásica.
- Nodo: Se debe indicar el nodo al que corresponde cada una de las estaciones de medición.
- Fase: de tratarse de una línea monofásica, se deberá indicar la fase.

La función que cumple este archivo “*.csv” es la de indicarle al algoritmo “resultados_fallas.py” como buscar los resultados de la simulación en el archivo “*.lis”, y como almacenarlos correctamente para los cálculos posteriores.

En la siguiente tabla (Tabla 3-2) se muestra un ejemplo del formato del archivo csv utilizado.

Tabla 3-2. Ejemplo de formato de los datos de estaciones de medición utilizado con el algoritmo de fallas.

Estación	Tipo (m o t)	Nodo	Fase (a, b o c)
1	t	800	-
2	t	850	-
3	m	818	a
4	m	820	a
5	t	824	-
6	t	852	-
7	t	888	-
8	t	834	-
9	t	844	-
10	t	860	-

Nota: para el caso de estaciones de medición trifásicas no debe ingresarse ningún valor en la última columna (Fase).

Se generó por nodo y tipo de falla un archivo “*.csv” con el formato mostrado en la tabla Tabla 3-3. Contiene la información de valores RMS de tensiones (V_a , V_b , V_c) y corrientes (I_a , I_b , I_c), y los ángulos de los fasores para estas tensiones (ϕ_{V_a} , ϕ_{V_b} , ϕ_{V_c}) y corrientes (ϕ_{I_a} , ϕ_{I_b} , ϕ_{I_c}) medidas luego de simular de la falla. La manera en que se calculan los valores RMS y los fasores se detalla en 3.4.1.

En la segunda columna, “Z[Ω]”, se guardan los valores de impedancia de falla para cada simulación. En este trabajo, la impedancia de falla es de naturaleza resistiva.

En la tercera columna, “t [s]”, se guardan los instantes de tiempo en los que se realizan las mediciones de tensión y corriente. Las mediciones de tensión y corriente se efectúan en el

ciclo 1 post-falla y el ciclo 6 post-falla. Con esto se buscó relevar información de un tiempo muy cercano a la falla, y otra medición varios ciclos posteriores a la falla para analizar como varía el comportamiento de la red post-falla al evolucionar en el tiempo.

Tabla 3-3. Esquema de formato de almacenamiento de resultados de simulación de fallas.

Estación	Z [Ω]	t [s]	V _a	V _b	V _c	Θ_{V_a}	Θ_{V_b}	Θ_{V_c}	I _a	I _b	I _c	Θ_{I_a}	Θ_{I_b}	Θ_{I_c}
1														
2														
...														
10														

Con la metodología desarrollada se obtuvieron por nodo y tipo de falla una cantidad de 620 registros. Un registro equivale a una fila de la Tabla 3-3:

$$Falla_{Nodo\ x-Tipo\ x} = 31\ iteraciones * (10\ Mediciones_{t-1\ ciclo} + 10\ Mediciones_{t-5\ ciclo}) = 620$$

Siendo:

Nodo x: El número de nodo donde se simuló la falla

Tipo x: Corresponde al tipo de falla simulado, monofásica, bifásica a tierra, y bifásica sin contacto a tierra.

Las 31 iteraciones representan la cantidad de veces que se varió la impedancia de falla para el mismo tipo de falla en el mismo nodo de la red.

Esto se repitió para la totalidad de los nodos. Hay que tener en cuenta que algunas de las líneas de la red eléctrica simulada eran del tipo monofásicas (7 nodos se ubican en líneas monofásicas). De esta manera, se simularon 34 fallas monofásicas, 27 bifásicas con puesta a tierra y 27 bifásicas sin puesta a tierra.

Esto da un total de registros de:

$$\begin{aligned} Total_{registros} &= 620 * (34\ Fallas_{monofásica} + 27\ Fallas_{bifásica} + 27\ Fallas_{Bifásica\ a\ tierra}) \\ &= 54560\ registros \end{aligned}$$

3.4.1 Cálculo de valores RMS y fasores con MODELS

Se utilizó la herramienta MODELS de ATP para medir la tensión y corriente en los 10 nodos seleccionados (estaciones de medición) durante la simulación. Para ello se hizo uso de dos códigos MODELS desarrollados por ATP [52], los cuales fueron modificados levemente para que la escritura de los resultados tenga un formato apropiado.

Para el funcionamiento de estos códigos es necesario definir los nodos sobre los que se efectuaran las mediciones, es decir el INPUT. El formato es:

```

/MODELS
MODELS
INPUT
  MM0001 {v(N800A)} #“v” indica que se mide tensión, “N800” es el nodo, “A” la fase
  MM0002 {v(N800B)}
  MM0003 {v(N800C)}
  ....
  MM0090 {i(NM01A)} #“i” indica que se mide corriente, “NM01” es el nodo, “A” la fase
  MM0091 {i(NM01B)}
  MM0092 {i(NM01C)}

```

Para utilizar los algoritmos, deben llamarse en MODELS utilizando la expresión “USE”. Se muestran dos ejemplos, el primero para hacer uso del algoritmo de cálculo de valor RMS y el segundo para calcular los fasores.

```

USE RMSMOD3 AS DEFAULT #cálculo de tensiones de fase RMS en el nodo 850
INPUT
  x[1]:= MM0020
  x[2]:= MM0021
  x[3]:= MM0022
DATA
  xrms_ini:= -1.
  scale:= 24900.
  SampleFreq:= 480.
  tsampleo:= 0.1333
  n:= 3.
  nodo:= 850.
  muestra:= 1.
  tipo:= 1.
ENDUSE

```

```

USE FASOR AS DEFAULT #cálculo de fasores de las 3 tensiones de fase en el nodo 850
INPUT
  v[1]:= MM0020
  v[2]:= MM0021
  v[3]:= MM0022
DATA
  N:= 3.
  TIME:= 0.1333
  SampleFreq:= 480.
  nodo:= 850.
  muestra:= 1.
ENDUSE

```

3.4.1.1 Cálculo de valores RMS

El código encargado de realizar el cálculo RMS para las tensiones y corrientes se encuentra anexo en 6.1.5.4. Este algoritmo resuelve la siguiente ecuación:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T v^2(t) dt}$$

$$I_{rms} = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt}$$

ATP discretiza esta integral como se indicó en el *Capítulo 2: 2.2.1 Cálculo numérico*. Para efectuar la integral en un período de tiempo, debe almacenarse los valores de tensión y corriente de un período hacia atrás en el tiempo. Esto se hace utilizando la siguiente expresión:

```
DELAY CELLS(ix2[1..n]): recip(freq*timestep) +2
```

Siendo:

recip: indica que la expresión es 1/(..).

freq: es la frecuencia del sistema, en esta simulación 60 [Hz].

timestep: paso del tiempo definido en la simulación. En esta simulación se definió un paso del tiempo de $5 \cdot 10^{-7}$.

$$Celdas\ guardadas = \frac{1}{(freq * timestep)} + 2 = \frac{período}{timestep} + 2$$

Es decir, se almacenan los datos de las celdas (cada celda hace referencia al dato medido instantáneamente en ese paso de tiempo) correspondientes a un período completo de la señal medida, más 2 celdas extras que no son tenidas en cuenta en el cálculo, pero que se recomienda agregar.

Para calcular la integral se utiliza la expresión:

```
ix2[k] := integral(x2[k])
```

El cálculo de los valores RMS se realiza 1 ciclo post-falla, y 5 ciclos luego de la falla. Esto se indica al llamar al código MODEL para realizar el cálculo RMS mediante la sentencia "USE":

```
USE RMSMOD3 AS DEFAULT #se realiza el llamado del código MODEL que realiza el
cálculo RMS
INPUT
x[1]:= MM0020
x[2]:= MM0021
x[3]:= MM0022
DATA
xrms_ini:= -1.
scale:= 24900. #se utiliza para obtener los valores en p.u
```

```

SampleFreq:= 480.
tsamplero:= 0.1333 #indica el tiempo en el que se llamará al código MODEL RMS
n:= 3.
nodo:= 850.
muestra:= 1.
tipo:= 1.
ENDUSE

```

Donde:

tsamplero: indica en qué tiempo se llama al código MODEL. Recordar que el código calcula el valor RMS con los datos medidos un período atrás en el tiempo. Con lo cual, la medición es llevada a cabo descartando el primero ciclo post-falla e integrando el ciclo siguiente:

$$t_{samplero} = t_{falla} + 2T$$

$$V_{rms} = \sqrt{\frac{1}{T} \int_{t_{falla}+T}^{t_{falla}+T+2T} v^2(t) dt}$$

Siendo T el período de la señal medida, en este caso la tensión.

El resto de los datos mostrados como “n” (números de fases, 1 o 3), “nodo” (número de nodo medido), “muestra” (1, 2 ... n según cuantas muestras de valores RMS se lleven a cabo), “tipo” (1 tensión, 2 corriente), son datos utilizados para la escritura de los resultados,

3.4.1.2 Cálculo de fasores

Para el cálculo de los fasores se utilizó el código anexo en 6.1.5.5.

Utilizando los mismos INPUT definidos para el cálculo de los valores RMS, este algoritmo calcula los fasores de tensión y corriente utilizando la *Transformada Rápida de Fourier (FFT) Radix2-8 algorithm*, método desarrollado por *Cooley and Tukey* [53]. Esta es una técnica matemática para transformar una señal digital en el dominio del tiempo en una representación en el dominio de la frecuencia de la amplitud relativa de diferentes regiones de frecuencia en la señal, siendo un algoritmo muy eficiente.

La idea básica de la *FFT* es descomponer la *DFT* (transformada discreta de Fourier) de una secuencia de dominio de tiempo de longitud N en *DFT* sucesivamente más pequeñas cuyos cálculos requieren menos operaciones aritméticas. El desarrollo matemático excede el propósito de este trabajo, pero se puede consultar en *Cooley and Tukey* [53]

El procedimiento de calcular la serie discreta de una DFT de N puntos en dos DFT de N/2 puntos puede adoptarse para calcular la serie de DFT de N/2 puntos a partir de elementos de DFT de N/4 puntos. Para este propósito, cada secuencia de N/2 puntos debe dividirse en dos

subsecuencias y calcular sus DFT consecutivamente. El diezmo de la secuencia de datos se puede repetir una y otra vez hasta que la secuencia resultante se reduzca a una DFT básica. El algoritmo aplicado en este trabajo realiza el cálculo de una DFT de N=8 puntos (Figura 3-10).

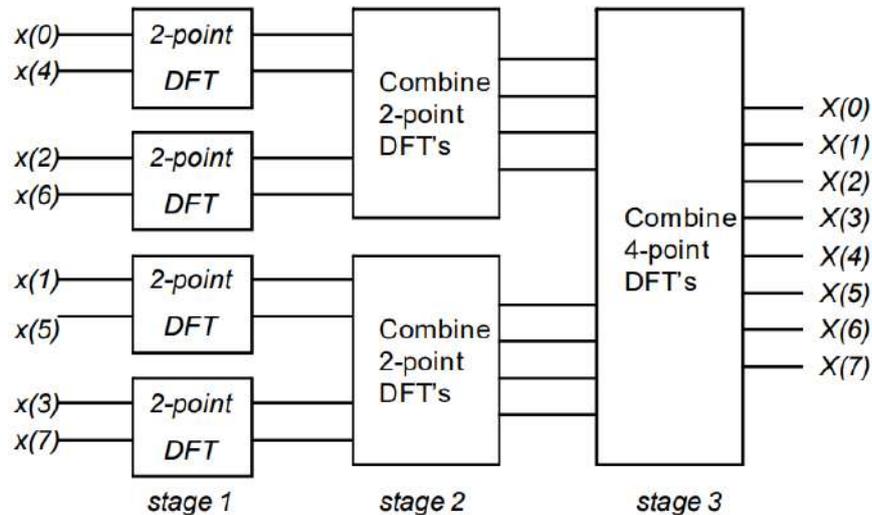


Figura 3-10. Descomposición de una DFT de 8 puntos.

El cálculo se realiza en tres etapas, comenzando con los cálculos de cuatro DFT de 2 puntos, luego dos DFT de 4 puntos y, finalmente, un DFT de 8 puntos. La combinación de las DFT más pequeñas para formar la DFT más grande para N = 8 se ilustra en la Figura 3.

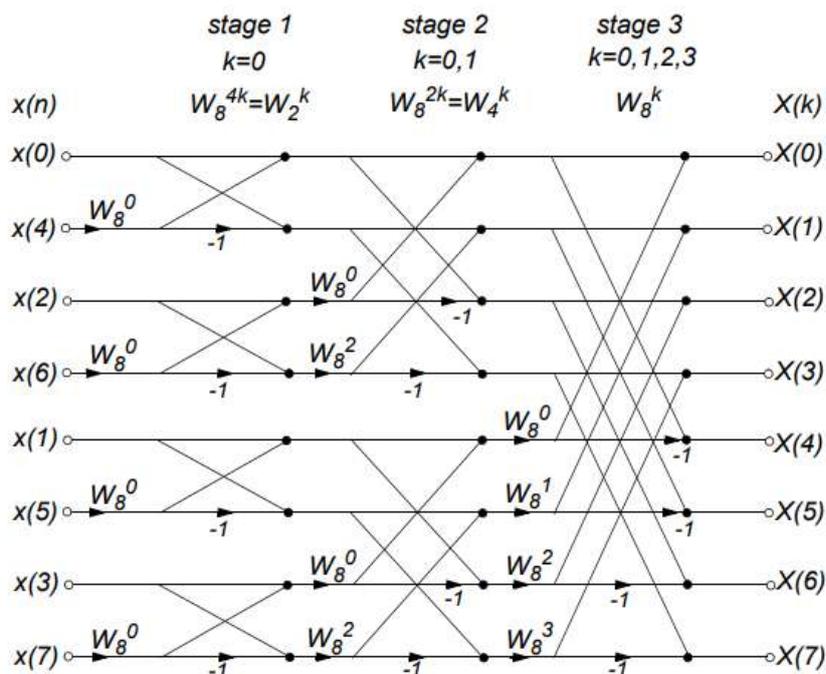


Figura 3-11. Flujo de datos del algoritmo DIT FFT de 8 puntos radix-2.

En la Figura 3-11, cada punto de color negro representa una suma compleja y cada flecha representa una multiplicación compleja.

3.4.2 Tratamiento de datos

Con el objetivo de simplificar el tratamiento de los datos obtenidos debido a su volumen, se optó por utilizar una librería compatible con Python *Matplotlib* [44] para generar de una manera automática los gráficos que se mostrarán en los siguientes capítulos. Los códigos en detalle se encuentran anexados en el apartado 6.1.4.

A partir de los archivos “*.csv” generados durante la simulación de las fallas, se elaboraron dos gráficos: un diagrama fasorial comparativo de los fasores de tensión pre y post falla; y un gráfico de tensión p.u vs Z_{falla} y uno del salto de fase versus Z_{falla} , generándose dos gráficos por Estación de medición (10 estaciones) para cada tipo de falla (monofásica, bifásica con y sin contacto a tierra).

A modo de ejemplo, en la Figura 3-12, Figura 3-13 y Figura 3-14 se muestran los gráficos fasoriales y los gráficos de tensión p.u vs Z_{falla} y el de salto de fase vs Z_{falla} a partir de mediciones en la Estación de medición 1, para fallas en el nodo 830. En los gráficos fasoriales además de las flechas que representan los fasores pre-falla (color transparente) y las flechas que simbolizan los fasores post-falla para una falla franca (color intenso), se muestran puntos consecutivos que representan los valores puntuales de los fasores para los distintos valores de Z_{falla} . Con esto se buscó representar visualmente la evolución de los fasores al variar las impedancias de falla, visualizando las diferentes “firmas” del evento de tensión.

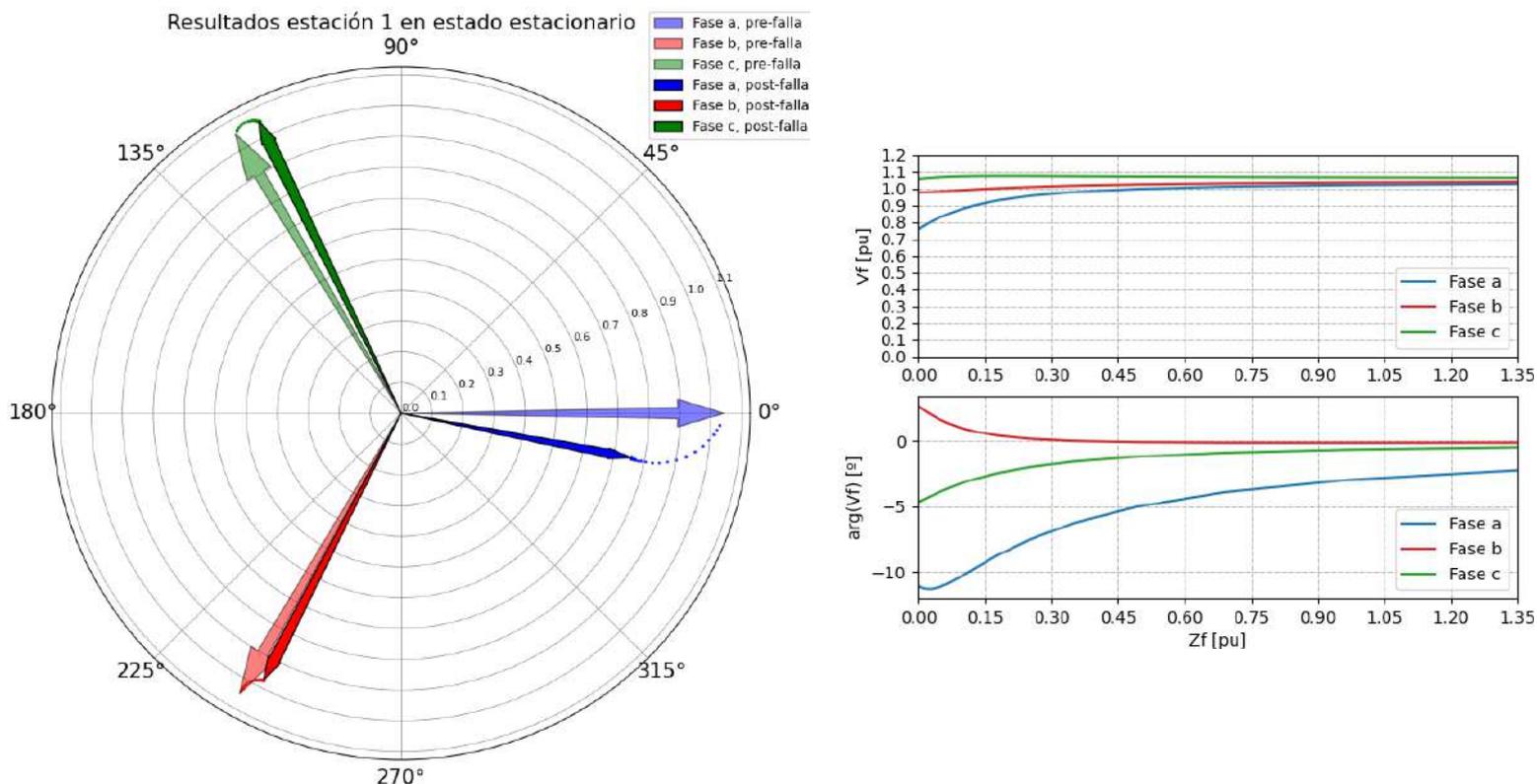


Figura 3-12. Falla monofásica en la fase A en el nodo 830, medida en la Estación de medición 1. A la izquierda gráfico de fasores de tensión. A la derecha-arriba gráficos de tensión p.u vs Z_{falla} . A la derecha-abajo gráfico de salto de fase vs Z_{falla} .

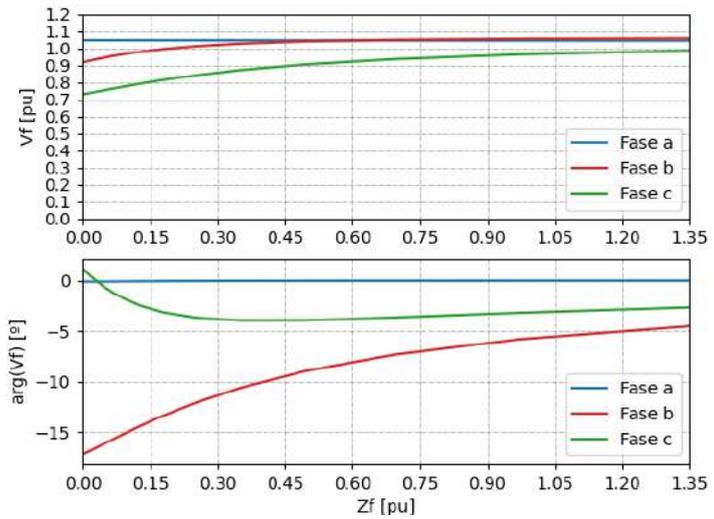
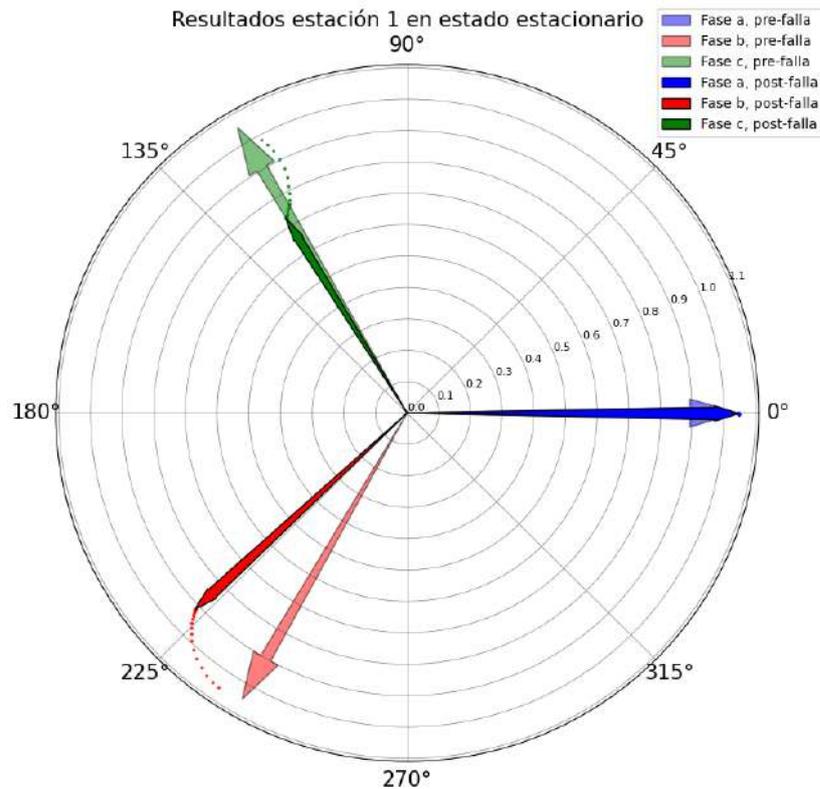


Figura 3-13. Falla bifásica sin contacto a tierra entre fases B y C en el nodo 830, medida en la Estación de medición 1. A la izquierda gráfico de fasores de tensión. A la derecha-arriba gráficos de tensión p.u vs Z falla. A la derecha-abajo gráfico de salto de fase vs Z falla.

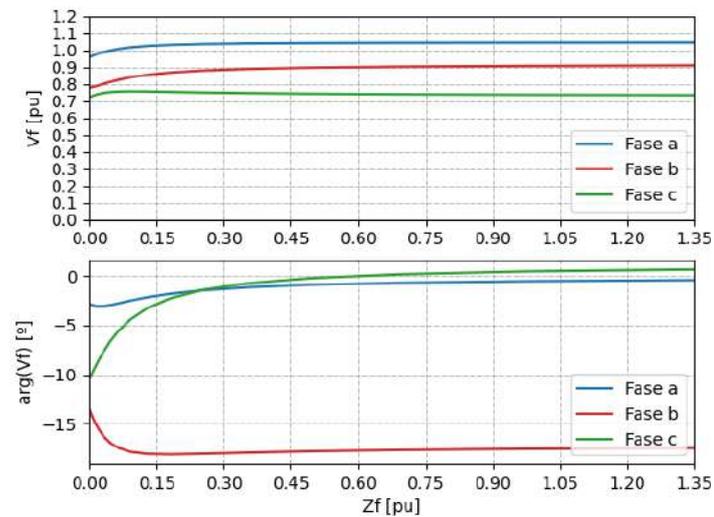
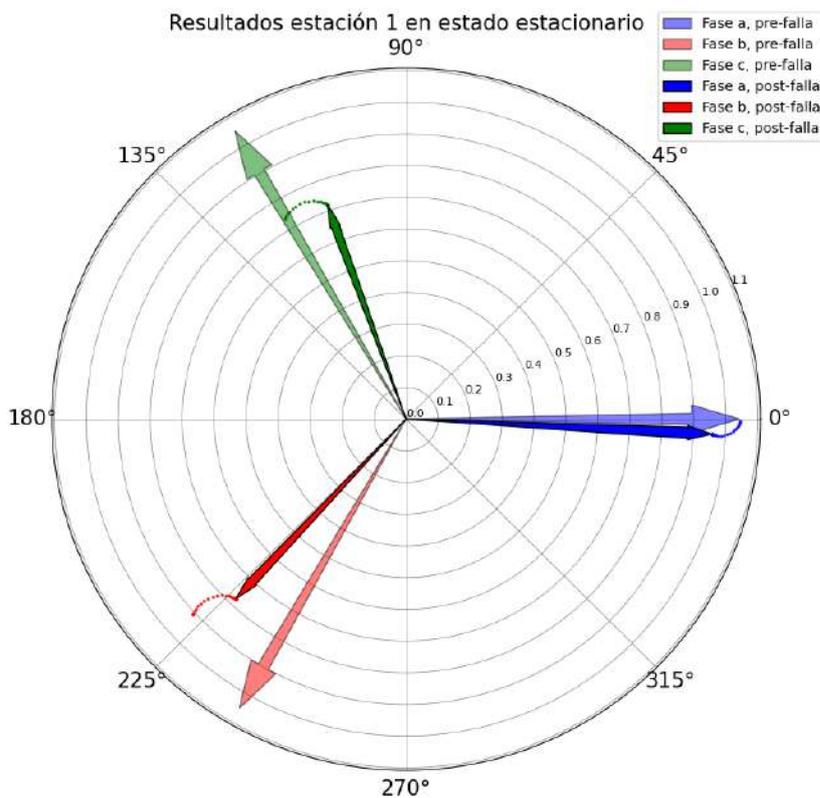


Figura 3-14. Falla bifásica con contacto a tierra entre fases B y C en el nodo 830, medida en la Estación de medición 1. A la izquierda gráfico de fasores de tensión. A la derecha-arriba gráficos de tensión p.u vs Z falla. A la derecha-abajo gráfico de salto de fase vs Z falla.

Si bien es un volumen considerable de datos a procesar, se decidió trabajar con la totalidad de los nodos con el objetivo de generar una base de datos. En el Capítulo 5 se utilizarán estos gráficos en conjunto con los algoritmos de detección de falla para analizar resultados.

3.5 Algoritmo de clasificación de eventos

En este trabajo se aplicarán los algoritmos *ASA* (*Algoritmo de secuencias absolutas*) [42] y *eASA* (*Algoritmo de secuencias absolutas modificado*) [63] con el objetivo de evaluar sus desempeños para clasificar los eventos de tensión generados.

Ambos algoritmos utilizan como dato de entrada tres componentes simétricas fundamentales, positiva, negativa y homopolar, de las tensiones trifásicas post-falla, y un estimador encargado de evaluar el tipo de evento más probable. A la hora de clasificar el evento, estos algoritmos utilizan un criterio de clasificación basado en modelos teóricos de fallas típicas denominado criterio ABC.

3.5.1 Clasificación ABC

La clasificación ABC presentada en [45], fue desarrollado como parte de una predicción estocástica de huecos de tensión y puede distinguir entre 9 tipos de eventos (*dips* y *swells*). Una característica importante de esta clasificación es que puede discriminar entre diferentes formas de onda de voltaje, y por lo tanto resulta más descriptivo debido a que tiene más variedad de eventos de tensión brindando más detalles a la hora de establecer una relación entre los eventos clasificados y los tipos de fallas que los causaron. Este criterio originalmente definía siete eventos de voltaje diferentes (tipos A a G), y luego fue ampliado con dos eventos adicionales (tipos H e I) por *Ignatova* [46].

Dentro de esta clasificación se distinguen siete tipos de *dips* trifásicos. El *dip* balanceado tipo A, los *dips* desbalanceados con la menor tensión residual en una fase conocidos como tipo B, D y F, y los *dips* desbalanceados con la menor tensión residual en dos fases, conocidos como tipo C, E y G.

Los tres tipos de eventos restantes de este criterio ABC, a diferencia de los planteados previamente que sólo representan *dips* de tensión, caracterizan tres *swells* de tensión. El evento tipo H presenta *swells* de tensión en dos fases y un *dip* en la fase restante, mientras que los eventos tipo I* e I** presentan un *swell* en una de las fases y *dips* en las otras dos. A continuación se desarrollará brevemente las clasificaciones que distingue el criterio ABC.

3.5.1.1 Evento tipo A

Se trata de un *dip* de tensión balanceado, es decir, se produce la misma disminución del valor eficaz en las tres tensiones de fase. Puede tener diversos orígenes, como por ejemplo un cortocircuito trifásico, el arranque de un motor trifásico asíncrono de elevada potencia en relación a la potencia de cortocircuito de la red, o cambios bruscos de carga trifásica balanceada debida a la conmutación de elementos de maniobra o la actuación de protecciones.

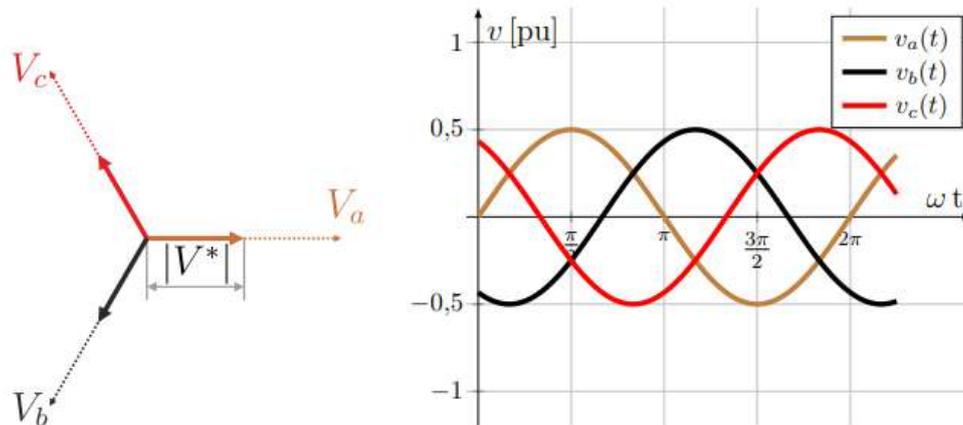


Figura 3-15. Diagrama fasorial y forma de onda instantánea de un evento tipo A con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.2 Evento tipo B

Es un *dip* de tensión desbalanceado cuya principal caída de tensión aparece sólo en una de las tres tensiones de fase. Este evento generalmente se origina debido a un cortocircuito entre una fase y tierra o una fase y neutro en un tramo de red tal que pueda tener lugar una componente de tensión homopolar distinta de cero. La aparición de esta componente homopolar es la responsable de que las dos tensiones de fase restantes no se vean afectadas en su magnitud ni en su fase, como se aprecia en la Figura 3-16.

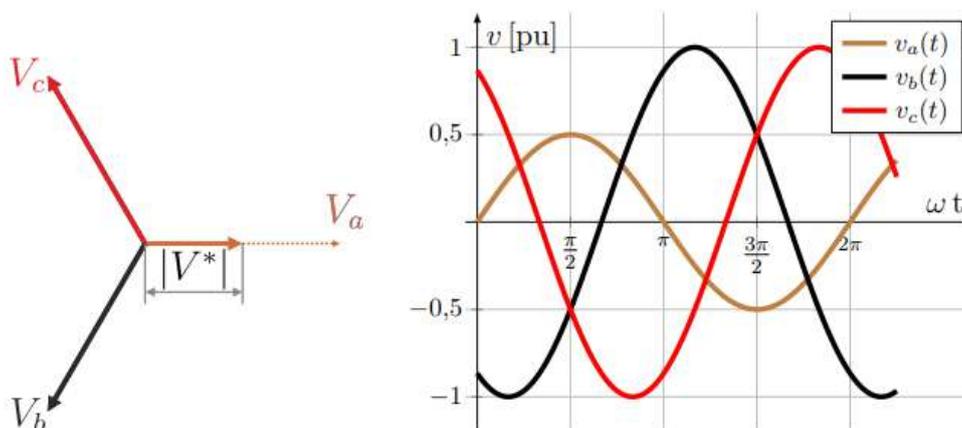


Figura 3-16. Diagrama fasorial y forma de onda instantánea de un evento tipo B con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.3 Evento tipo C

Es un *dip* de tensión desbalanceado en el cual disminuye el valor eficaz de dos de las tres tensiones de fase. Este evento se origina en general cuando se da un cortocircuito entre dos fases sin contacto a tierra o neutro. Además de verse afectadas las magnitudes de las dos tensiones, se produce un corrimiento de sus fases. Como se observa en la Figura 3-17, una de ellas sufre un adelanto de fase y la otra un atraso, para que la suma de las tres tensiones sea nula. Adicionalmente, la componente homopolar de tensión no puede tener lugar, ya que en el cortocircuito no hay contacto a tierra, de manera que no interviene la red de secuencia cero en el cálculo de la tensión de falla.

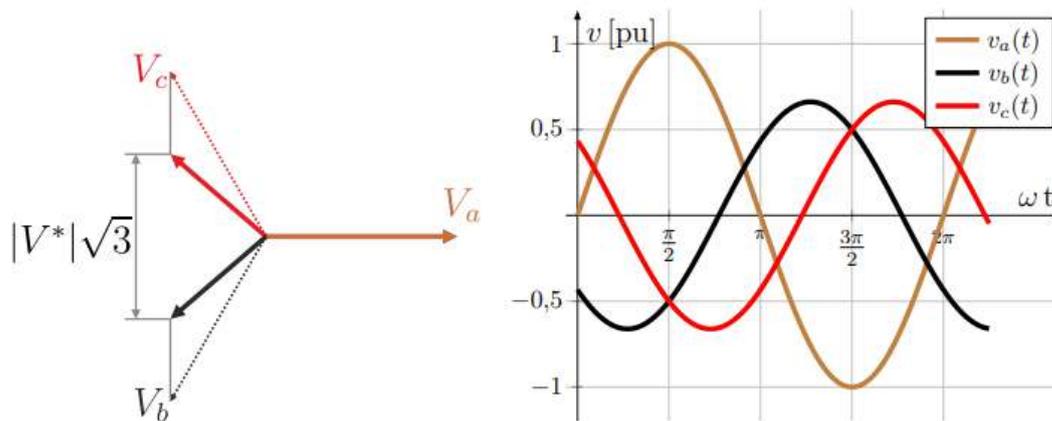


Figura 3-17. Diagrama fasorial y forma de onda instantánea de un evento tipo C con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.4 Evento tipo D

Es un *dip* de tensión desbalanceado con la principal caída en una fase, y la disminución de la parte real de las otras dos, originando sus desfasajes. No tiene un origen primario, sino que es consecuencia de la propagación de eventos originados a otros niveles de tensión a través de transformadores de potencia. Existen dos posibilidades, la primera es que provenga de un evento tipo B propagado a través de transformadores que eliminan la componente homopolar como los de conexión Y y o Dd. La segunda posibilidad es que el evento provenga de un evento tipo C propagado a través de transformadores que eliminan la componente homopolar y transforman tensiones de línea a fase o viceversa, como los de conexión YNd, Yd, Dyn y Dy.

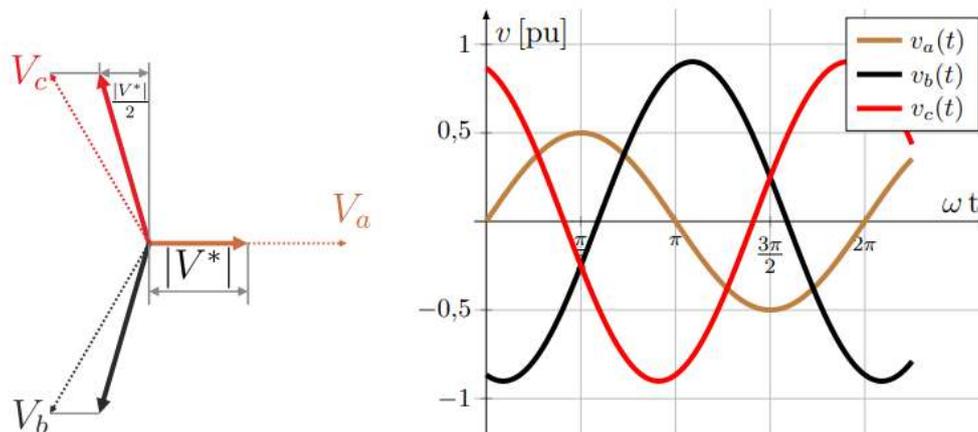


Figura 3-18. Diagrama fasorial y forma de onda instantánea de un evento tipo D con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.5 Evento tipo E

Se trata de un *dip* de tensión desbalanceado, que al igual que el evento tipo C, presenta una disminución del valor eficaz de dos de las tres tensiones de fase. A diferencia del evento tipo C, se origina cuando se da un cortocircuito entre dos fases con contacto a tierra. En este caso puede existir componente homopolar de la tensión fundamental, por lo tanto, como se aprecia en la Figura 3-19, no se produce un corrimiento de fase en las fases afectadas.

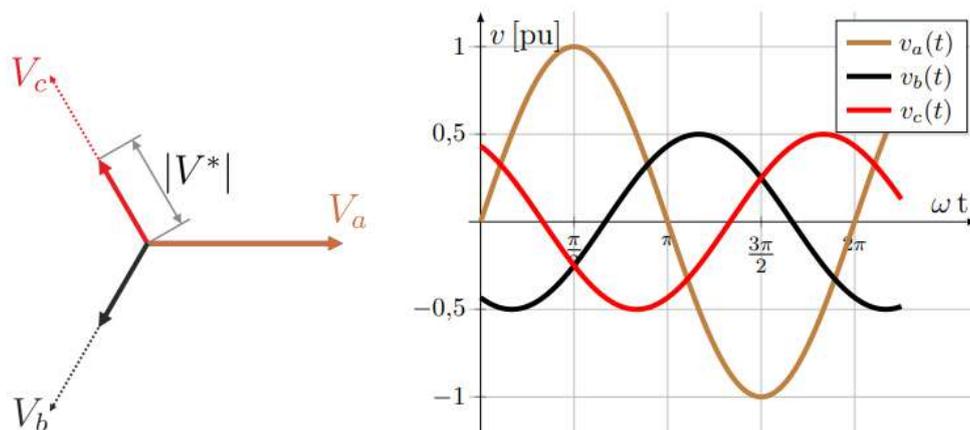


Figura 3-19. Diagrama fasorial y forma de onda instantánea de un evento tipo E con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.6 Evento tipo F

Es un *dip* de tensión desbalanceado con la principal caída en una tensión de fase, con una disminución de la componente real e imaginaria de las otras dos tensiones originando un desfase en las mismas. Al igual que el evento tipo D, no tiene un origen primario, sino que es resultado de la propagación de eventos originados a otros niveles de tensión a través de transformadores de potencia. En general, es el resultado de un evento tipo E propagado a través de transformadores que eliminan la componente homopolar y que transforman

tensiones de línea en tensiones de fase o viceversa, como por ejemplo transformadores con conexión YNd, Yd, Dyn y Dy.

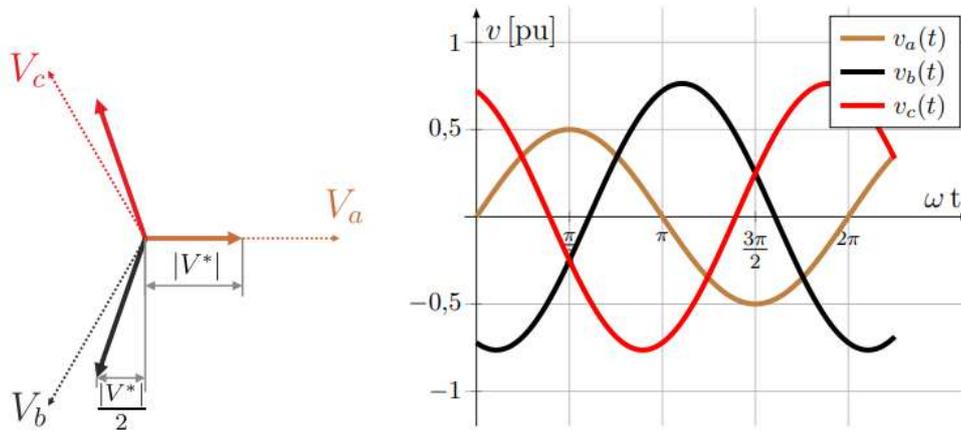


Figura 3-20. Diagrama fasorial y forma de onda instantánea de un evento tipo F con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.7 Evento tipo G

Es un *dip* de tensión desbalanceado con la principal caída en dos tensiones de fase y una disminución de la magnitud, aunque de menor proporción en la tensión restante. Al igual que los eventos D y F no tiene un origen primario, sino que es el resultado de la propagación de eventos originados a otros niveles de tensión a través de transformadores de potencia. En general, es el resultado de un evento tipo E propagado a través de transformadores que eliminan la componente homopolar, como aquellos con conexión Yy o Dd.

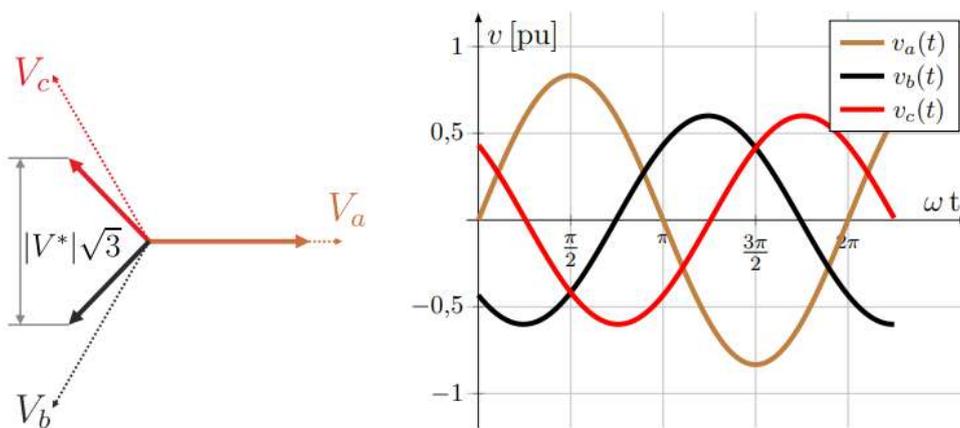


Figura 3-21. Diagrama fasorial y forma de onda instantánea de un evento tipo G con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.8 Evento tipo H

Es un evento de tensión caracterizado por presentar un *dip* de tensión en una de las tres tensiones de fase y un *swell* en las dos tensiones de fase restantes como se observa en la Figura 3-22. Se puede originar por una falla monofásica a tierra en un sistema trifásico sin neutro, o con neutro impedante (presenta alta impedancia a tierra), o por una falla monofásica a neutro de elevada impedancia.

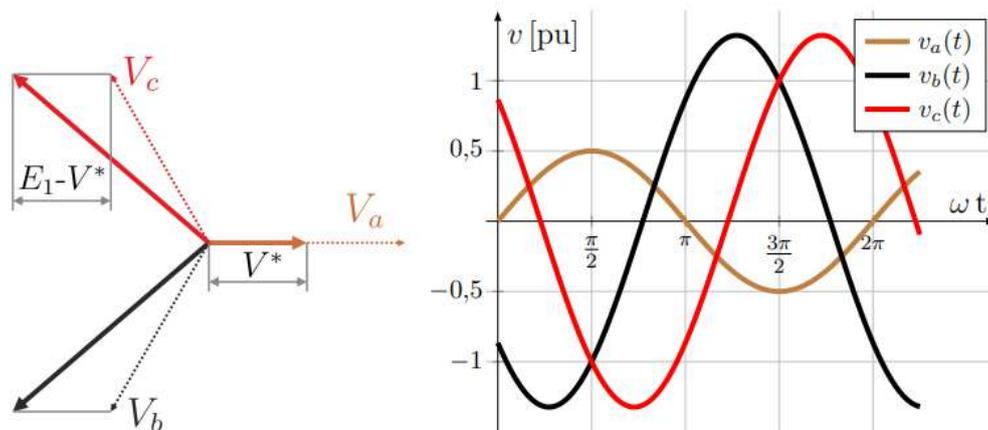


Figura 3-22. Diagrama fasorial y forma de onda instantánea de un evento tipo H con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.9 Evento tipo I*

Es un evento caracterizado por presentar un *swell* de tensión en una de las tres tensiones de fase y un *dip* de tensión en las otras dos tensiones de fase, tal como se ve en la Fig. B.9. El mismo se puede originar por un cortocircuito bifásico a tierra en un sistema trifásico sin neutro o con neutro impedante, o por una falla bifásica a neutro de elevada impedancia. El mismo está definido para tensiones residuales comprendidas entre $0.75 \text{ p.u.} < V^* < 0.9 \text{ p.u.}$

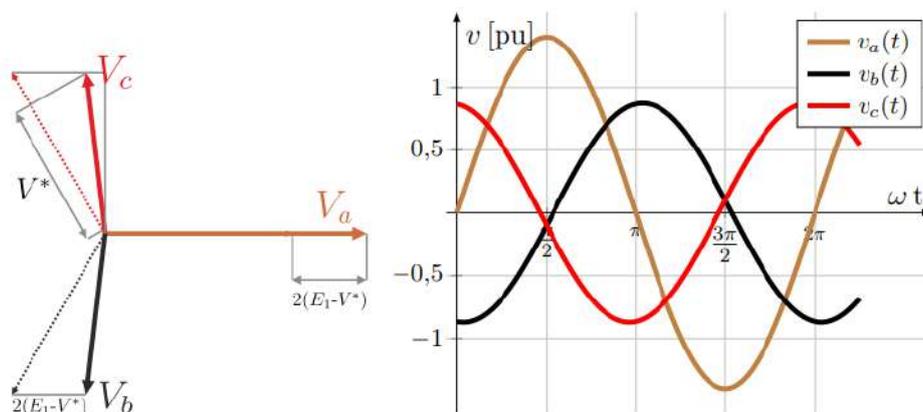


Figura 3-23. Diagrama fasorial y forma de onda instantánea de un evento tipo I* con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.1.10 Evento tipo I**

Es un evento caracterizado por presentar un *swell* de tensión en una de las tres tensiones de fase y un *dip* de tensión en las otras dos tensiones de fase, tal como se ve en la Fig. B.10. El mismo se puede originar por un cortocircuito bifásico a tierra en un sistema trifásico sin neutro o con neutro impedante o por una falla bifásica a neutro de elevada impedancia. El mismo está definido para tensiones residuales comprendidas entre $0.1 \text{ p.u.} < V^* \leq 0.75 \text{ p.u.}$

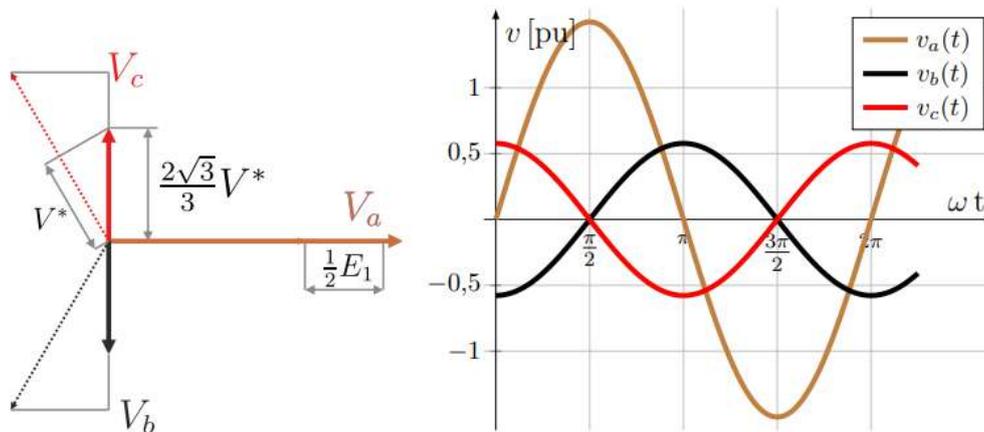


Figura 3-24. Diagrama fasorial y forma de onda instantánea de un evento tipo I** con tensión residual $V^*=0,5$ [p.u.]. [42]

3.5.2 Algoritmo de secuencias absolutas

Para realizar la clasificación de eventos de tensión bajo el criterio ABC se empleará un algoritmo propuesto por *Strack J.* [42] denominado *algoritmo de secuencias absolutas* o *ASA*, que utiliza la información provista por el valor absoluto de las tres componentes simétricas fundamentales de las tensiones trifásicas calculadas a partir de mediciones reales y un estimador que evalúa cual es el tipo de evento más probable de acuerdo a las tensiones medidas. A diferencia de otros algoritmos que utilizan umbrales fijos, este presenta umbrales adaptivos por lo cual se adapta mejor a los cambios de amplitud y fase durante una falla en la red.

Para la aplicación del algoritmo *ASA*, en este trabajo se calculan las componentes simétricas de tensión de las fases A, B y C a partir de las tensiones medidas y almacenadas en cada una de las Estaciones de medición trifásicas. Este procedimiento se repite para la totalidad de las fallas simuladas, es decir para fallas en todos los nodos de la red (monofásica, y bifásica con y sin contacto a tierra si corresponde) y aplicando las variaciones en los valores de impedancia de falla. Este procedimiento se realiza aplicando el código Python que se encuentra anexado en el apartado 6.1.5.1.

El cálculo de las componentes simétricas desarrollado por *Fortescue* [47] consiste en descomponer un sistema asimétrico en tres “sistemas simétricos”. Para explicar brevemente

como se realiza esta descomposición de la terna original, en primer lugar, se define el parámetro “a” o vector operador de giro, cuyo valor es:

$$a = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$$

Por lo tanto, el vector “a” tendrá magnitud unidad y orientación a 120° en dirección positiva desde el eje de referencia.

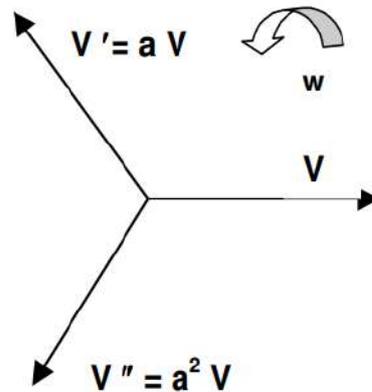


Figura 3-25. Vector V al aplicarse el vector operador “a” y “a²”. [48]

Terna de secuencia positiva (se indicará con subíndice 1).

Si el sistema original posee secuencia a-b-c, esta terna poseerá la misma secuencia. Como esta es una terna perfecta solo será necesario determinar el módulo y fase de uno de los fasores, en función de los fasores originales. Aplicando el operador “a” la terna de secuencia directa queda identificada como:

$$V_{a1} = |V_{a1}| \angle 0^\circ$$

$$V_{b1} = a^2 V_{a1} = |V_{a1}| \angle -120^\circ$$

$$V_{c1} = a V_{a1} = |V_{a1}| \angle 120^\circ$$

Terna de secuencia negativa (se indicará con subíndice 2).

Si el sistema original posee secuencia a-b-c, esta terna poseerá una secuencia de fases a-c-b y quedará expresada por:

$$V_{a2} = |V_{a2}| \angle 0^\circ$$

$$V_{b2} = a V_{a2} = |V_{a2}| \angle 120^\circ$$

$$V_{c2} = a^2 V_{a2} = |V_{a2}| \angle -120^\circ$$

Terna de secuencia nula (se indicará con subíndice 0).

En este caso los tres fasores forman un sistema monofásico de manera tal que:

$$V_{a0} = V_{b0} = V_{c0} = V_0$$

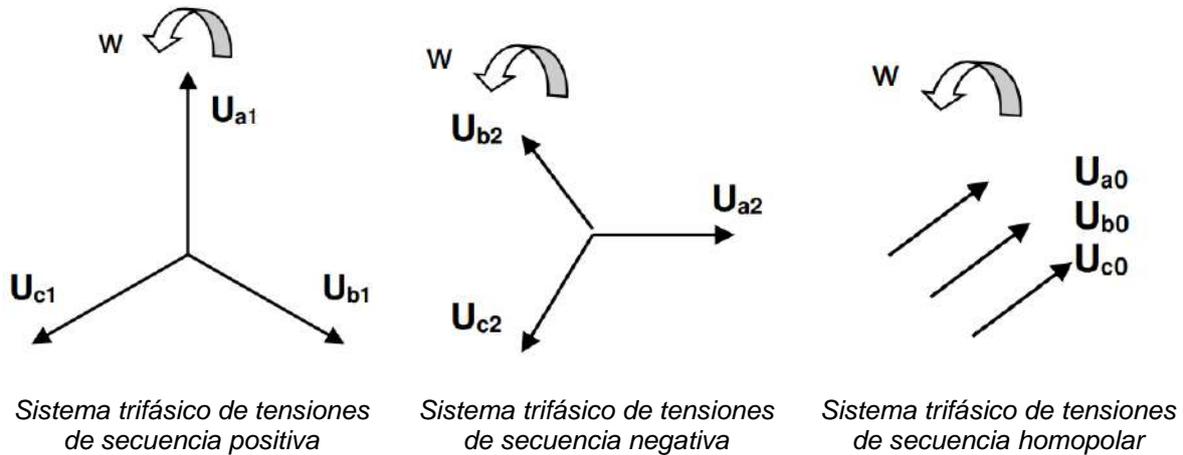


Figura 3-26. Sistemas simétricos de fasores de tensiones trifásicas. [48]

Para que el teorema de Fortescue se cumpla se debe cumplir el siguiente sistema de ecuaciones:

$$V_a = V_{a0} + V_{a1} + V_{a2}$$

$$V_b = V_{b0} + V_{b1} + V_{b2} = V_{a0} + a^2 V_{a1} + a V_{a2}$$

$$V_c = V_{c0} + V_{c1} + V_{c2} = V_{a0} + a V_{a1} + a^2 V_{a2}$$

Siendo V_a , V_b y V_c los fasores de tensiones de fases. Y en formato matricial:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \cdot \begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix}$$

A la matriz de 3x3 se la llama matriz de transferencia directa [F]. Si de la ecuación matricial previa se despeja el vector de las componentes simétricas se obtiene:

$$\begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \cdot \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

El desarrollo detallado y ejemplos de aplicación del método de las componentes simétricas puede consultarse en *Ferro* [48].

Una vez obtenidos los valores de las componentes simétricas correspondientes a las tensiones de fase trifásica post-falla, se procede a calcular las predicciones del modelo, que

corresponden a los valores absolutos de las componentes simétricas obtenidas a partir del modelo analítico de clasificación ABC de cada tipo de evento descrito en 3.5.1.

Tabla 3-4. Componentes simétricas función de h y E_1 para cada tipo de evento utilizada por el Algoritmo ASA. [42]

Tipo	$ V_{01} $ [pu]	$ V_{+1} $ [pu]	$ V_{-1} $ [pu]
A	0	$E_1 h$	0
B	$E_1(1-h)/3$	$E_1(2+h)/3$	$E_1(1-h)/3$
C	0	$E_1(1+h)/2$	$E_1(1-h)/2$
D	0	$E_1(1+h)/2$	$E_1(1-h)/2$
E	$E_1(1-h)/3$	$E_1(1+2h)/3$	$E_1(1-h)/3$
F	0	$E_1(1+2h)/3$	$E_1(1-h)/3$
G	0	$E_1(1+2h)/3$	$E_1(1-h)/3$
H	$E_1(1-h)$	E_1	0
I*	$2E_1(1-h)$	E_1	0
I**	$E_1/2$	$E_1(1+2h/3)$	$E_1(1-2h/3)$

*Nota: El tipo de evento I se define por tramos: si $h \leq 0.75$ $I = I^{**}$; si $h > 0.75$, $I = I^*$. Todas las tensiones se expresan en por unidad [pu] adoptando como base la tensión RMS nominal o declarada de la red bajo análisis.*

El cálculo simplificado de la tensión de prefalla “ E_1 ” es:

$$E_1[n] = \frac{V_{sr_a} + V_{sr_b} + V_{sr_c}}{3}$$

Donde V_{sr-} se estimó como el valor de tensión de fase RMS antes de la falla. En consecuencia, E_1 corresponde al promedio aritmético de las tres tensiones de fase RMS antes de la falla. Esta ecuación es una aproximación del valor de prefalla utilizado en *Strack* [42], y puede simplificarse de esta manera por tratarse de una simulación, en donde el valor de tensión de pre-falla se encontraba en régimen estacionario sin variaciones. En la realidad, los valores de tensiones suelen tener variaciones y por tal motivo estimar la tensión de prefalla de todas las tensiones de referencia deslizantes no es tan simple, debiendo seguir lineamientos normalizados, por ejemplo, el que propone la norma IEC 61000-4-30:2015. Para este trabajo, la simplificación de la ecuación se considera válida.

A continuación, se detallan los valores de tensión para cada estación de medición junto con el valor obtenido del parámetro E_1 (6.1.5.2).

Tabla 3-5. Valores de tensión RMS y E_1 de las estaciones de medición trifásicas. Notar que las estaciones de medición 3 y 4 son obviadas en esta tabla (ubicadas en líneas monofásicas). Esto es porque el algoritmo ASA se aplica con tensiones trifásicas.

Estación	V_{RMSa} [p.u]	V_{RMSb} [p.u]	V_{RMSc} [p.u]	E_1
1	1.046	1.050	1.054	1.050
2	1.014	1.023	1.019	1.019
5	1.005	1.013	1.010	1.009
6	0.957	0.966	0.960	0.961
7	1.001	0.999	0.998	1.000
8	1.030	1.027	1.027	1.028
9	1.029	1.027	1.027	1.028

Para efectuar los cálculos correspondientes al algoritmo, es necesario obtener los valores de tensión máximo y mínimo. Los mismos se obtienen mediante la siguiente expresión:

$$V_{min} = \{|V_a|, |V_b|, |V_c|, \frac{|V_a|}{\sqrt{3}}, \frac{|V_b|}{\sqrt{3}}, \frac{|V_c|}{\sqrt{3}}\}$$

$$V_{max} = \{|V_a|, |V_b|, |V_c|, \frac{|V_a|}{\sqrt{3}}, \frac{|V_b|}{\sqrt{3}}, \frac{|V_c|}{\sqrt{3}}\}$$

El parámetro “h” representa la amplitud del evento y se estima como $h = V/E_1$ donde V es la tensión en la fase fallada durante el evento. La misma se puede estimar como la menor de las seis tensiones RMS fundamentales, es decir, $V = V_{min}$

Una vez calculadas las predicciones del modelo, se evalúa la similitud entre ellas y las mediciones reales, a partir del error cuadrático medio “ e_k ” o MSE por sus siglas en inglés.

$$e_k^2 = (|V_{01}| - |V_{01k}|)^2 + (|V_{+1}| - |V_{+1k}|)^2 + (|V_{-1}| - |V_{-1k}|)^2$$

Donde “ V_i ” corresponden a las componentes simétricas de las mediciones reales y V_{ik} a las que predice el modelo (Tabla 3-4).

A modo de resumen, se adjunta el diagrama de flujo (Figura 3-27) correspondiente al algoritmo de detección de eventos de tensión ASA, que consta de los siguientes pasos:

1. Si las tres tensiones RMS fundamentales están en el rango [0.9, 1.1] p.u, entonces el estado de la red eléctrica es normal desde el punto de vista de la detección de eventos, basados en la norma IEC 61000-4-30:2015 [49] y en la norma IEEE 1159-2009 [50]. Estas normas también definen un *swell* cuando la tensión en cualquier fase es mayor que 1.1 p.u. En cualquier otro caso, se procede al paso 2.
2. Computar V_{min} y V_{max} .

3. $V_{\max} > 1.1$ p.u entonces la red eléctrica está en estado de *swell* y se continua con el paso 4. En caso contrario ($V_{\min} < 0.9$ pu), la red eléctrica atraviesa el estado de *dip* y se continua con el paso 5.
4. Computar el MSE para el evento tipo H y para el evento tipo I entre las mediciones reales y las predicciones del modelo. Luego calcular el mínimo entre ambos para determinar el tipo de evento más probable.
5. Computar el MSE para los eventos tipo A, B, C, D, E, F y G y calcular el mínimo de todos ellos para determinar el tipo de evento más probable. Nótese que para los eventos tipo C y D se calcula el mismo error debido a que los valores de componentes simétricas del modelo (Tabla 3-4) son iguales. Lo mismo se aplica a los eventos tipo F y G. No obstante, es posible distinguir entre estos tipos de eventos de acuerdo a V_{\min} , para eso se procede al paso 6.
6. Si V_{\min} es una tensión de fase entonces el evento es tipo A, B, D o F de acuerdo al mínimo error calculado. En otro caso, el evento es tipo A, C, E o G de acuerdo al mínimo error calculado.

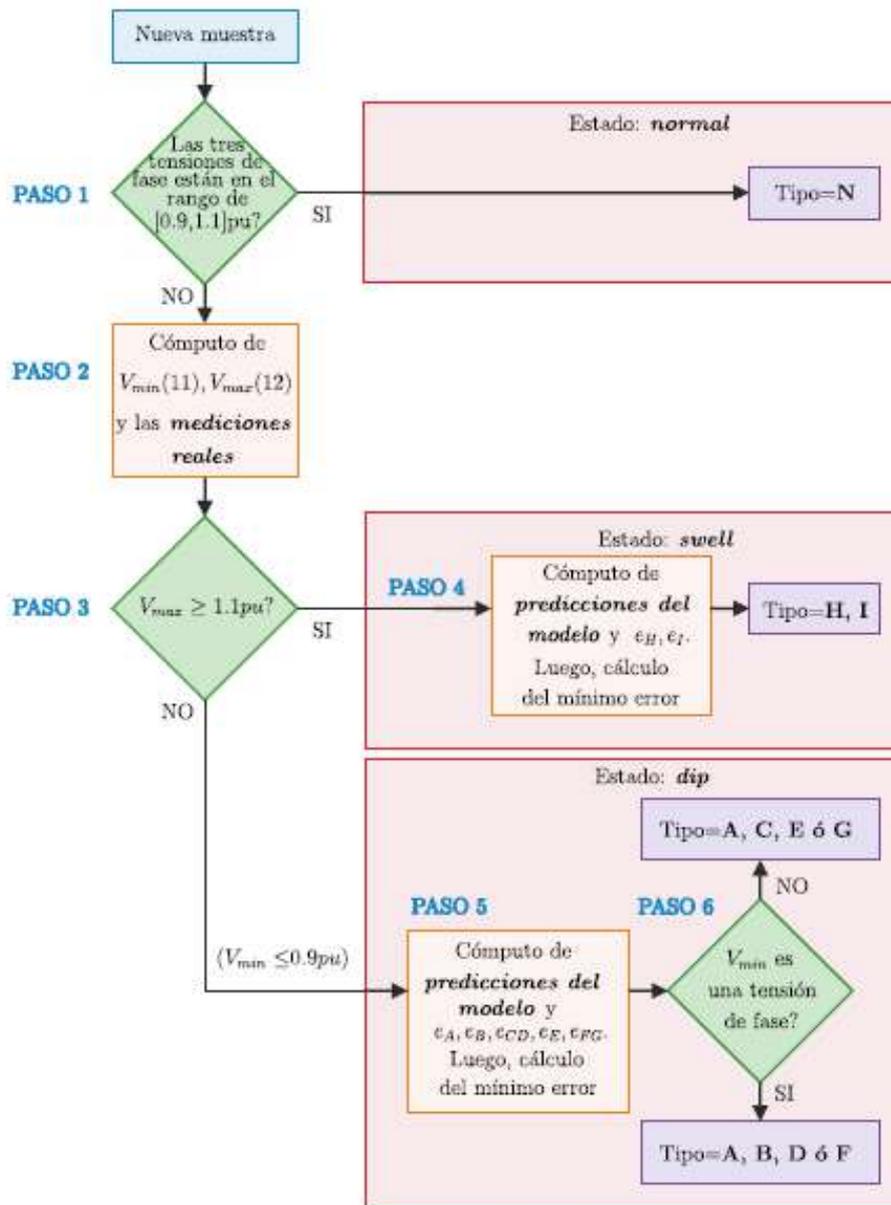


Figura 3-27. Diagrama de flujo del algoritmo ASA. [42]

En el diagrama de flujo anterior, “nueva muestra” corresponde a los resultados de tensiones RMS que se obtienen durante la simulación de las distintas fallas aplicando el algoritmo desarrollado en la sección 3.4.

De esta manera, se obtuvo por cada nodo (en total los 34 nodos) un archivo “*.csv” con los resultados del tipo de evento más probables para cada falla. El código de programación correspondiente a los cálculos desarrollados en esta sección se encuentra anexo en el apartado 6.1.5.3.

En la Figura 3-28 se muestra el diagrama de flujo utilizado para la clasificación de eventos de tensión utilizando el algoritmo ASA.

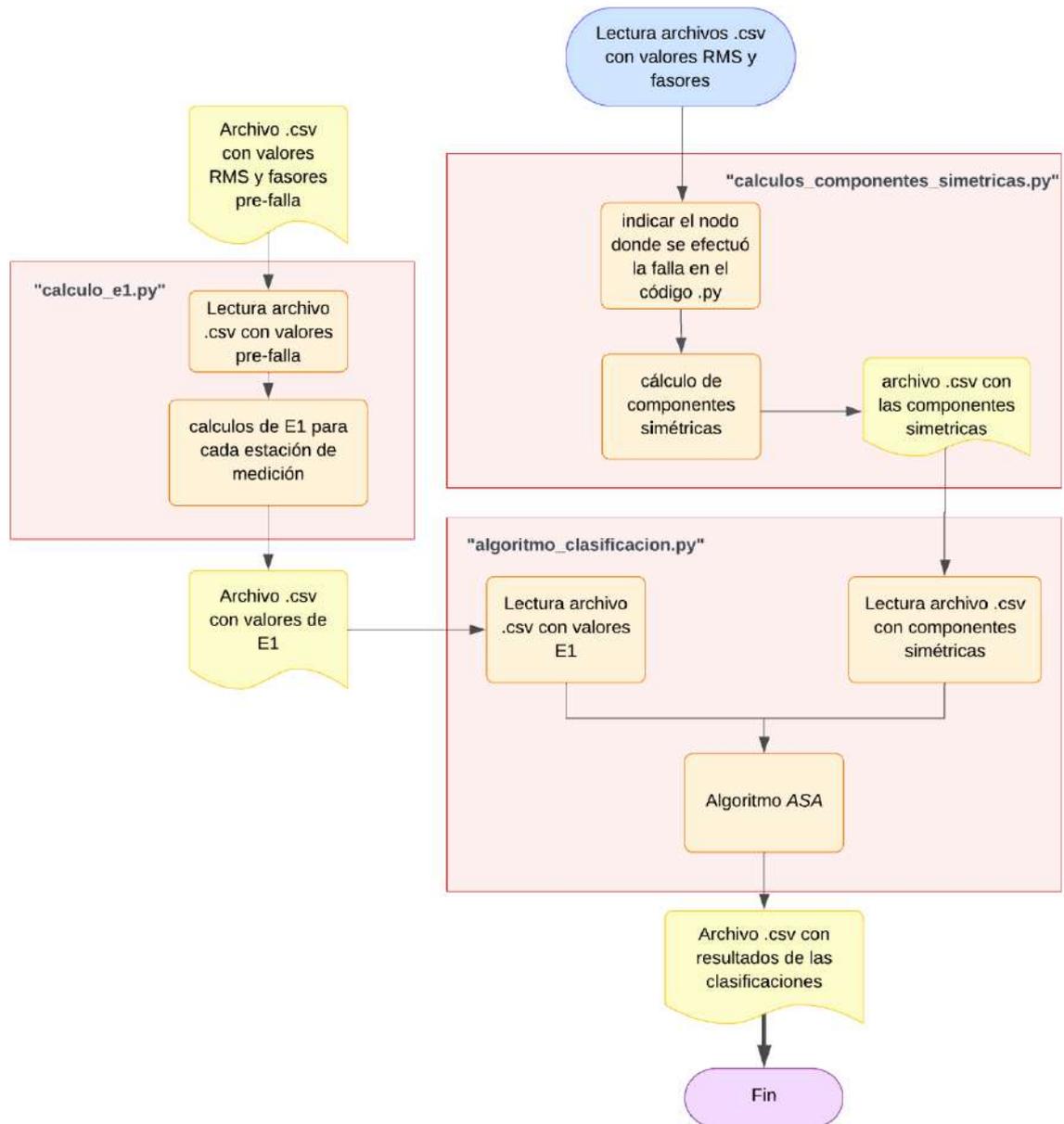


Figura 3-28. Diagrama de flujo que representa la aplicación del algoritmo ASA para clasificar eventos a partir de las fallas simuladas.

El tratamiento y análisis de los resultados obtenidos a partir de este algoritmo se desarrollarán el Capítulo 5.

3.5.3 Algoritmo de secuencias absolutas mejorado

El algoritmo eASA [63] es una mejora del algoritmo ASA desarrollado en la sección anterior, y que surgió como resultado de una serie de cambios que se explicarán a continuación.

Como se discutió en la sección anterior, la clasificación de los eventos de tensión se realiza comparando las medidas reales de tensión de la red con los modelos matemáticos de cada evento de tensión a evaluar. En el algoritmo ASA, primero se identifica la ocurrencia de

sobrevoltaje o bajo voltaje para limitar el número de modelos de eventos posibles y, en consecuencia, reducir el costo computacional. Esta primera decisión del algoritmo puede conducir a errores de clasificación. Para evitar estos errores, es más conveniente evaluar todos los modelos simultáneamente para determinar el tipo de evento de tensión que más se acerca a las señales medidas, con la desventaja de aumentar el número de cálculos necesarios para implementar el método.

Otro cambio introducido en el algoritmo eASA se detalla en la Tabla 3-6, donde se resume el modelo analítico de cada tipo de evento considerando el ángulo de fase de secuencia negativa y homopolar respecto a la secuencia positiva. La diferencia con la Tabla 3-4 es que, en este caso, se utilizan los valores de los vectores correspondientes a cada secuencia, en lugar del módulo de cada vector. Este cambio se refleja en las ecuaciones correspondientes a los eventos D y F, donde el término para la componente inversa cambió de $E_1(1-h)/2$ a $E_1(h-1)/2$ para el caso D, y $E_1(1-h)/3$ a $E_1(h-1)/3$. Este cambio se realizó por trabajar con las expresiones vectoriales.

Tabla 3-6. Componentes simétricas función de h y E_1 para cada tipo de evento utilizada por el Algoritmo eASA.

Tipo	V_{01} [pu]	V_{+1} [pu]	V_{-1} [pu]
A	0	$E_1 h$	0
B	$E_1(1-h)/3$	$E_1(2+h)/3$	$E_1(1-h)/3$
C	0	$E_1(1+h)/2$	$E_1(1-h)/2$
D	0	$E_1(1+h)/2$	$E_1(h-1)/2$
E	$E_1(1-h)/3$	$E_1(1+2h)/3$	$E_1(1-h)/3$
F	0	$E_1(1+2h)/3$	$E_1(h-1)/3$
G	0	$E_1(1+2h)/3$	$E_1(1-h)/3$
H	$E_1(h-1)$	E_1	0
I*	$2E_1(1-h)$	E_1	0
I**	$E_1/2$	$E_1(1+2h/3)$	$E_1(1-2h/3)$

Dado que ASA emplea los valores absolutos de las componentes de secuencia directa, inversa y homopolar, el método se puede aplicar a mediciones instantáneas de voltajes de red sin requerir ninguna sincronización específica. Esta característica funcional permite simplificar la implementación y reducir el error de clasificación de algunas perturbaciones, como las rotaciones de fase simétricas [51].

El algoritmo eASA mantiene esta característica funcional haciendo que el sistema de medición sea independiente del método de sincronización adoptado. Sin embargo, eASA no utiliza los valores absolutos de las componentes simétricas, sino que tiene en cuenta su signo.

Por esta razón, es necesario generar un conjunto de valores corregidos, previo al estimador *MSE*, utilizando las siguientes ecuaciones:

$$V_{+1c} = |V_{+1}|$$

$$V_{-1c} = \begin{cases} |V_{-1}| & , si \quad -90^\circ \leq \Delta\varphi_{-1} < 90^\circ \\ -|V_{-1}| & , si \quad -90^\circ \geq \Delta\varphi_{-1} > 90^\circ \end{cases}$$

$$V_{01c} = \begin{cases} |V_{01}| & , si \quad -90^\circ \leq \Delta\varphi_{01} < 90^\circ \\ -|V_{01}| & , si \quad -90^\circ \geq \Delta\varphi_{01} > 90^\circ \end{cases}$$

Donde V_{+1c} , V_{-1c} y V_{01c} son la secuencia positiva, negativa y homopolar corregida respectivamente. $\Delta\varphi_{-1}$ es la diferencia de fase entre V_{+1} y V_{-1} , y $\Delta\varphi_{01}$ es la diferencia de fase entre V_{+1} y V_{01} . Es importante notar que estas ecuaciones estiman las secuencias negativa y homopolar para que siempre estén en fase o en contrafase a la secuencia positiva. Aunque esta aproximación implica la exclusión de ángulos de fase intermedios, permite discriminar entre eventos con vectores de secuencia negativa y homopolar con rotaciones de fase entre 90° y 270° con respecto a los que no.

La aplicación de este algoritmo se realizó modificando el algoritmo “eventosdetension.py” anexo en 6.1.5.3. Se mantuvo invariante el inicio del código donde se leen las componentes simétricas calculadas previamente, y se modificó el cuerpo del algoritmo implementando los cambios nombrados para el algoritmo *eASA*. La porción de código encargado de la escritura de los resultados también se mantuvo invariante.

El diagrama de flujo utilizado para aplicar el algoritmo *eASA* resulta ser el mismo que el utilizado para el algoritmo *ASA*, mostrado en la Figura 3-28. La única diferencia radica en que en el recuadro “*Algoritmo ASA*” ahora es “*Algoritmo eASA*”.

En el Capítulo 5 se hará una comparativa de los desempeños de estos algoritmos.

3.6 Conclusión

En este capítulo se logró mostrar la totalidad de los algoritmos utilizados para las diferentes etapas de las simulaciones y cálculos.

El *Algoritmo de Inicialización* resultó ser de utilidad para parametrizar el circuito eléctrico, brindando la posibilidad de realizar modificaciones desde un archivo *csv* externo, y aplicando los cambios directamente en el archivo que *ATP* utiliza para realizar los cálculos sin tener que acudir a la interfaz gráfica de *ATPDraw* una vez creada la arquitectura del circuito.

Como una extensión de este algoritmo, surge el *Algoritmo de Convergencia*, el cual permite modificar los valores de las cargas eléctricas en forma iterativa para cargas de potencia o corriente constante. La sobreescritura de estos valores en el archivo *ATP* es igual a la que

utiliza el *Algoritmo de Inicialización*. Este algoritmo no fue probado exhaustivamente, y no se conoce su comportamiento cuando se modifican muchas cargas eléctricas en simultáneo. Esto aumenta la probabilidad de que sea necesario continuar con el desarrollo de este algoritmo para aplicarlo en gran escala.

Para simular automáticamente una gran cantidad de fallas en la red, surge el *Algoritmo de Fallas*. Este algoritmo se inicia una vez generado el archivo *ATP* con una falla monofásica, una bifásica con contacto a tierra o una bifásica sin contacto a tierra, modelada en el circuito. Luego, mediante dos códigos desarrollados en Python, modifica la impedancia de falla, ejecuta una simulación y guarda los resultados, repitiendo estos pasos n veces, según la configuración que defina el usuario. Esto permitió agilizar las simulaciones en gran medida, generando un muestreo de fallas lo suficientemente grande para evaluar el desempeño de los algoritmos de clasificación de eventos *ASA* y *eASA*.

A su vez, se consideró que la generación de una base de datos con los resultados obtenidos puede ser de mucha utilidad para futuros proyectos en los que se desee continuar con esta línea de investigación, siendo este otro de los motivos por los cuales se optó por procesar la mayor cantidad de datos posible. La totalidad de los datos resultantes de las fallas eléctricas simuladas en el alimentador de pruebas de 34 nodos están disponibles en [64]. Cada carpeta fue nombrada con el número de nodo en donde se llevó a cabo la falla eléctrica.

Finalmente, para clasificar los eventos de cada una de las fallas simuladas, se utiliza el *Algoritmo de detección de fallas*. Este algoritmo está conformado por varios códigos desarrollados en Python que ejecutan todas las acciones necesarias para la correcta implementación como la lectura de archivos *csv* generados previamente, realizar los cálculos como por ejemplo el de las componentes simétricas, aplicar los algoritmos *ASA* y *eASA* y guardar los resultados obtenidos. El criterio para desarrollar este algoritmo de clasificación de eventos de tensión fue el de tener presente que existe una variedad de Algoritmos de detección de fallas disponibles que pueden aplicarse a los resultados obtenidos. La arquitectura del algoritmo permite que esto pueda llevarse a cabo de una manera rápida, modificando fragmentos de los códigos desarrollados, por ejemplo el bloque encargado de implementar el algoritmo de clasificación de eventos, y manteniendo invariante el código que realiza acciones como el de lectura de datos y el que permite la escritura de los resultados.

Como código útil para el análisis posterior de los resultados, se destacan aquellos desarrollados para generar los gráficos fasoriales, y los que evalúan los cambios de tensiones y saltos de fase en función de la impedancia de falla.

3.7 Bibliografía

- [41] IEEE. Distribution System Analysis Subcommittee. (2017) IEEE 34 Node Test Feeder. Recuperado de: <https://cmte.ieee.org/pes-testfeeders/wp-content/uploads/sites/167/2017/08/feeder34.zip>
- [42] Strack J. L. (2015) Identificación y clasificación automática de eventos de tensión en redes eléctricas inteligentes. Universidad Nacional de Mar del Plata.
- [43] Cmath: Mathematical functions for complex numbers. Recuperado de: <https://docs.python.org/3/library/cmath.html>
- [44] Matplotlib (2012). Matplotlib: Visualization with Python. Recuperado de: <https://matplotlib.org/>
- [45] M. H. J. Bollen, (1999) Understanding Power Quality Problems: Voltage Sags and Interruptions. Wiley - IEEE Press.
- [46] V. Ignatova (2006). Méthodes d'analyse de la qualité de l'énergie électrique. Application aux creux de tension et a' la pollution harmonique, Ph.D. thesis, Université Joseph Fourier. <https://theses.hal.science/tel-00170713>
- [47] C. L. Fortescue (1918). *Method of symmetrical co-ordinates applied to the solution of polyphase networks*. Transactions of the American Institute of Electrical Engineers, vol. XXXVII, no. 2, pp. 1027-1140, July 1918. Recuperado de: <https://doi.org/10.1109/T-AIEE.1918.4765570>
- [48] Gustavo L. Ferro (2015). *El método de las Componentes simétricas*. Facultad de Ingeniería (U.N.M.D.P), Departamento de Ingeniería Eléctrica, Área Electrotecnia.
- [49] IEC 61000-4-30:2015-02. *Electromagnetic compatibility (EMC) - Part 4-30: Testing and measurement techniques - Power quality measurement methods*. IEC Std. (2015). Recuperado de: <https://webstore.iec.ch/publication/21844>
- [50] IEEE Std 1159-2009. *IEEE Recommended Practice for Monitoring Electric Power Quality*. IEEE Std. (2009).
- [51] J. L. Strack, I. Carugati, C. M. Orallo, S. O. Maestri, P. G. Donato, M. A. Funes. (2019). *Three-phase voltage events classification algorithm based on an adaptive threshold*. Electric Power Systems Research 172 (2019) 167-176. <https://doi.org/10.1016/j.epsr.2019.03.012>

[52] Power System Toolbox, Página web de ATPDraw – sección Help. Recuperado de <https://www.atpdraw.net/help/hs2030101.htm>

[53] Cooley, James W., Tukey, John W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90), 297–297. <https://doi.org/10.1090/S0025-5718-1965-0178586-1>

CAPÍTULO 4

VALIDACIÓN DE RED

ALIMENTADOR DE PRUEBA DE 34 NODOS



4 Alimentador de prueba de 34 nodos

Resumen

El objetivo de este capítulo es el de validar la red eléctrica simulada, el alimentador de pruebas de 34 nodos, a partir de la información brindada por la IEEE. Se desarrollarán los modelos elegidos para realizar las simulaciones, los datos cargados en el software *ATPDraw*, criterios tomados y las simplificaciones aplicadas. Luego se mostrarán los resultados de la simulación y se realizará un análisis comparativo con la red de eléctrica de referencia. A partir de estos resultados se obtendrán conclusiones y propuestas de mejora del modelo.

4.1 Características de la red

En este trabajo se decidió trabajar con el alimentador de pruebas de 34 nodos provisto por la IEEE, siendo una red modelada a partir de una red eléctrica real de EE.UU. La información para generar el modelo de la red fue extraída de *K. P. Schneider et al* [54] y será presentada a lo largo del presente capítulo.

El esquema de la red eléctrica se presenta en la Figura 4-1.

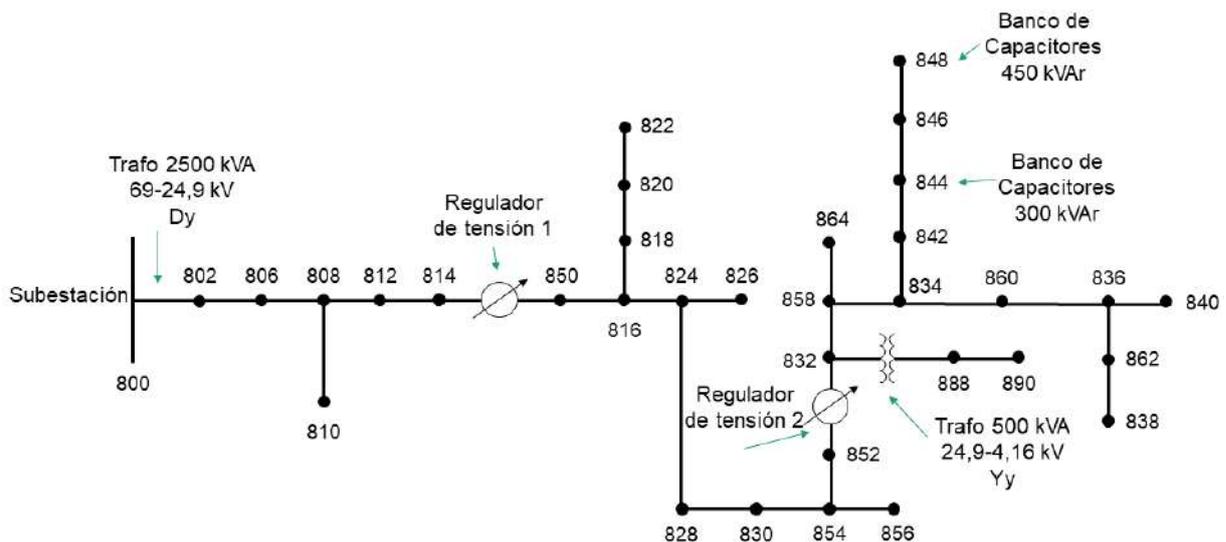


Figura 4-1. Esquema del circuito eléctrico simulado: Alimentador de pruebas de 34 nodos. [54]

Esta red es del tipo radial, siendo alimentada desde el nodo 800 (*Subestación*). A su vez, está compuesta por dos transformadores de potencia, dos reguladores de tensión, dos bancos de capacitores y un total de 25 cargas eléctricas (monofásicas, bifásicas y trifásicas). En la Figura 4-1 se pueden apreciar los 34 nodos que componen a la red junto a su numeración o *ID*. La *ID* de los nodos se utiliza para ubicar los elementos nombrados y, para el caso de las líneas eléctricas, permite conocer qué tipo de línea es (trifásica o monofásica) y cuáles son sus características físicas como, por ejemplo, su longitud.

4.1.1 Líneas eléctricas

Para crear las líneas eléctricas se utilizó la información que se adjunta siguiente tabla. En la misma se indica el inicio y fin del segmento de línea (Nodo A a Nodo B), su longitud y el tipo de configuración de la línea (Columna “*Config.*”). La codificación de los tipos de líneas se adjunta en la Tabla 4-2, extraída de *Kersting* [55].

Tabla 4-1. Longitud de segmentos de línea de red de 34 nodos.

Nodo A	Nodo B	Longitud (m)	Config.
800	802	786.4	300
802	806	527.3	300
806	808	9823.7	300
808	810	1769.1	303
808	812	11430.0	300
812	814	9061.7	300
814	850	3.0	301
816	818	521.2	302
816	824	3112.0	301
818	820	14676.1	302
820	822	4188.0	302
824	826	923.5	303
824	828	256.0	301
828	830	6230.1	301
830	854	158.5	301
832	858	1493.5	301
832	888	0.0	XFN-1 (Trafo 2)
834	860	615.7	301
834	842	85.3	301
836	840	262.1	301
836	862	85.3	301
842	844	411.5	301
844	846	1109.5	301
846	848	161.5	301
850	816	94.5	301
852	832	3.0	301
854	856	7111.0	303
854	852	11225.8	301
858	864	493.8	302
858	834	1777.0	301
860	836	816.9	301

862	838	1481.3	304
888	890	3218.7	300

Tabla 4-2. Configuraciones de líneas aéreas.

Config.	Phasing	Phase	Neutral	Spacing ID
300	B A C N	1/0	1/0	500
301	B A C N	#2 6/1	#2 6/1	500
302	A N	#4 6/1	#4 6/1	510
303	B N	#4 6/1	#4 6/1	510
304	C N	#2 6/1	#2 6/1	510

Nota: La columna 2 (Phasing) indica si la línea es trifásica o monofásica (y la fase correspondiente). Las columnas 3 (Phase) y 4 (Neutral) indican el código del material de los conductores. La columna 5 (Spacing) indica el código correspondiente al tipo de disposición de las líneas en el poste. Ver Kersting [55].

Como se observa en la Tabla 4-2, en esta red eléctrica se encuentran cinco tipos de líneas eléctricas que se diferencian entre sí por su distribución espacial y por ser trifásicas o monofásicas. Para la línea trifásica existen dos modelos, mientras que para las líneas monofásicas se utilizan tres modelos. En Kersting [55] se explica en detalle cómo interpretar esta información, y en la Figura 4-2 se muestran tres disposiciones espaciales diferentes con sus respectivos *spacing ID* (columna 5 de la Tabla 4-2):

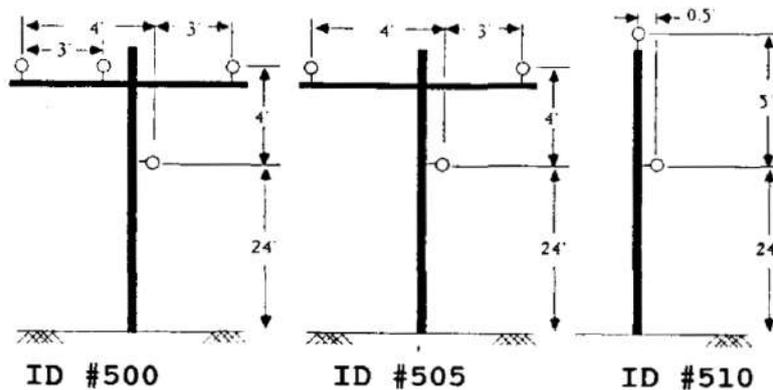


Figura 4-2: Líneas aéreas con sus respectivos *Spacing ID*. [55]

En este trabajo, las líneas eléctricas fueron modeladas utilizando el modelo Pi Nominal a partir de la información de las matrices de impedancia brindadas por la IEEE.:

Tabla 4-3. Configuración 300. Matriz Z (R+jX [Ω por milla]) a la izquierda y matriz B [μ S por milla] a la derecha.

Z (R+jX) [Ω por milla]			B [μ S por milla]		
1.3368+j1.3343	0.2101+j0.5779	0.2130+j0.5015	5.3350	-1.5313	-0.9943
	1.3238+j1.3569	0.2066+j0.4591		5.0979	-0.6212
		1.3294+j1.3471			4.8880

Tabla 4-4. Configuración 301. Matriz Z (R+jX [Ω por milla]) a la izquierda y matriz B [μ S por milla] a la derecha.

Z (R+jX) [Ω por milla]			B [μ S por milla]		
1.9300+j1.4115	0.2327+j0.6442	0.2359+j0.5691	5.1207	-1.4364	-0.9402
	1.9157+j1.4281	0.2288+j0.5238		4.9055	-0.5951
		1.9219+j1.4209			4.7154

Tabla 4-5. Configuración 302. Matriz Z (R+jX [Ω por milla]) a la izquierda y matriz B [μ S por milla] a la derecha.

Z (R+jX) [Ω por milla]			B [μ S por milla]		
2.7995+j1.4855	0	0	4.2251	0	0
	0	0		0	0
		0			0

Tabla 4-6. Configuración 303. Matriz Z (R+jX [Ω por milla]) a la izquierda y matriz B [μ S por milla] a la derecha.

Z (R+jX) [Ω por milla]			B [μ S por milla]		
0	0	0	0	0	0
	2.7995+j1.4855	0		4.2251	0
		0			0

Tabla 4-7. Configuración 304. Matriz Z (R+jX [Ω por milla]) a la izquierda y matriz B [μ S por milla] a la derecha.

Z (R+jX) [Ω por milla]			B [μ S por milla]		
0	0	0	0	0	0
	1.9217+j1.4212	0		4.3637	0
		0			0

Nota: Las longitudes de línea se trabajaron en metros, por lo tanto, fue necesario realizar la conversión a [Ω por metro] y [μ S por metro] previo a cargar los datos en ATPDraw.

Estos valores de impedancia fueron cargados en el Modelo PI, disponible en ATPDraw y que fue desarrollado en el Capítulo 2 de este trabajo, utilizando la misma metodología para la totalidad de las líneas de esta red eléctrica.

En la Figura 4-3 se muestra de ejemplo el tramo de línea 828-830 de configuración 301. Se observan algunos de los valores de impedancia cargados y la longitud del tramo. En este tramo, caracterizado por tener una carga distribuida, fue dividido en dos tramos de longitudes iguales. Esto quiere decir que se insertaron dos *Modelos Pi* en serie asignando la mitad de la distancia a cada uno. En la Sección 4.1.2 se explica en detalle el criterio tomado para las cargas distribuidas.

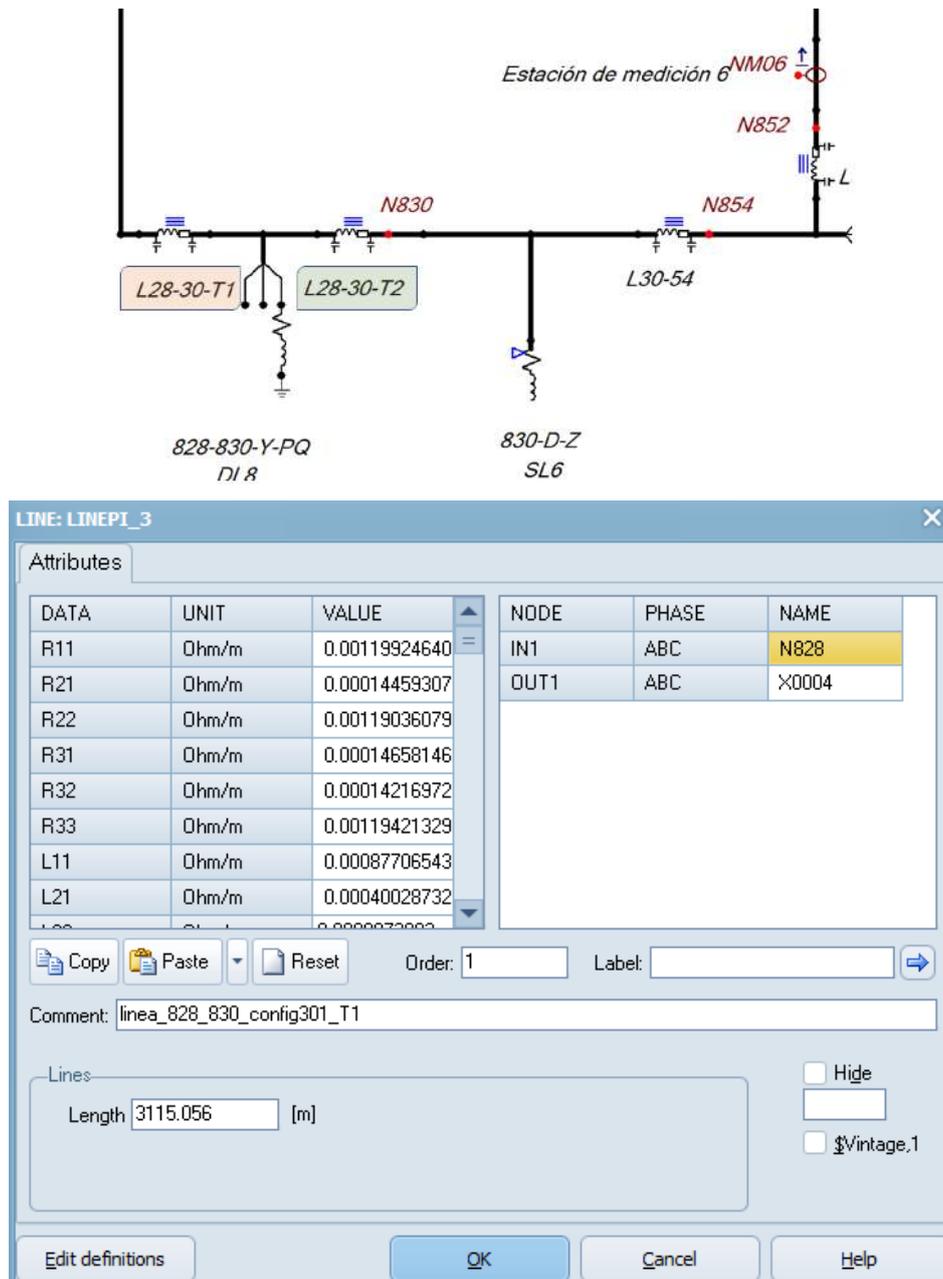


Figura 4-3. Ejemplo de un tramo eléctrico utilizando el Modelo PI en ATPDraw. En la figura de arriba se muestran los dos tramos que componen a la línea 828-830 y la carga distribuida "DL8".

4.1.2 Cargas eléctricas

Las cargas pertenecientes a la red fueron modeladas a partir de la información brindada por la IEEE, adjuntada en la Tabla 4-8 (cargas distribuidas) y la Tabla 4-9 (cargas puntuales).

Para este trabajo, se modeló a la carga distribuida como una carga puntual de valor total ubicada en el medio del tramo de línea correspondiente, considerando la información brindada por Kersting [56]. En el *Capítulo 3.4 del libro Uniformly Distributed Loads* [57] se analizan en detalle diferentes estrategias para modelar cargas distribuidas. Estas estrategias

varían según la información disponible de la distribución de las cargas, la precisión que se desea obtener y el fenómeno que se quiera estudiar en la red simulada.

El modelo seleccionado para simular este tipo de cargas es el que representa en buena medida la caída de tensión en la línea cuando hay presentes cargas distribuidas uniformes. Partiendo de una línea con las siguientes cargas distribuidas:

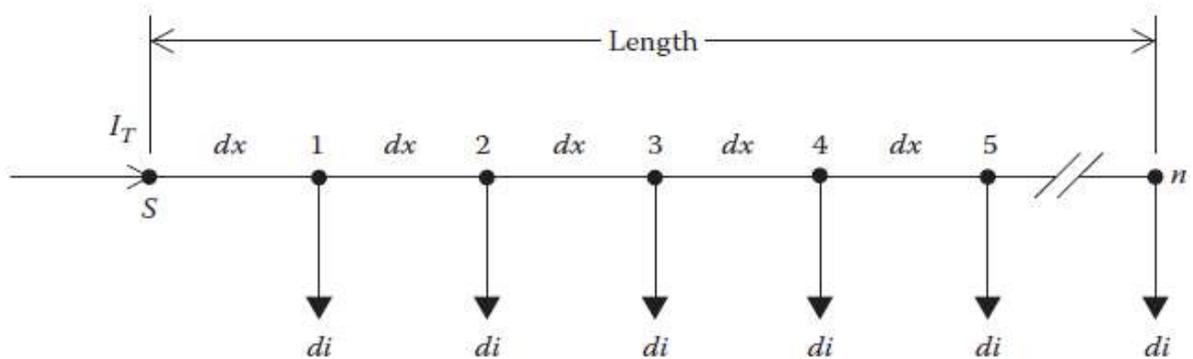


Figura 4-4. Línea con n cargas distribuidas uniformes. [57]

Siendo:

l = longitud del alimentador

$z = r + jx$ = impedancia de la línea en Ω /milla

dx = longitud de cada sección de línea

di = corrientes de carga en cada nodo

n = número de nodos y número de tramos de línea

I_T = corriente total en el alimentador

Las corrientes de carga están dadas por:

$$d_i = \frac{I_T}{n}$$

La caída de tensión en el primer segmento de línea viene dada por:

$$v_{drop1} = Re\{z dx(n di)\}$$

La caída de tensión en el segundo segmento de línea viene dada por:

$$v_{drop2} = Re\{z dx[(n - 1)di]\}$$

Para una línea con n segmentos la caída de voltaje total desde el nodo fuente hasta el último nodo viene dada por:

$$v_{drop\ total} = Re\{z dx di [n + (n - 1) + (n - 2) + \dots + (1)]\}$$

Aplicando una serie matemática y simplificando términos se obtiene:

$$v_{drop\ total} = Re \left\{ z\ dx\ di \left[\frac{n(n+1)}{2} \right] \right\}$$

Finalmente, definiendo a la distancia incremental como:

$$dx = \frac{l}{n}$$

Y la corriente incremental como:

$$di = \frac{I_T}{n}$$

Reemplazando y haciendo tender n a infinito se obtiene la caída total en la línea:

$$v_{drop\ total} = Re \left\{ \frac{1}{2} Z I_T \right\}$$

Con $Z=z \cdot l$ [Ω]

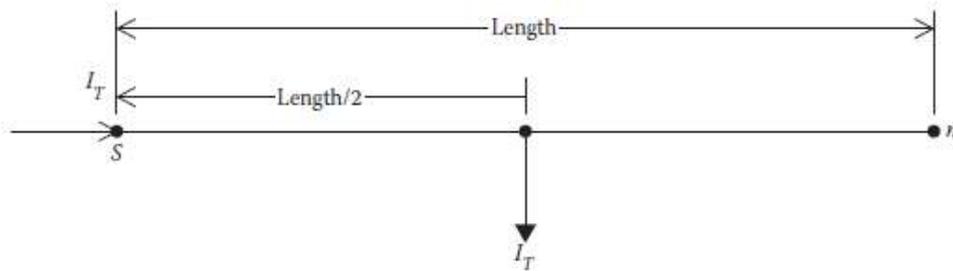


Figura 4-5. Modelo obtenido para representar cargas distribuidas uniformes como si se tratase de una única carga de manera de obtener la misma caída de tensión en la línea. [57]

En la siguiente figura se resumen los comportamientos de las caídas de tensión y las pérdidas en las líneas dependiendo el tipo de distribución de las cargas:

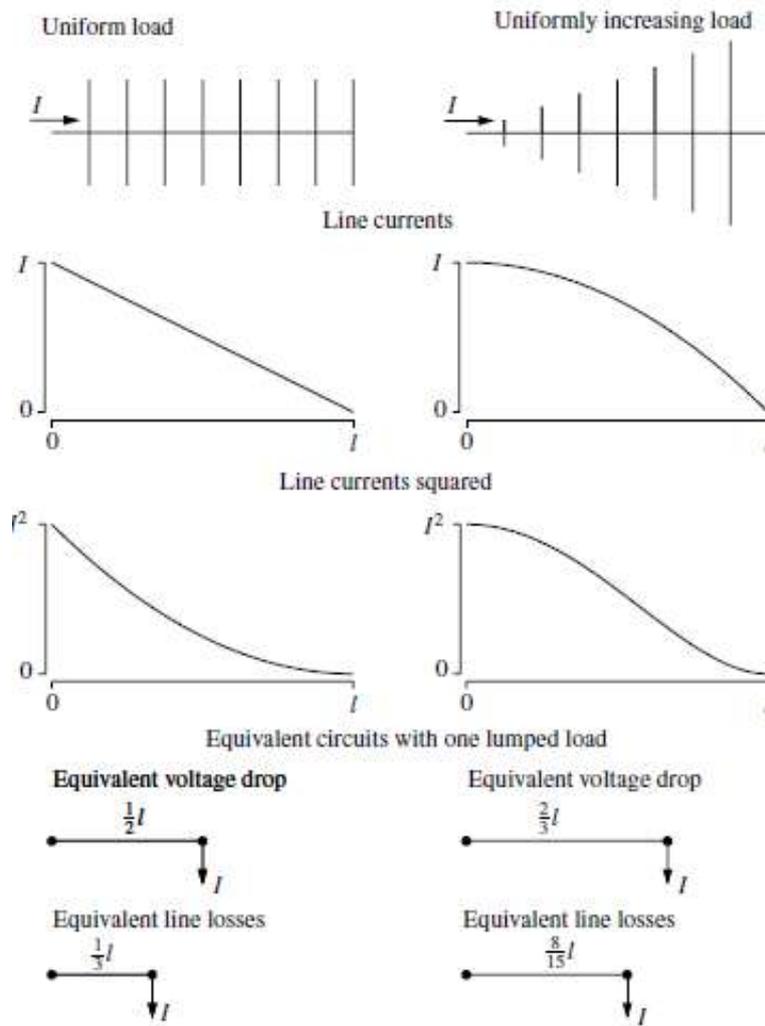


Figura 4-6. Comportamiento en las caídas de tensión y las pérdidas en las líneas según el tipo de carga distribuida. [59]

En base a lo desarrollado, los tramos de las líneas eléctricas que alimentan cargas distribuidas fueron divididas en dos segmentos de longitudes iguales entre sí e iguales a la mitad de la longitud total del tramo y se situó a la carga en el centro de estas líneas (Ver Figura 4-3).

Tabla 4-8. Potencia activa y reactiva por fase de las cargas distribuidas y su tipo de conexión.

Nodo A	Nodo B	Conexión	Fase A P [kW]	Fase B Q [kVAr]	Fase C P [kW]	Fase A P [kW]	Fase B Q [kVAr]	Fase C P [kW]
802	806	Y-PQ	0	0	30	15	25	14
808	810	Y-I	0	0	16	8	0	0
818	820	Y-Z	34	17	0	0	0	0
820	822	Y-PQ	135	70	0	0	0	0
816	824	D-I	0	0	5	2	0	0
824	826	Y-I	0	0	40	20	0	0

824	828	Y-PQ	0	0	0	0	4	2
828	830	Y-PQ	7	3	0	0	0	0
854	856	Y-PQ	0	0	4	2	0	0
832	858	D-Z	7	3	2	1	6	3
858	864	Y-PQ	2	1	0	0	0	0
858	834	D-PQ	4	2	15	8	13	7
834	860	D-Z	16	8	20	10	110	55
860	836	D-PQ	30	15	10	6	42	22
836	840	D-I	18	9	22	11	0	0
862	838	Y-PQ	0	0	28	14	0	0
842	844	Y-PQ	9	5	0	0	0	0
844	846	Y-PQ	0	0	25	12	20	11
846	848	Y-PQ	0	0	23	11	0	0

Para el caso de las cargas puntuales, en la primera columna de la Tabla 4-9 solo se indica un único nodo que representa el punto de conexión de la carga.

Tabla 4-9. Potencia activa y reactiva por fase de las cargas puntuales y su tipo de conexión.

Nodo	Conexión	Fase A		Fase B		Fase C	
		P [kW]	Q [kVAr]	P [kW]	Q [kVAr]	P [kW]	Q [kVAr]
860	Y-PQ	20	16	20	16	20	16
840	Y-I	9	7	9	7	9	7
844	Y-Z	135	105	135	105	135	105
848	D-PQ	20	16	20	16	20	16
890	D-I	150	75	150	75	150	75
830	D-Z	10	5	10	5	25	10

Tanto para las cargas distribuidas como para las puntuales, figuran los valores de potencias activas y reactivas por fase y el tipo de conexión de la carga (estrella/triángulo).

4.1.3 Banco de capacitores

La red cuenta con dos bancos de capacitores ubicados en los nodos 844 y 848, respectivamente. La IEEE no provee mayor información que la detallada en la Tabla 4-10, es decir, que la potencia reactiva por fase. Por tal motivo, se supuso que durante la simulación estos bancos de capacitores presentan un comportamiento estático, sin contar con un control de *taps* para regular la potencia reactiva y se encuentran aportando a la red la potencia reactiva nominal mostrada en la tabla. Este criterio se considera válido teniendo en cuenta los tiempos de simulación para cada falla, siendo aproximadamente 10 ciclos post-falla.

El modelo seleccionado de *ATPDraw* para representar a estos capacitores fue el de una carga trifásica capacitiva en estrella. Se le agregó una pequeña impedancia resistiva en serie para representar el conexionado a la red de valor 0,1 [Ω] por fase.

Tabla 4-10. Datos de potencia reactiva de los capacitores de la red de 34 nodos.

Nodo	Fase A [kVAr]	Fase B [kVAr]	Fase C [kVAr]
844	100	100	100
848	150	150	150

4.1.4 Transformadores

Esta red cuenta con dos transformadores, uno al inicio que representa a la subestación encargada de alimentar a toda la red eléctrica (tipo radial) y otro transformador entre los nodos 832 y 888 (Ver Figura 4-1). Los datos de potencia, tensiones y valores de impedancia porcentuales se muestran en la Tabla 4-11.

Tabla 4-11. Parámetros del transformador.

ID	kVA	kV-high	kV-low	R - %	X - %
Subestación	2500	69 - D	24.9 -Gr. W	1	8
XFM -1	500	24.9 - Gr.W	4.16 - Gr. W	1.9	4.08

El modelo de *ATPDraw* seleccionado para simular estos transformadores fue el transformador híbrido. Como se detalló en el Capítulo 2 de este trabajo, no se buscó modelar con mucha complejidad a estos transformadores ya que no se están estudiando fenómenos transitorios y de elevada frecuencia en donde el modelo del transformador tiene un papel clave sobre los resultados de la simulación. Más bien, el aporte de estos transformadores es el de propagar los eventos de tensión simulados, por ejemplo, eliminando la componente homopolar si la conexión del transformador lo permite, saturándose al efectuarse la falla, modificando los niveles de tensión, etc.

Para el caso de la Subestación, la tensión del lado de baja no fue 24,9 kV tensión de línea, sino que fue 26,145 kV tensión de línea (un 5 % más al valor que figura en la Tabla 4-11). Se definió ese valor porque en los perfiles de tensión entregados por la IEEE (Ver Tabla 4-15) para el nodo 800 coincidente con la subestación, se indica que la tensión en las tres fases es de 1,05 [p.u.]. De esta manera, en la simulación se supone que el transformador tiene taps de elevación de tensión, que otorgan ese aumento del 5% de tensión en el lado de baja.

Para el segundo transformador de ID XFM-1 no se aplicaron diferencias en los valores de tensión, manteniéndose la relación de transformación que se define en la Tabla 4-11.

Para utilizar el modelo Híbrido de *ATP* se ingresaron los siguientes valores:

- Número de fases: 3
- Número de devanados: 2
- Tipo de núcleo: 3 piernas
- Frecuencia: 60 [Hz] (la red de alimentación simulada es una red real de EE.UU)
- Valores de inductancia y resistencia porcentuales: Los valores detallados en la Tabla 4-11.
- Núcleo: en ambos casos se utilizaron los datos brindados por la rutina de *ATP*.
- Rating y conexiones: Datos detallados en la Tabla 4-11.

The figure shows two screenshots of the ATPDraw software interface, each divided into 'Structure' and 'Ratings & connections' sections.

Top Screenshot (Transformer ID XFM-1):

- Structure:**
 - Number of phases: 3
 - Number of windings: 2
 - Type of core: 3-leg stacked
 - Test frequency [Hz]: 60
 - Data based on: Ind., Res., Cap., Core (all selected)
 - Design param.: (unselected)
 - Test report: (unselected)
 - Typical values: (selected)
- Ratings & connections:**

	Prim.	Sec.
L-L voltage [kV]	69	26.145
Power [MVA]	2.5	2.5
Connections	D	Y
Phase shifts	0	0
Node names	X0044	S800

 - Winding sequence: core-inner-outer (selected)
 - Ext. neutral connections: (unselected)
 - Hide core nodes: (checked)

Bottom Screenshot (Transformer ID XFM-1):

- Structure:**
 - Number of phases: 3
 - Number of windings: 2
 - Type of core: 3-leg stacked
 - Test frequency [Hz]: 60
 - Data based on: Ind., Res., Cap., Core (all selected)
 - Design param.: (unselected)
 - Test report: (unselected)
 - Typical values: (selected)
- Ratings & connections:**

	Prim.	Sec.
L-L voltage [kV]	24.9	4.16
Power [MVA]	0.5	0.5
Connections	Y	Y
Phase shifts	0	0
Node names	N832	N888

 - Winding sequence: core-inner-outer (selected)
 - Ext. neutral connections: (unselected)
 - Hide core nodes: (checked)

Figura 4-7. Datos ingresados en ATPDraw de los transformadores. Arriba: datos de transformador de ID Subestación. Abajo: datos de transformador de ID XFM-1

Los dos transformadores cuentan con su neutro conectado a tierra. Para suponer una puesta a tierra en las subestaciones de cada transformador, se agregó una impedancia resistiva entre el neutro de los transformadores y tierra. No se calculó una puesta a tierra de la supuesta subestación en donde se encuentran los transformadores. Sin embargo, se usaron de base las prácticas recomendadas en *IEEE Std 142-2007* [58] en donde se describen los procedimientos recomendados para la puesta a tierra del sistema para sistemas de energía industriales y comerciales, y la selección y aplicación adecuadas de las impedancias de puesta a tierra. Entre los diferentes criterios desarrollados en el documento, se optó por seleccionar el criterio de puesta a tierra de baja resistencia, de manera de permitir que fluya una corriente de falla a tierra más alta para obtener suficiente corriente para la operación selectiva del relé. En este caso, no se están simulando elementos de protección en la red,

pero sí se busca generar huecos de tensión a partir de fallas, por lo que limitar la corriente de falla no es una prioridad.

En general, para estos valores de tensiones suelen utilizarse resistencias de puesta a tierra neutra (*Neutral Grounding Resistors* o *NGR* por sus siglas en inglés) cuyos valores vienen normalizados y cuyas resistencias vienen integradas dentro de cabinas construidas bajo los estándares correspondientes, por ejemplo, IEEE-C57.32. En ambos transformadores, la impedancia resistiva de neutro a tierra asignada fue de 1 [Ω], como se muestra en la Figura 4-8.

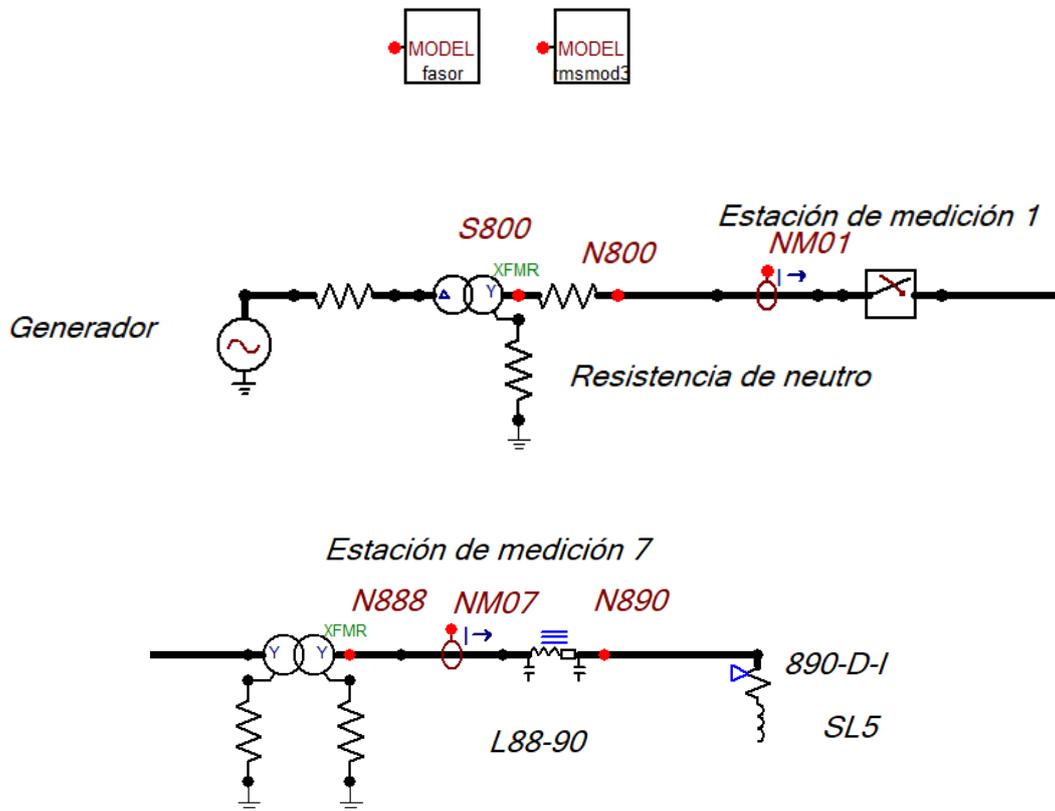


Figura 4-8. Imágenes de los transformadores de la red simulada. Arriba: subestación en el nodo 800. Abajo: transformador de ID XFM-1.

4.1.5 Reguladores de tensión

La información que proporciona la IEEE para modelar estos dos reguladores de tensión se adjunta en Tabla 4-12. A continuación, se describirá brevemente como interpretar estos datos:

Tabla 4-12. Parámetros de los dos reguladores de tensión.

Regulador 1			
	Segmento de línea	814 - 850	
	Ubicación	814	
	Fases	A - B -C	
	Conexión	3-Ph,LG	
	Fases monitoreadas	A-B-C	
	Ancho de banda	2.0 volts	
	PT Ratio	120	
	Primary CT Rating	100	
Datos de compensación	Ph-A	Ph-B	Ph-C
R	2.7	2.7	2.7
X	1.6	1.6	1.6
Voltaje	122	122	122
Regulador 2			
	Segmento de línea	852 - 832	
	Ubicación	852	
	Fases	A - B -C	
	Conexión	3-Ph,LG	
	Fases monitoreadas	A-B-C	
	Ancho de banda	2.0 volts	
	PT Ratio	120	
	Primary CT Rating	100	
Datos de compensación	Ph-A	Ph-B	Ph-C
R	2.5	2.5	2.5
X	1.5	1.5	1.5
Voltaje	124	124	124

Las primeras dos filas indican el sitio donde están instalados estos reguladores usando como referencia el número de nodo. Así, el primer regulador se encuentra entre los nodos 814 y 850 y el segundo regulador se encuentra entre los nodos 852 y 832.

Los siguientes datos están relacionados con el *Control del regulador de voltaje* desarrollado en el *Capítulo 1.1.3*. El ancho de banda o *Bandwidth* es el rango de voltaje total alrededor del voltaje establecido *Set Voltage* o *SV* que el control considerará aceptable. Es decir que el control del regulador no enviará ninguna señal para modificar el estado de los *taps* si el valor de tensión medido se encuentra en \pm el valor seteado. Los datos de *PT Ratio* y *Primary CT*

rating son las relaciones de transformación de los transformadores de medición de tensión y corriente respectivamente. En la Figura 4-9 se esquematiza el circuito de regulación:

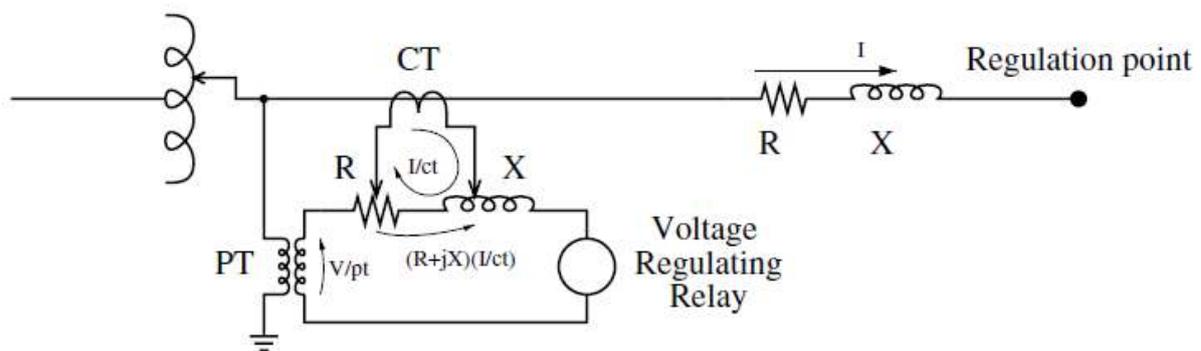


Figura 4-9. Circuito de control del regulador de tensión. [59]

Los datos R y X hacen referencia a que estos reguladores de tensión tienen un control, llamado compensador de caída de línea, para aumentar la compensación de tensión en la línea durante cargas pesadas, que se controla variando los valores R y X . El controlador ajusta los niveles de los *taps* en función del voltaje en el relé regulador de voltaje (*Voltage Regulating Relay*), que es el voltaje del *PT* sumado al voltaje en el circuito compensador de caída de línea. Sin compensación, el relé regulador de voltaje ajusta las tomas según el voltaje del *PT*. *Short* [59] detalla el funcionamiento de este controlador, presentando comparativas entre un regulador sin controlador y otro con controlador y presenta las ecuaciones para obtener los valores de R y X óptimos para diferentes escenarios. Estos reguladores tienen tiempos de actuación (*Time delay*) del orden de 30 a 60 segundos. Esto significa que, si el nivel de tensión está fuera del rango de ancho de banda seteado, el controlador envía el orden de cambiar de tomas para modificar el valor de tensión luego de transcurrido el *tiempo de delay* como se observa en la Figura 4-10.

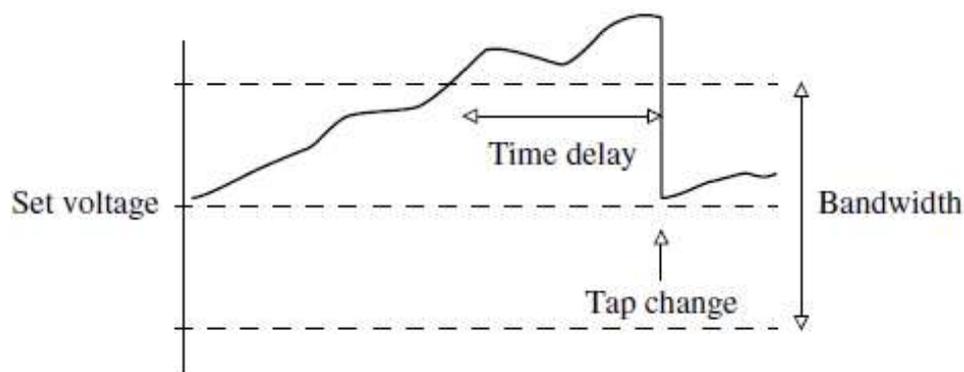


Figura 4-10. Pasos de actuación de un regulador de tensión. [59]

A partir de este análisis, se decidió definir a los regulares de tensión como un elemento estático, ya que el tiempo asociado al control de los *taps* se encuentra por encima de dos órdenes de magnitud si se lo compara con los tiempos de simulación. Es decir, durante la

simulación mantuvieron sus niveles de *taps* constantes y el nivel de regulación se obtuvo de la siguiente tabla proporcionada por la IEEE:

Tabla 4-13. Posición de *taps* para los dos reguladores de tensión simulados.

Fase	Volt Hold	R-Volt	X-Volt	PT Ratio	CT Rate	TAP
Regulador 1						
A	122.0	2.7	1.6	120.0	100.0	12
B	122.0	2.7	1.6	120.0	100.0	5
C	122.0	2.7	1.6	120.0	100.0	5
Regulador 2						
A	124.0	2.5	1.5	120.0	100.0	13
B	124.0	2.5	1.5	120.0	100.0	11
C	124.0	2.5	1.5	120.0	100.0	12

En la última fila se muestran los *taps* activos para cada fase. Estos reguladores tienen un total de 32 *taps*, 16 en aumento y 16 en descenso, generando una variación de tensión de $\pm 10\%$. Es decir, cada nivel de *tap* aporta un 5/8% de tensión, que en la escala de 120 V sobre los que están definidos, es un valor de 0,75 V por *tap* [59] [60]. A partir de esta información se calcula el porcentaje de elevación de tensión para cada fase resumida en la siguiente tabla:

Tabla 4-14. Porcentajes de elevación de tensión a partir de la cantidad de *taps* para cada regulador.

Fase	TAP	Porcentaje de tensión
Regulador 1		
A	12	+3.75%
B	5	+1.5625%
C	5	+1.5625%
Regulador 2		
A	13	+4.0625%
B	11	+3.4375%
C	12	+3.75%

Se decidió modelar a los reguladores de tensión como transformadores monofásicos, cuya relación de transformación eleve la tensión del lado secundario en el porcentaje indicado en la tercera fila de la Tabla 4-14. Es decir, cada regulador de tensión está representado como 3 transformadores monofásicos, uno por fase (Figura 4-11). Esta simplificación se considera válida para los tiempos de simulación utilizados en este trabajo.

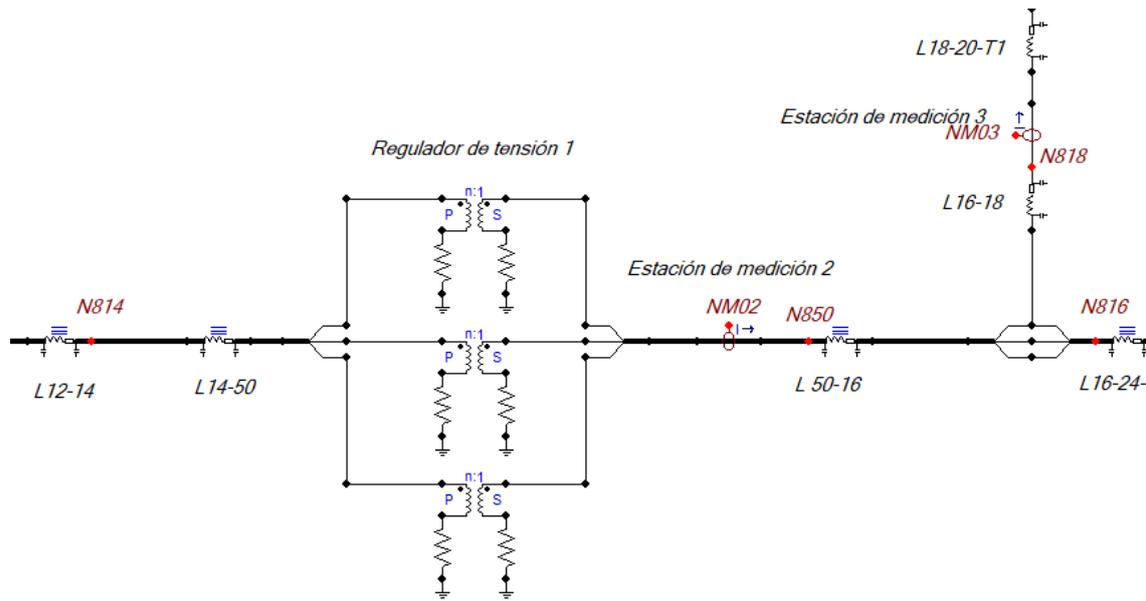


Figura 4-11. Modelado de regulador de tensión utilizado.

4.2 Resultados en régimen estacionario

Para realizar la validación de la red eléctrica de 34 nodos simulada, se realizaron una serie de mediciones en régimen estacionario (pre-falla) y se compararon los resultados con la información disponible que proporciona la IEEE.

Para ello, se hizo uso de los algoritmos desarrollados en el Capítulo 3, más precisamente de los desarrollados en la sección 3.4.1 *Cálculo de valores RMS y fasores con MODELS*. Los resultados se resumen en la Tabla 4-15 y Tabla 4-16 para los valores de tensiones obtenidas en los 34 nodos de la red, y la Tabla 4-17 y la Tabla 4-18 para los resultados de las corrientes en las 10 estaciones de medición. En el caso de las corrientes, solo se realizaron mediciones en estos 10 puntos de la red por considerarse que esta información ya era representativa del sistema.

4.2.1 Resultados de las tensiones

Tabla 4-15. Comparativa de los valores de tensión RMS en [p.u] obtenidos en las simulaciones y los valores brindados por la IEEE.

Nodo	Tensión de fases [p.u]						Error porcentual %		
	IEEE			APTDrow			Fase A	Fase B	Fase C
	Fase A	Fase B	Fase C	Fase A	Fase B	Fase C			
800	1,05	1,05	1,05	1,0435	1,0502	1,055	-0,65	0,02	0,5
802	1,0475	1,0484	1,0484	1,0407	1,0486	1,0534	-0,68	0,02	0,5
806	1,0457	1,0474	1,0474	1,0391	1,0475	1,0523	-0,66	0,01	0,49

808	1,0136	1,0296	1,0289	1,0081	1,0288	1,0318	-0,55	-0,08	0,29
810		1,0294			1,0285			-0,09	
812	0,9763	1,01	1,0069	0,9721	1,0082	1,0074	-0,42	-0,18	0,05
814	0,9467	0,9945	0,9893	0,9436	0,9919	0,9881	-0,31	-0,26	-0,12
850	1,0176	1,0255	1,0203	1,0143	1,0229	1,0189	-0,33	-0,26	-0,14
816	1,0172	1,0253	1,02	1,0139	1,0226	1,0187	-0,33	-0,27	-0,13
818	1,0163			1,013			-0,33		
820	0,9926			0,9898			-0,28		
822	0,9895			0,9839			-0,56		
824	1,0082	1,0158	1,0116	1,0052	1,013	1,0097	-0,3	-0,28	-0,19
826		1,0156			1,0126			-0,3	
828	1,0074	1,0151	1,0109	1,0045	1,0124	1,009	-0,29	-0,27	-0,19
830	0,9894	0,9982	0,9938	0,9872	0,9957	0,9913	-0,22	-0,25	-0,25
854	0,989	0,9978	0,9934	0,9868	0,9952	0,9908	-0,22	-0,26	-0,26
852	0,9581	0,968	0,9637	0,9571	0,9658	0,96	-0,1	-0,22	-0,37
832	1,0359	1,0345	1,036	1,0349	1,0322	1,032	-0,1	-0,23	-0,4
858	1,0336	1,0322	1,0338	1,0324	1,0298	1,0297	-0,12	-0,24	-0,41
834	1,0309	1,0295	1,0313	1,0297	1,027	1,0272	-0,12	-0,25	-0,41
842	1,0309	1,0294	1,0313	1,0296	1,027	1,0271	-0,13	-0,24	-0,42
844	1,0307	1,0291	1,0311	1,0294	1,0266	1,0269	-0,13	-0,25	-0,42
846	1,0309	1,0291	1,0313	1,0297	1,0266	1,0272	-0,12	-0,25	-0,41
848	1,031	1,0291	1,0314	1,0297	1,0266	1,0272	-0,13	-0,25	-0,42
860	1,0305	1,0291	1,031	1,0292	1,0266	1,0268	-0,13	-0,25	-0,42
836	1,0303	1,0287	1,0308	1,029	1,0262	1,0266	-0,13	-0,25	-0,42
840	1,0303	1,0287	1,0308	1,029	1,0262	1,0266	-0,13	-0,25	-0,42
862	1,0303	1,0287	1,0308	1,029	1,0262	1,0266	-0,13	-0,25	-0,42
838		1,0285			1,0258			-0,27	
864	1,0336			1,0324			-0,12		
888	0,9996	0,9983	1	1,0015	0,9994	0,9978	0,19	0,11	-0,22
890	0,9167	0,9235	0,9177	0,9247	0,9301	0,9218	0,8	0,66	0,41
856		0,9977			0,995			-0,27	

A su vez, se compararon los valores de los ángulos de las fases obtenidos para cada estación con los valores brindados por la IEEE, obteniendo los siguientes resultados:

Tabla 4-16. Comparativa de los valores de las fases de tensión en [°] obtenidos mediante las simulaciones y los valores brindados por la IEEE.

Nodo	Ángulos de tensiones de fase [°]						Diferencia en grados [°]		
	IEEE			APTDdraw			Fase A	Fase B	Fase C
	Fase A	Fase B	Fase C	Fase A	Fase B	Fase C	Fase A	Fase B	Fase C
800	0	-120	120	0	-119,75	120,49	0	0,25	0,49
802	-0,05	-120,07	119,95	-0,05	-119,82	120,44	0	0,25	0,49
806	-0,08	-120,11	119,92	-0,09	-119,86	120,41	-0,01	0,25	0,49
808	-0,75	-120,95	119,3	-0,77	-120,68	119,8	-0,02	0,27	0,5
810		-120,95			-120,68			0,27	
812	-1,57	-121,92	118,59	-1,63	-121,63	119,1	-0,06	0,29	0,51
814	-2,26	-122,7	118,01	-2,34	-122,39	118,52	-0,08	0,31	0,51
850	-2,26	-122,7	118,01	-2,34	-122,39	118,52	-0,08	0,31	0,51
816	-2,26	-122,71	118,01	-2,35	-122,4	118,52	-0,09	0,31	0,51
818	-2,27			-2,35			-0,08		
820	-2,32			-2,41			-0,09		
822	-2,33			-2,41			-0,08		
824	-2,37	-122,94	117,76	-2,47	-122,61	118,27	-0,1	0,33	0,51
826		-122,94			-122,61			0,33	
828	-2,38	-122,95	117,75	-2,48	-122,62	118,25	-0,1	0,33	0,5
830	-2,63	-123,39	117,25	-2,77	-123,01	117,75	-0,14	0,38	0,5
854	-2,64	-123,4	117,24	-2,77	-123,02	117,74	-0,13	0,38	0,5
852	-3,11	-124,18	116,33	-3,29	-123,71	116,82	-0,18	0,47	0,49
832	-3,11	-124,18	116,33	-3,29	-123,71	116,82	-0,18	0,47	0,49
858	-3,17	-124,28	116,22	-3,35	-123,8	116,71	-0,18	0,48	0,49
834	-3,24	-124,39	116,09	-3,42	-123,91	116,58	-0,18	0,48	0,49
842	-3,25	-124,39	116,09	-3,42	-123,92	116,57	-0,17	0,47	0,48
844	-3,27	-124,42	116,06	-3,45	-123,94	116,55	-0,18	0,48	0,49
846	-3,32	-124,46	116,01	-3,49	-123,99	116,5	-0,17	0,47	0,49
848	-3,32	-124,47	116	-3,5	-124	116,49	-0,18	0,47	0,49
860	-3,24	-124,39	116,09	-3,41	-123,91	116,58	-0,17	0,48	0,49
836	-3,23	-124,39	116,09	-3,41	-123,91	116,58	-0,18	0,48	0,49
840	-3,23	-124,39	116,09	-3,41	-123,91	116,58	-0,18	0,48	0,49
862	-3,23	-124,39	116,09	-3,41	-123,92	116,58	-0,18	0,47	0,49
838		-124,39			-123,92			0,47	
864	-3,17			-3,35			-0,18		

888	-4,64	-125,73	114,82	-4,75	-125,1	115,44	-0,11	0,63	0,62
890	-5,19	-126,78	113,98	-5,31	-126,04	114,68	-0,12	0,74	0,7
856		-123,03			-123,03			0	

4.2.2 Resultados de las corrientes

Para analizar los flujos de corrientes obtenidos de las simulaciones se confeccionó la siguiente tabla:

Tabla 4-17. Comparativa de los valores de corrientes RMS en [A] obtenidos en las simulaciones y los valores brindados por la IEEE. Esta comparativa se realizó en 10 puntos de la red.

		Corrientes de fases [A]						Error porcentual %		
		IEEE			APTDDraw					
Estación	Nodo	Fase A	Fase B	Fase C	Fase A	Fase B	Fase C	Fase A	Fase B	Fase C
1	800	51,56	44,57	40,92	50,8	45,9	42,7	-1,5	3	4,3
2	850	48,47	40,04	38,17	47,7	41,1	39,8	-1,6	2,6	4,3
3	818	13,03			12,7			-2,5		
4	820	10,62			10,3			-3		
5	824	35,87	36,93	38,05	35,5	36	38,5	-1	-2,5	1,2
6	852	34,35	35,9	36,52	33,9	35	37	-1,3	-2,5	1,3
7	888	69,9	70,04	69,5	66,8	66,8	66,2	-4,4	-4,6	-4,7
8	834	20,29	22,37	23,23	20,7	22,8	23,5	2	1,9	1,2
9	844	14,47	16,29	15,11	14,5	16,4	15,1	0,2	0,7	-0,1
10	860	5,87	7,68	5,29	6,05	7,95	5,5	3,1	3,5	4

Finalmente, para analizar los ángulos de las fases de las corrientes se confeccionó la siguiente tabla:

Tabla 4-18. Comparativa de los valores de las fases de las corrientes en [°] obtenidos de las simulaciones y los valores brindados por la IEEE. Esta comparativa se realizó en 10 puntos de la red.

		Ángulos de corrientes de fase [°]						Diferencia en grados [°]		
		IEEE			APTDDraw					
Estación	Nodo	Fase A	Fase B	Fase C	Fase A	Fase B	Fase C	Fase A	Fase B	Fase C
1	800	-12,74	-127,7	117,37	-11,87	-127,99	116,4	0,87	-0,29	-0,97
2	850	-14,73	-128,69	116,23	-13,89	-128,89	115,29	0,84	-0,2	-0,94
3	818	-26,77			-26,81			-0,04		
4	820	-28,98			-29,06			-0,08		
5	824	-10,7	-127,39	116,25	-9,59	-128,09	115,82	1,11	-0,7	-0,43
6	852	-11	-128,66	115,41	-9,85	-129,42	114,97	1,15	-0,76	-0,44

7	888	-32,29	-152,73	87,39	-31,84	-152,39	87,92	0,45	0,34	0,53
8	834	2,18	-116,07	130,06	1,32	-116,82	129,7	-0,86	-0,75	-0,36
9	844	37,12	-95,71	150,97	36,52	-96,2	150,81	-0,6	-0,49	-0,16
10	860	-33,62	-156,52	86,1	-33,65	-156,29	86,65	-0,03	0,23	0,55

Analizando la información recopilada previamente, se observa que para el caso de las tensiones RMS en régimen estacionario, en la totalidad de los nodos el error se encuentra por debajo del 1%. Se observa que la Fase A es la que presenta mayor error, principalmente al inicio de la red. Analizando los flujos de corrientes, la Fase A presenta flujo de corriente y puede justificar el motivo por el cual esta fase presenta esa disminución en la tensión producto de las pérdidas en la línea y en el transformador de la subestación. Al alejarnos de la subestación, más precisamente al superar el nodo 816 que es el encargado de alimentar una línea monofásica de fase A que contiene a los nodos 818, 820 y 822 y sus respectivas cargas, el error en la tensión de esta fase disminuye.

Si se analiza la fase C, sucede lo opuesto que para la Fase A, es decir, presenta un valor de error similar, pero en este caso la tensión obtenida en la simulación es mayor que la informada por la IEEE. Nuevamente, al analizar los flujos de corrientes se observa que la fase C es la que presenta menor flujo de corriente y que, en consecuencia, posee menores pérdidas en la línea y el transformador de la subestación. Para esta fase, la tensión medida al inicio de la red resulta estar un 0,5% por encima de la referencia.

En la mitad de la red el error en la tensión de esta fase pasa a ser negativo, es decir, los valores de tensiones RMS obtenidos en la simulación se encuentran, en el peor de los casos, un 0,42% por debajo de la referencia. Al analizar los flujos de corrientes se observa que las fases A y B se descargan progresivamente, principalmente si se analiza entre las estaciones de medición 1 y 5, pero la fase C lo hace en menor medida y con valores por encima del valor referencia. Este comportamiento genera mayores caídas de tensión en la línea y justificaría estos valores en las tensiones medidas.

Analizando las corrientes medidas en la simulación se observa que el mayor error obtenido se encuentra en el nodo 888, el cual se encuentra en el lado de baja tensión del segundo transformador de esta red. Como se comentó en el *Capítulo 3 Sección 1.3 Algoritmo de convergencia*, el valor de la carga del nodo 890 fue modificado hasta obtener un margen de error menor a un 5% en los valores de corriente RMS. El análisis en este capítulo indica que, de ser posible, el error en la corriente debería disminuirse aún más para obtener una mejora en los perfiles de tensión.

En lo que respecta a las fases de las tensiones y las corrientes, la máxima diferencia expresada en grados fue de 0,74 ° para el caso de la tensión y de 1,15 ° para el caso de la corriente. La región del circuito donde se obtuvieron estas diferencias fue la misma, siendo

en el nodo 852 para el caso de la tensión, lugar previo a la bifurcación hacia la parte final del circuito y el transformador XFM-1 que conecta al nodo 888 y que coincide con el nodo de mayor error de fase para el caso de la corriente. Si se analizan estas regiones, en lo que respecta a la parte final del circuito, se observan la presencia de los bancos de capacitores (nodos 848 y 844) que pueden ser los causantes de esta diferencia en las fases medidas. A su vez, si se analiza la región del transformador XFM-1, coincide con el lugar donde se obtuvo el mayor error en la corriente RMS, que al tratarse de baja tensión, es de un valor considerable pudiendo ser la causante de este desfase. Esto denota la sensibilidad de la red frente al modelado de elementos y, en consecuencia, la importancia de modelar correctamente dichos elementos de la red como los bancos de capacitores y los transformadores.

4.3 Conclusión

Luego de realizar una comparativa entre el alimentador de pruebas de 34 nodos simulado en *ATPDraw* y el mismo modelo disponible y proporcionado por la IEEE, se concluye que esta red eléctrica fue validada satisfactoriamente. Para ello, se analizaron los valores RMS de tensiones y corrientes junto a sus ángulos de fases en puntos estratégicos de la red para el caso de las corrientes y en la totalidad de los nodos para el caso de la tensión, comparándolos con los valores de referencia.

Los mayores errores se obtuvieron en los flujos de corriente, alcanzando valores de hasta 4,7%. Esto puede explicarse principalmente por la presencia de cargas que la IEEE define del tipo corriente constante y que, durante la aplicación del algoritmo de convergencia, solo se tuvo en cuenta una única carga con este tipo de comportamiento. Esto significa que las cargas de la red que estén conectadas a tensiones distintas de la nominal tendrán un comportamiento en sus impedancias tal que sus consumos de corriente se mantendrán constantes. Vale la pena aclarar que este comportamiento es idealizado y no debe perderse de vista que el punto de partida para modelar estas redes eléctricas, en el mejor de los casos, son redes eléctricas reales que han sido simplificadas. Aun así, se evidenció que en esta red simulada los perfiles de tensión son variados, por lo tanto, las cargas con este tipo de característica pasan a tener un papel importante en los resultados de la simulación y vale la pena analizar cómo impacta su comportamiento en las simulaciones. Analizando la potencia instalada en la red, de los 2054,1 [kVA] de potencia total, 650 [kVA] son cargas del tipo corriente constante, es decir, representan el 31,6% de la carga total.

En lo que respecta a los perfiles de tensión, se concluye que los resultados obtenidos son aceptables teniendo errores menores a 0,68% comparados con los valores de referencia.

Para los valores de los ángulos de las fases, nuevamente la corriente es la que presenta mayor error llegando a tener desviaciones de 1,15°. Analizando las zonas de la red donde la

desviación de los ángulos de las fases aumenta, se encuentra que los bancos de capacitores pueden ser uno de los causantes de estas desviaciones, como así también los transformadores. Siendo dos elementos que, para este trabajo, fueron modelados de una manera simplificada, este resultado indica que el modelado de ambos elementos constituye un aspecto a mejorar a futuro.

Otro aspecto que se debe considerar a la hora de analizar los resultados, son los errores numéricos introducidos al discretizar las ecuaciones diferenciales que, si bien pueden desprejarse al no tener un efecto significativo sobre las conclusiones obtenidas, siempre están presentes en las simulaciones computacionales.

En conclusión, como aspectos a mejorar del presente trabajo, se propone ajustar los valores de las cargas eléctricas de la red a través del desarrollo del *algoritmo de convergencia*, principalmente aquellas con comportamiento de corriente constante por tener un papel importante en las pérdidas en la línea y que repercute en el perfil de tensión. También podría profundizarse más en el modelado de los transformadores, por ejemplo, cargando datos de ensayos de transformadores reales al momento de definir los parámetros de simulación. Otros elementos modelados que tuvieron un desempeño aceptable pero que podrían mejorarse son los reguladores de tensión y los bancos de capacitores.

A pesar de los aspectos a mejorar expuestos previamente, se considera que el desempeño de la red eléctrica modelada es satisfactorio para alcanzar los objetivos planteados en el presente trabajo.

En el próximo capítulo se expondrán, a partir de esta red validada, los resultados obtenidos de las fallas eléctricas y los algoritmos de clasificación de eventos de tensión.

4.4 Bibliografía

- [54] K. P. Schneider, B. A. Mather, B. C. Pal, C. W. Ten, G. J. Shirek, H. Zhu, J. C. Fuller, J. L. R. Pereira, L. F. Ochoa, L. R. de Araujo, R. C. Dugan, S. Matthias, S. Paudyal, T. E. McDermott, and W Kersting (2017). *Analytic Considerations and Design Basis for the IEEE Distribution Test Feeders*. IEEE Transactions on Power Systems, vol. PP, no. 99, pp. 1-1, 2017. Recuperado de: <https://cmte.ieee.org/pes-testfeeders/resources/>
- [55] Kersting W. H. (1991). *Radial distribution test feeders*. IEEE Transactions on Power Systems, vol. 6, no. 3, pp. 975-985, 1991. <https://doi.org/10.1109/59.119237>
- [56] Kersting W.H, Fellow, IEEE, and Phillips W. H (1992) *Modeling and Analysis of Rural Electric Distribution Feeders*. IEEE Transactions on Industry applications, Vol. 28, N°4, pp. 767-773. <https://doi.org/10.1109/28.148441>
- [57] William H Kersting (2018). *Distribution System Modeling and Analysis*, Fourth Edition-CRC Press.
- [58] Institute of Electrical and Electronics Engineers. (2007) IEEE Std 142-2007: IEEE Recommended Practice for Grounding of Industrial and Commercial Power Systems.
- [59] T.A. Short (2004). *ELECTRIC POWER distribution handbook*. CRC Press LLC
- [60] James O. Owuor, Josiah L. Munda, Adisa A. Jimoh (2011). *The IEEE 34 node radial test feeder as a simulation testbench for distributed generation*. IEEE Africon 2011-The Falls Resort and Conference Centre, Livingstone, Zambia, 13 - 15. <https://doi.org/10.1109/AFRCON.2011.6072095>

CAPÍTULO 5

MODELADO DE RED

DESEMPEÑO DE ALGORITMOS DE
DETECCIÓN Y CLASIFICACIÓN DE
EVENTOS DE TENSIÓN



5 Desempeño de algoritmos de detección y clasificación de eventos de tensión

Resumen

En el presente capítulo se expondrán los resultados obtenidos a partir de la simulación de las fallas monofásicas, bifásicas sin contacto a tierra y bifásicas con contacto a tierra en distintos puntos de la red variando el valor de impedancia de falla de 0 p.u a 1.35 p.u.

A partir del análisis del desempeño de los algoritmos ASA [62] y eASA [63] en la clasificación de eventos de tensión, se llegó a la conclusión de que el algoritmo eASA presentó mejores resultados, obteniendo errores de clasificación inferiores en comparación al algoritmo ASA. Esto evidencia, a su vez, la influencia de las modificaciones introducidas en el algoritmo eASA, generando diferencias notables en el desempeño de ambos algoritmos a pesar de que mantienen una arquitectura global similar.

5.1 Análisis de resultados

Como fue mencionado en capítulos anteriores, el objetivo del presente trabajo es analizar el desempeño de algoritmos de detección de eventos de tensión aplicados en una red eléctrica validada. Para el caso particular, fueron estudiados los algoritmos ASA [62] y eASA [63] a través de su aplicación sobre una red eléctrica de 34 nodos.

En esta sección, se expondrán los resultados obtenidos, a partir de las distintas fallas detectadas, por cada modelo de clasificación de evento de tensión y tipo de falla (monofásica, bifásica o bifásica con contacto a tierra) para cada estación de medición.

Se debe reiterar que la clasificación de las fallas fue realizada a partir de las mediciones relevadas en ocho puntos de la red que corresponden a las ocho estaciones de medición trifásicas definidas en el Capítulo 3. Las estaciones de medición 3 y 4 no han sido consideradas en este análisis debido a que los algoritmos de detección de fallas empleados son aplicables únicamente a eventos de tensión en líneas eléctricas trifásicas.

A continuación, se adjunta el esquema de la red de 34 nodos simulada indicando la ubicación de las estaciones de medición para una mejor comprensión de los resultados expuestos.

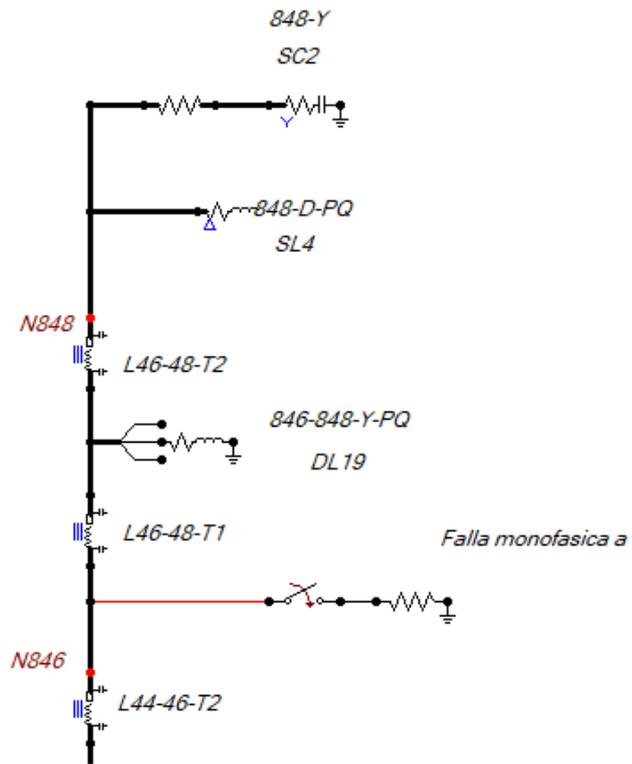


Figura 5-2. Imagen de un fragmento del circuito eléctrico simulado con una falla monofásica.

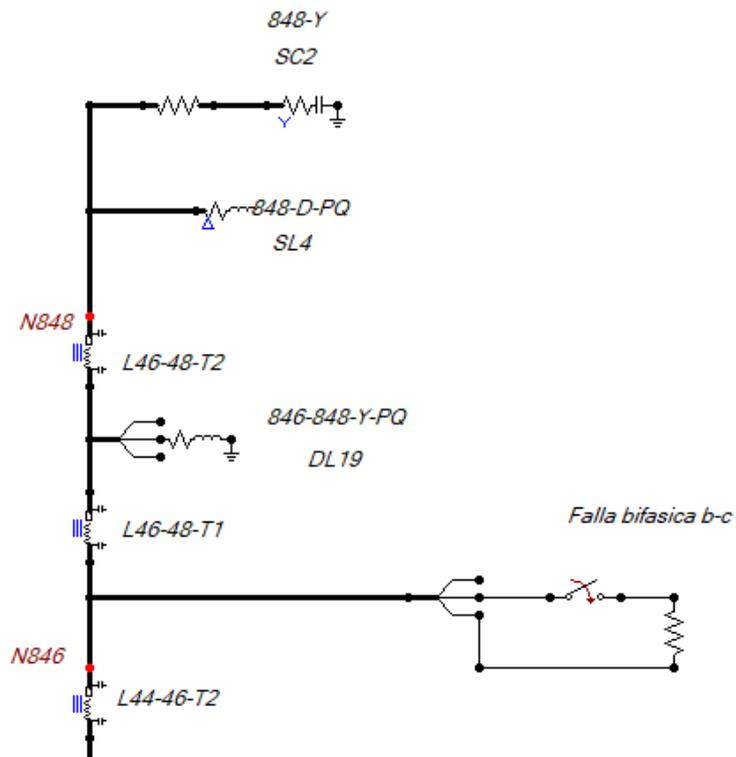


Figura 5-3. Imagen de un fragmento del circuito eléctrico simulado con una falla bifásica sin contacto a tierra.

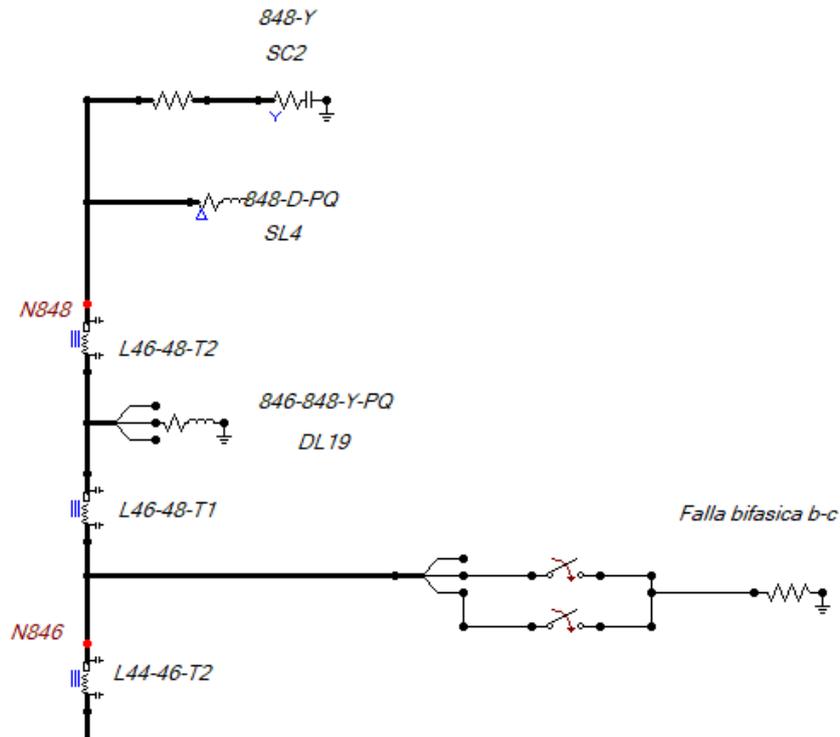


Figura 5-4. Imagen de un fragmento del circuito eléctrico simulado con una falla bifásica con contacto a tierra.

5.1.1 Estación de medición 1

Para comenzar el análisis de los resultados se iniciará mostrando las diferencias de desempeño de los algoritmos en la estación de medición 1 ubicada en el nodo 800, es decir, al inicio de la red. Esta estación fue la que presentó la mayor cantidad de casos no detectados, clasificados como “eventos normales”. Esto se debe a que las fallas al final del circuito, cuando la impedancia de falla se acrecentaba, generaban huecos de tensión que no eran lo suficientemente profundos para ser detectados en este punto de la red.

A continuación, se presentan los resultados para luego de simular fallas bifásicas con y sin contacto a tierra y fallas monofásicas.

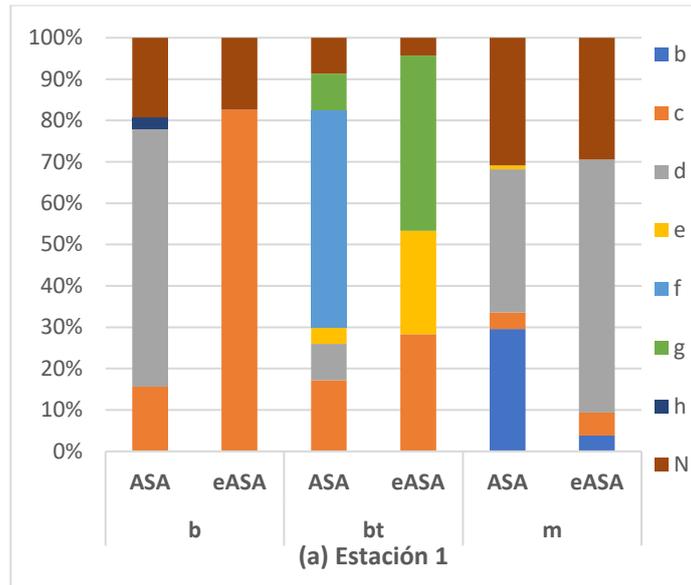


Figura 5-5. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 1.

5.1.1.1 Falla bifásica sin contacto a tierra

En los resultados del algoritmo ASA el 63% de las fallas fueron clasificadas como Tipo D, el 19,3% de las fallas como Tipo N (evento no detectado), el 16% de las fallas como Tipo C y el 1,7% restante como Tipo H. Es decir, el 63% de los eventos fueron clasificados erróneamente como un evento generado por una falla monofásica.

Analizando los resultados del algoritmo eASA el 82,6% de las fallas fueron clasificadas como Tipo C y un 17,4% como N (evento no detectado).

En primera instancia, se observa que los eventos de tensión no detectados son levemente distintos. Esto se repite para otras fallas y otras estaciones de medición. La razón de esta diferencia es que el algoritmo ASA fue configurado para que determine si hubo un evento cuando las tensiones de fase sean inferiores a 0.9 [p.u]. Por otro lado, el algoritmo eASA fue configurado para que determine si hubo un evento de tensión cuando la tensión mínima “ $V_{\text{mínima}}$ ” sea inferior a 0.9 [p.u]. Recordando que la $V_{\text{mínima}}$ evalúa a las tensiones de fase y a las tensiones de línea sobre raíz de 3. Esto quiere decir que hay un pequeño porcentaje en donde las tensiones de línea sobre raíz de 3 son menores a las tensiones de fase, siendo el resultado del gran desequilibrio de las tensiones luego de la falla. Si bien para el caso de eventos no detectados esto no influye en gran medida, como se verá posteriormente, los resultados de la $V_{\text{mínima}}$ provocan errores de clasificación en el algoritmo ASA.

Comenzando el análisis con el algoritmo ASA y considerando los casos donde clasifica de manera errónea a las fallas como eventos de Tipo D, se deduce que la razón principal de la clasificación errónea se genera debido al valor de $V_{\text{mínima}}$ de la falla. Como primera toma de decisiones del algoritmo ASA se evalúa el valor de $V_{\text{mínima}}$ y si coincide con el valor de una de

las fases determina que el evento más probable es aquel en el que se ve afectada una fase (evento monofásico). Si, por el contrario, coincide con algunas de las tensiones de línea sobre raíz de tres determina que el evento más probable es aquel en el que hay dos fases afectadas (evento bifásico). De esta manera, se genera un primer filtro de eventos posibles y se calcula el error cuadrático medio de los eventos restantes. Esto genera errores debido a que en muchos casos el evento que presenta menor error cuadrático medio mínimo fue descartado previamente, con lo cual termina clasificando a un evento incorrecto. Este tipo de errores se verá repetido en el resto de las clasificaciones que se presentarán a lo largo de este capítulo.

En cambio, el algoritmo eASA no realiza este primer filtro y calcula el error cuadrático medio para todos los eventos, seleccionando aquel que presente el mínimo error cuadrático medio. Se adjunta un ejemplo de clasificación errónea del algoritmo ASA y una clasificación correcta del algoritmo eASA. Para ello se muestran los fasores de las tensiones de línea, la información de la falla y una tabla con los datos que cada algoritmo utiliza para seleccionar el evento más probable.

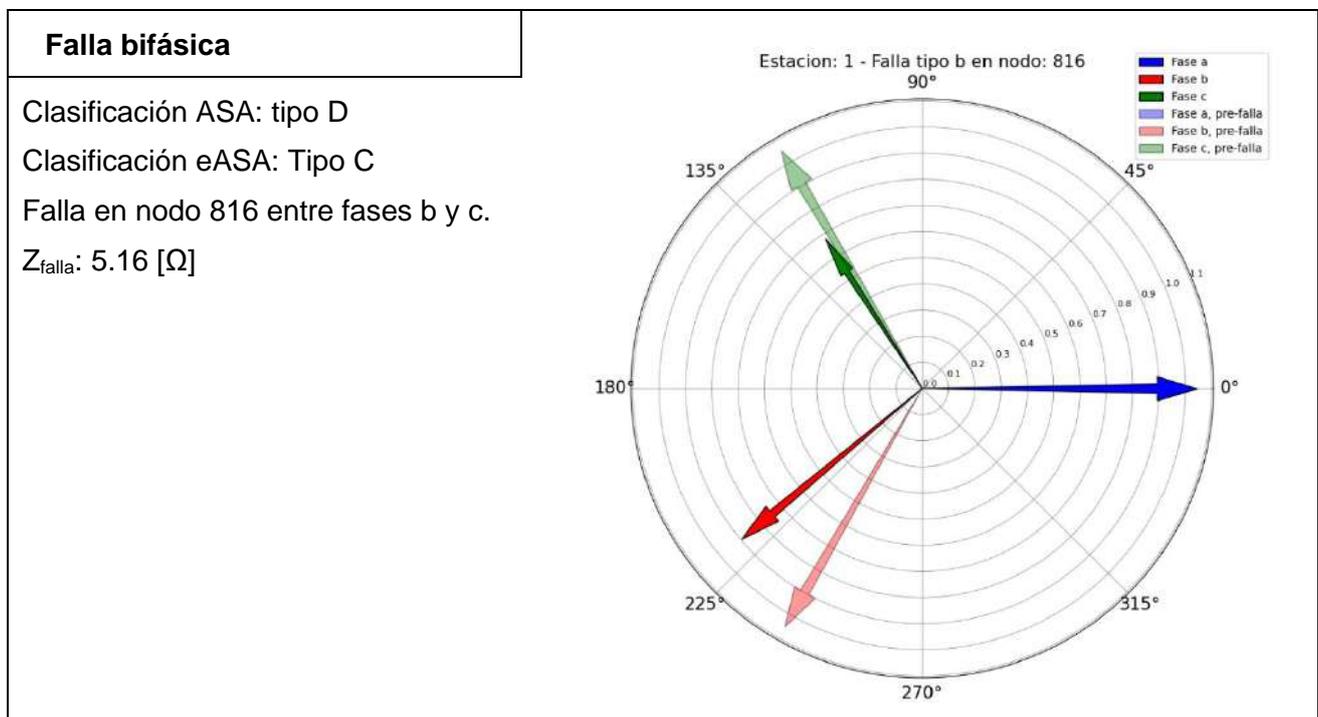


Figura 5-6. Falla bifásica en nodo 816, registrada en estación de medición 1.

Tabla 5-1. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica en nodo 816, registrada en estación de medición 1.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V _{mín}	V ₀₁	V ₊₁	V ₋₁	V _{01 mod}	V _{+1 mod}	V _{-1 mod}	Evento	Modelo
1,047	0,896	0,680	1,053	0,688	0,881	0,680	0,003	0,861	0,214	0,000	0,865	0,185	d	ASA
1,047	0,896	0,680	1,053	0,688	0,881	0,680	-0,003	0,861	0,214	0,000	0,865	0,185	c	eASA

En la Tabla 5-1 se observa que la $V_{\text{mínima}}$ coincide con la tensión de fase C y que es ligeramente menor a la tensión de línea afectada V_{bc} sobre raíz de 3, provocando que el algoritmo ASA clasifique incorrectamente. Al analizar el algoritmo eASA, la misma falla la clasifica como Tipo C, detectando correctamente el corrimiento de las fases B y C (tienden a acercarse) por la falta de la componente homopolar. Se seleccionó este caso porque presenta un corrimiento leve en la fase C pero una disminución notable en su magnitud, generando una firma que a simple vista podría considerarse que fue provocada por una falla monofásica en la fase C.

Como segundo ejemplo, se muestra la falla bifásica sin contacto a tierra en el nodo 832, siendo un nodo alejado de la estación de medición 1. En este caso, la falla genera un menor hueco de tensión principalmente en la fase C. Nuevamente el algoritmo ASA clasifica erróneamente esta falla como un tipo D debido al valor de $V_{\text{mínima}}$.

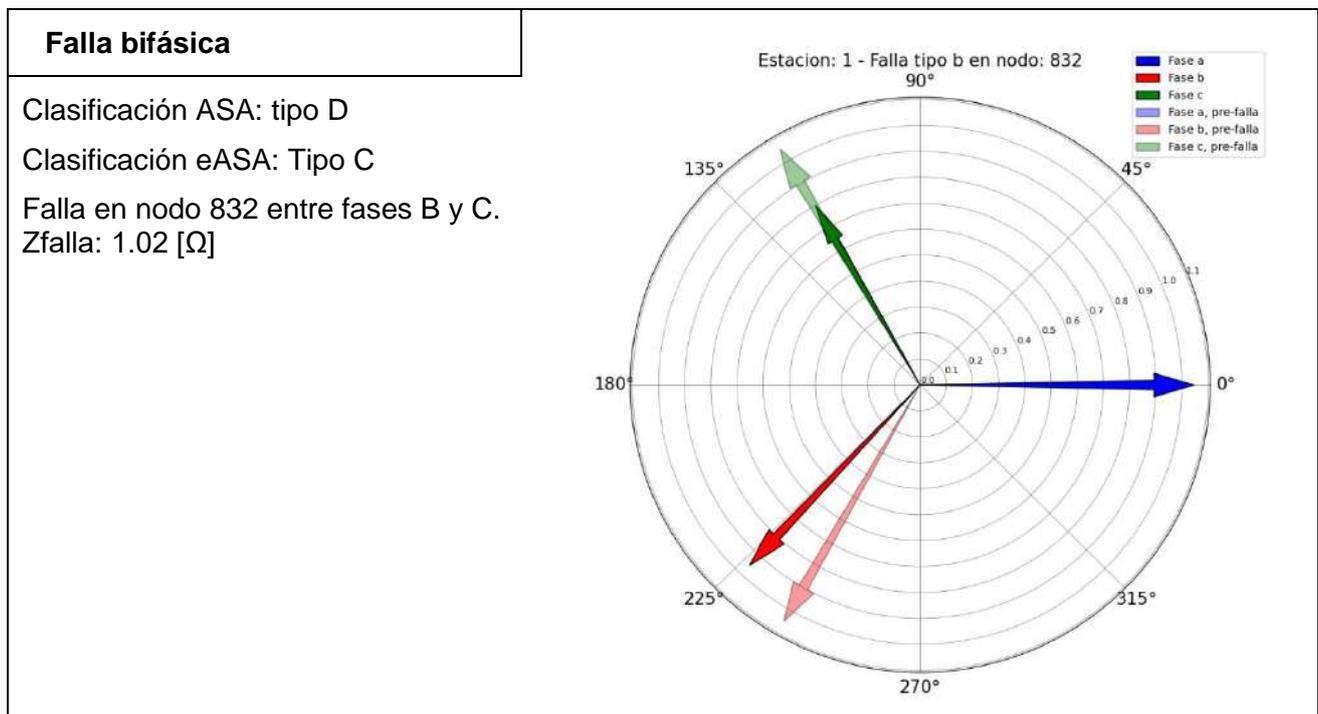


Figura 5-7. Falla bifásica en nodo 832, registrada en estación de medición 1.

Tabla 5-2. Valores de tensión RMS, $V_{\text{mín}}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica en nodo 832, registrada en estación de medición 1.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	$V_{\text{mín}}$	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
1,046	0,954	0,798	1,060	0,814	0,925	0,798	0,003	0,927	0,144	0,000	0,924	0,126	d	ASA
1,046	0,954	0,798	1,060	0,814	0,925	0,798	-0,003	0,927	0,144	0,000	0,924	0,126	c	eASA

En la red de 34 nodos simulada, la fase C empieza a cargarse en la mitad de la red hacia el final de la misma y al efectuarse una falla en el primer tramo de la red, donde está menos cargada, la fase C es la que sufre el mayor cambio de impedancia, reflejado en un mayor hueco en dicha fase. Al analizar los resultados de la estación de medición 2 para fallas bifásicas (Ver sección 5.1.2) se observa cómo este error en la clasificación disminuye notablemente.

Al analizar los casos donde el algoritmo ASA clasificó a la falla como tipo H, se encontró que la razón de esta clasificación radica en que hay una tensión de línea sobre raíz de tres cuyo valor es mayor a 1.1 [p.u], por lo cual el algoritmo decide que el evento más probable es un *swell* de tensión, y entre los casos H e I el menor error es un evento tipo H. Por el contrario, el Algoritmo eASA calcula el error cuadrático medio para la totalidad de los eventos y selecciona al evento Tipo C por presentar menor error.

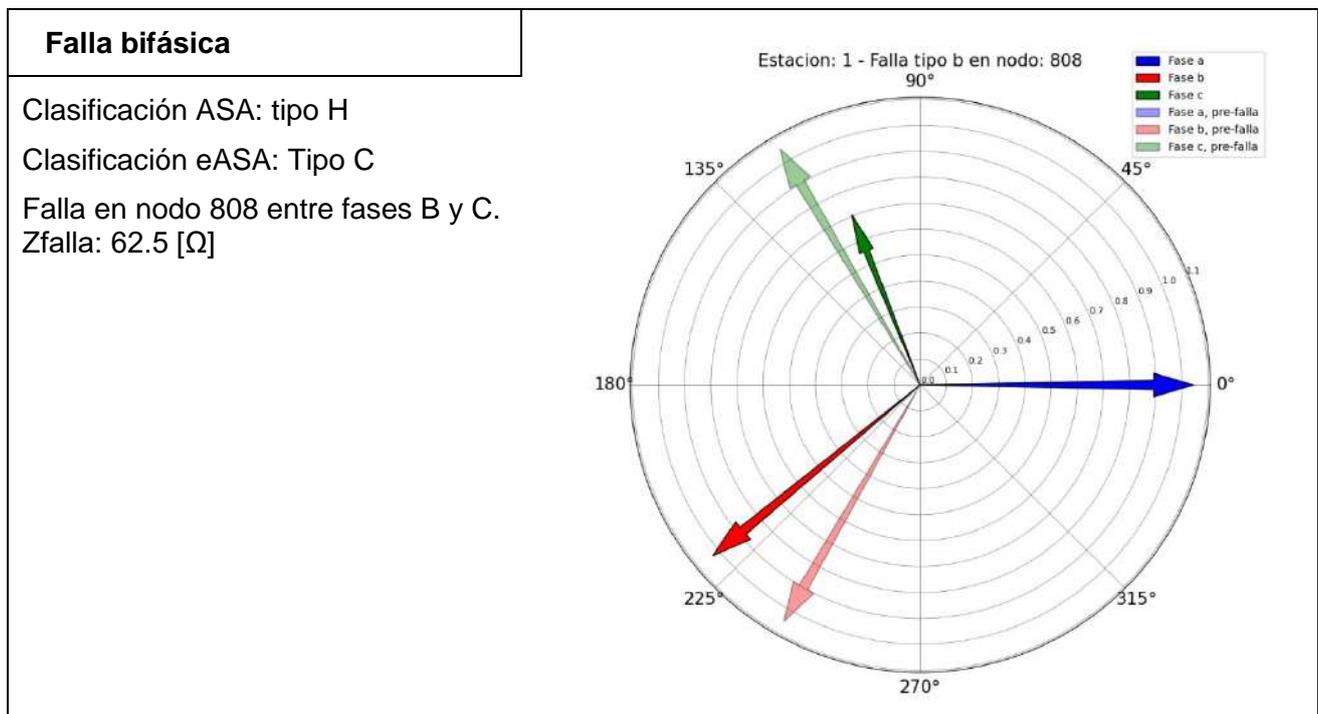


Figura 5-8. Falla bifásica en nodo 808, registrada en estación de medición 1.

Tabla 5-3. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica en nodo 808, registrada en estación de medición 1.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	$V_{mín}$	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
1,045	1,031	0,706	1,127	0,819	0,844	0,706	0,003	0,916	0,211	0,344	1,050	0,000	h	ASA
1,045	1,031	0,706	1,127	0,819	0,844	0,706	-0,003	0,916	0,211	0,000	0,878	0,172	c	eASA

Luego de analizar diferentes fallas bifásicas sin contacto a tierra en distintos nodos del circuito se llega a la conclusión que la razón principal de las clasificaciones erróneas del algoritmo ASA radica en descartar eventos de tensión en función de la tensión mínima o máxima. En cambio, el Algoritmo eASA no descarta posibles eventos dependiendo su tensión mínima o máxima, sino que selecciona como evento más probable aquel que tenga un menor error cuadrático medio. En este tipo de falla se destaca el desempeño del algoritmo eASA frente al ASA.

5.1.1.2 Falla bifásica con contacto a tierra

La falla bifásica con contacto a tierra fue la que generó mayores errores y variaciones en la clasificación de los algoritmos. Luego de analizar las fallas bifásicas a tierra generadas se notó que los desfases obtenidos en algunos casos estaban fuera del rango $[-45^\circ, 45^\circ]$, principalmente este comportamiento fue notado en las estaciones de mediciones ubicadas en el final del circuito y las cuales serán analizadas posteriormente.

Observando los resultados del algoritmo ASA, el 52,45% fueron clasificadas como tipo F, el 17,25% de las fallas fueron clasificadas como Tipo C, un 8,8% como tipo G, un 8,7% como Tipo D, un 4,1% como Tipo E y el 8,7 % restante como tipo N (evento no detectado). Es decir, el 61,15% de las fallas fueron clasificadas erróneamente un evento provocado por una falla monofásicas. La razón principal de esta clasificación errónea, al igual que como sucedió con las fallas bifásicas, es el descarte de eventos según su $V_{mínima}$ y $V_{máxima}$.

Analizando los resultados del algoritmo eASA el 28,3% de las fallas fueron clasificadas como Tipo C, un 25 % como tipo E, un 42,35% como tipo G y el 4,35% restante como tipo N (evento no detectado). A diferencia del algoritmo ASA, la totalidad de los eventos de tensión clasificados pertenecen a aquellos que son provocados por una falla bifásica.

Para realizar una comparativa entre los algoritmos se presentarán una serie de ejemplos de fallas bifásicas a tierra en distintos nodos.

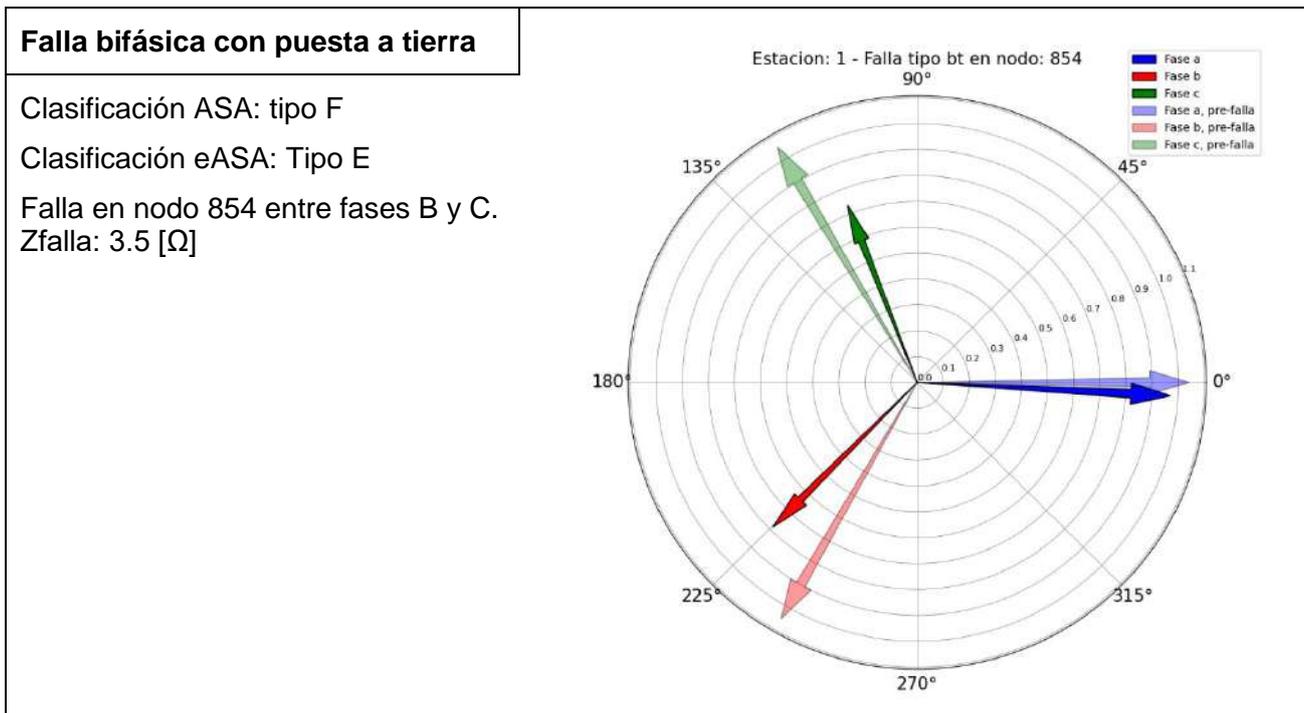


Figura 5-9. Falla bifásica con puesta a tierra en nodo 854, registrada en estación de medición 1.

Tabla 5-4. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica con puesta a tierra en nodo 854, registrada en estación de medición 1.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V _{mín}	V ₀₁	V ₊₁	V ₋₁	V ₀₁ mod	V ₊₁ mod	V ₋₁ mod	Evento	Modelo
0,972	0,787	0,737	0,927	0,738	0,832	0,737	0,056	0,829	0,110	0,000	0,841	0,104	f	ASA
0,972	0,787	0,737	0,927	0,738	0,832	0,737	0,056	0,829	0,110	0,104	0,841	0,104	e	eASA

Al igual que como se comentó previamente, el algoritmo ASA falla en la clasificación debido a que la $V_{mínima}$ del hueco de tensión es una tensión de fase (fase C) descartando el resto de los posibles eventos provocados por una falla bifásica. Es interesante notar como, a pesar que la falla tenga una rotación de fase simétricas (en sus tres fases) el algoritmo eASA detecta y logra clasificarla correctamente.

Se presenta una falla donde ambos algoritmos detectan un evento tipo G:

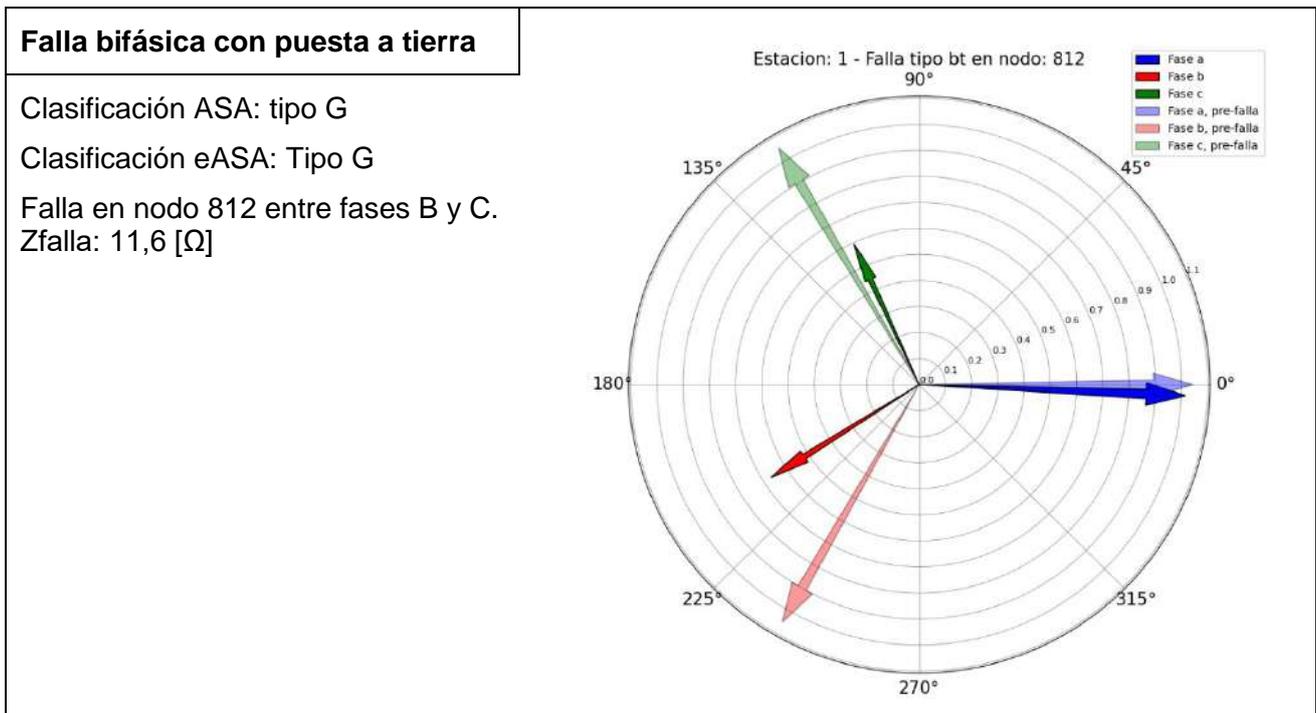


Figura 5-10. Falla bifásica con puesta a tierra en nodo 812, registrada en estación de medición 1.

Tabla 5-5. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica con puesta a tierra en nodo 812, registrada en estación de medición 1.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V _{mín}	V ₀₁	V ₊₁	V ₋₁	V ₀₁ mod	V ₊₁ mod	V ₋₁ mod	Evento	Modelo
1,017	0,668	0,595	0,931	0,549	0,804	0,549	0,082	0,746	0,220	0,000	0,716	0,167	g	ASA
1,017	0,668	0,595	0,931	0,549	0,804	0,549	0,082	0,746	0,220	0,000	0,716	0,167	g	eASA

Este caso es similar al anterior, existiendo una rotación de fase simétrica pero que se ve incrementada en la fase B, provocando que las fases B y C estén más cercanas. También se observa una disminución en la magnitud de la fase no afectada, en este caso en la fase A, provocando una firma similar a la del evento tipo G.

Por último, se muestra un caso donde el algoritmo eASA clasifica a esta falla como tipo C. El algoritmo ASA clasifica erróneamente la falla debido a que la tensión mínima aparece en la fase C.

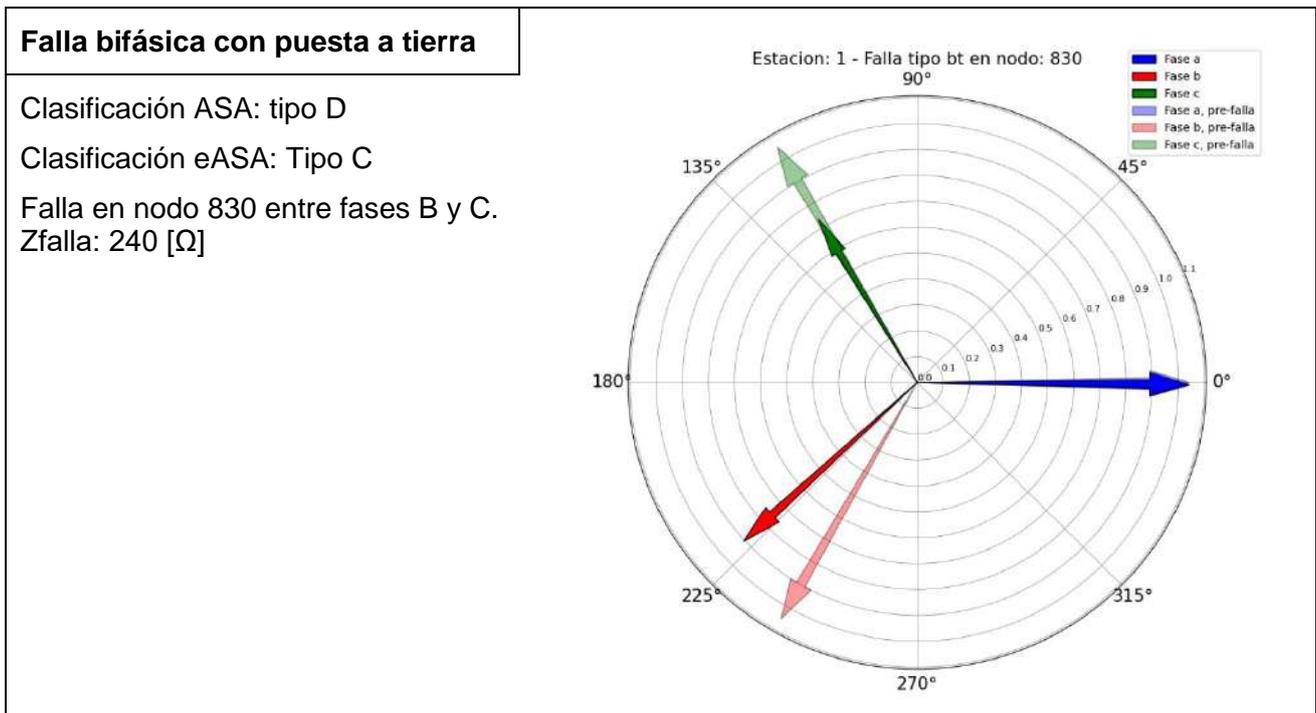


Figura 5-11. Falla bifásica con puesta a tierra en nodo 830, registrada en estación de medición 1.

Tabla 5-6. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica con puesta a tierra en nodo 830, registrada en estación de medición 1.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V _{mín}	V ₀₁	V ₊₁	V ₋₁	V ₀₁ mod	V ₊₁ mod	V ₋₁ mod	Evento	Modelo
1,044	0,906	0,736	1,047	0,738	0,901	0,736	0,003	0,886	0,180	0,000	0,893	0,157	d	ASA
1,044	0,906	0,736	1,047	0,738	0,901	0,736	-0,003	0,886	0,180	0,000	0,893	0,157	c	eASA

Se observa que la fase A se mantiene invariante, y las fases B y C disminuyen en magnitud. A su vez, la fase B tiene un salto de fase provocando que las fases B y C se acerquen entre sí. Estas características son típicas del evento tipo C.

Luego de analizar diferentes fallas bifásicas con contacto a tierra en distintos nodos del circuito se llega a la conclusión que la razón principal de las clasificaciones erróneas del algoritmo ASA sobre las mediciones en la estación 1, radica en descartar eventos de tensión en función de la tensión mínima o máxima. Al igual que para el caso de las fallas bifásicas sin contacto a tierra, el algoritmo eASA presenta un desempeño superior al seleccionar al evento más probable que presente menor error cuadrático medio. Se observó un buen desempeño clasificando eventos con saltos y rotaciones de fase.

5.1.1.3 Falla monofásica

Analizando los resultados del algoritmo ASA, el 31,15% de las fallas fueron clasificadas como Tipo B, un 34,6% como Tipo D, un 4% como Tipo C, un 0,9% como tipo E y el 29,35% restante como tipo N (evento no detectado). Es decir, el 4,9% de las fallas fueron clasificadas erróneamente como bifásicas.

Analizando los resultados del algoritmo eASA el 61,2% de las fallas fueron clasificadas como tipo D, el 3,8% de las fallas fueron clasificadas como Tipo B, un 5,5% como tipo C y el 29,5% restante como N (evento no detectado). En este caso, el 5,5% de las fallas fueron clasificadas erróneamente como bifásicas. En ambos algoritmos hubo clasificaciones tipo B y D, siendo eventos provocados por fallas monofásicas pero que se diferencian en que los eventos tipo D presentan un desfase en las fases no afectadas producto de la falta de componente homopolar.

Para analizar las causas de estas diferencias se presentarán una serie de casos que representan en buena medida el comportamiento general de los algoritmos.

Para este primer ejemplo se presenta una falla franca en el nodo 822:

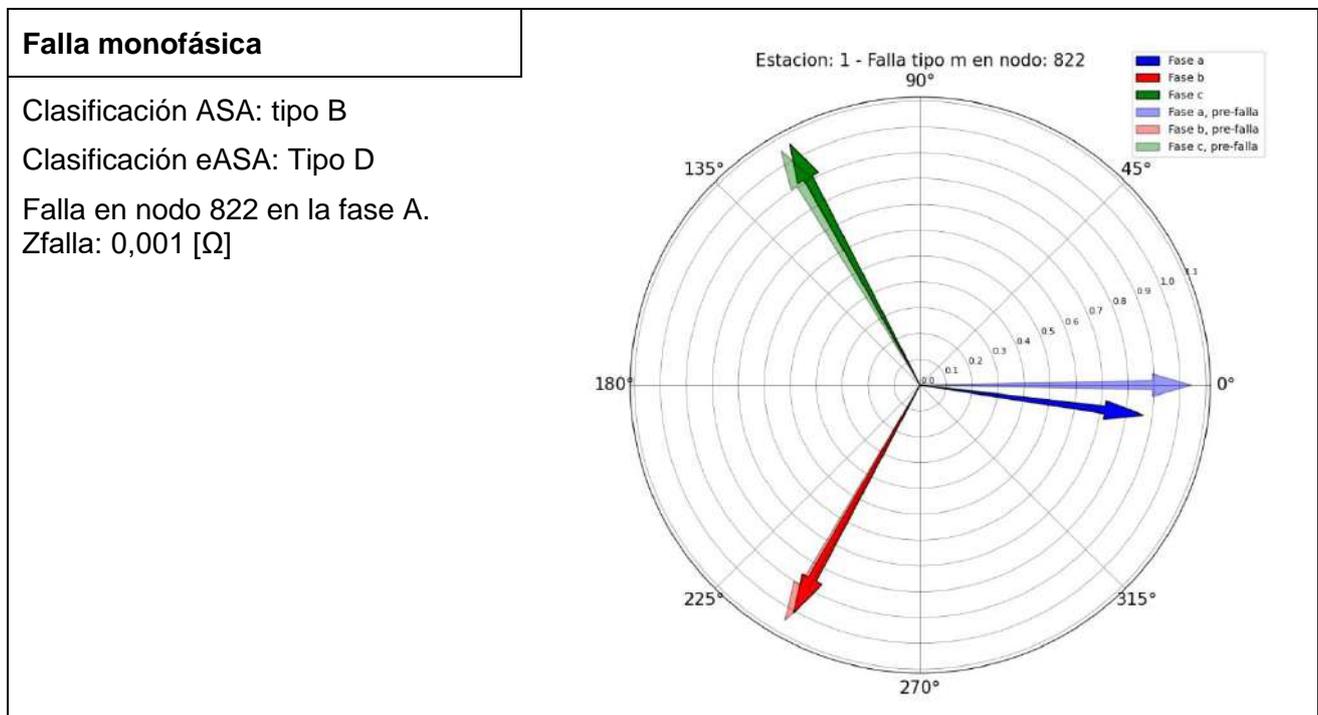


Figura 5-12. Falla monofásica en nodo 822, registrada en estación de medición 1.

Tabla 5-7. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 822, registrada en estación de medición 1.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	$V_{mín}$	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Event o	Model o
0,868	1,010	1,062	0,896	1,050	0,994	0,868	0,049	0,978	0,090	0,061	0,989	0,061	b	ASA
0,868	1,010	1,062	0,896	1,050	0,994	0,868	-0,049	0,978	-0,090	0,000	0,959	-0,091	d	eASA

Analizando la figura, se observa que las fases B y C presentan una apertura en sus ángulos de fase, comportamiento típico en las fallas tipo D, luego de que se elimine su componente homopolar. Si bien las tensiones tienen una componente homopolar, esta tiene un valor muy pequeño. Otra diferencia importante se observa en la Tabla 5-7, precisamente en la componente inversa que, para el caso del algoritmo eASA, presenta un valor negativo tanto en el valor del modelo teórico como en el valor medido calculado. Esta diferencia surge porque el algoritmo ASA analiza y compara el módulo de las componentes simétricas, mientras que el algoritmo eASA tiene en cuenta el signo, tanto de las componentes simétricas teóricas del modelo como en las componentes simétricas calculadas. Vale la pena recordar que este signo es parte de una convención que aplica el algoritmo luego de comparar los desfases entre la componente directa y la componente inversa, y entre la componente directa y la homopolar, de manera que contempla los corrimientos de las fases (Ver Capítulo 3.5.3). Esta es la razón principal por la cual el algoritmo eASA detecta un mayor porcentaje de estas fallas monofásicas como eventos del tipo D.

En el siguiente ejemplo se muestra como el algoritmo eASA clasifica erróneamente como un evento de tipo C, mientras que el algoritmo ASA clasifica como tipo B.

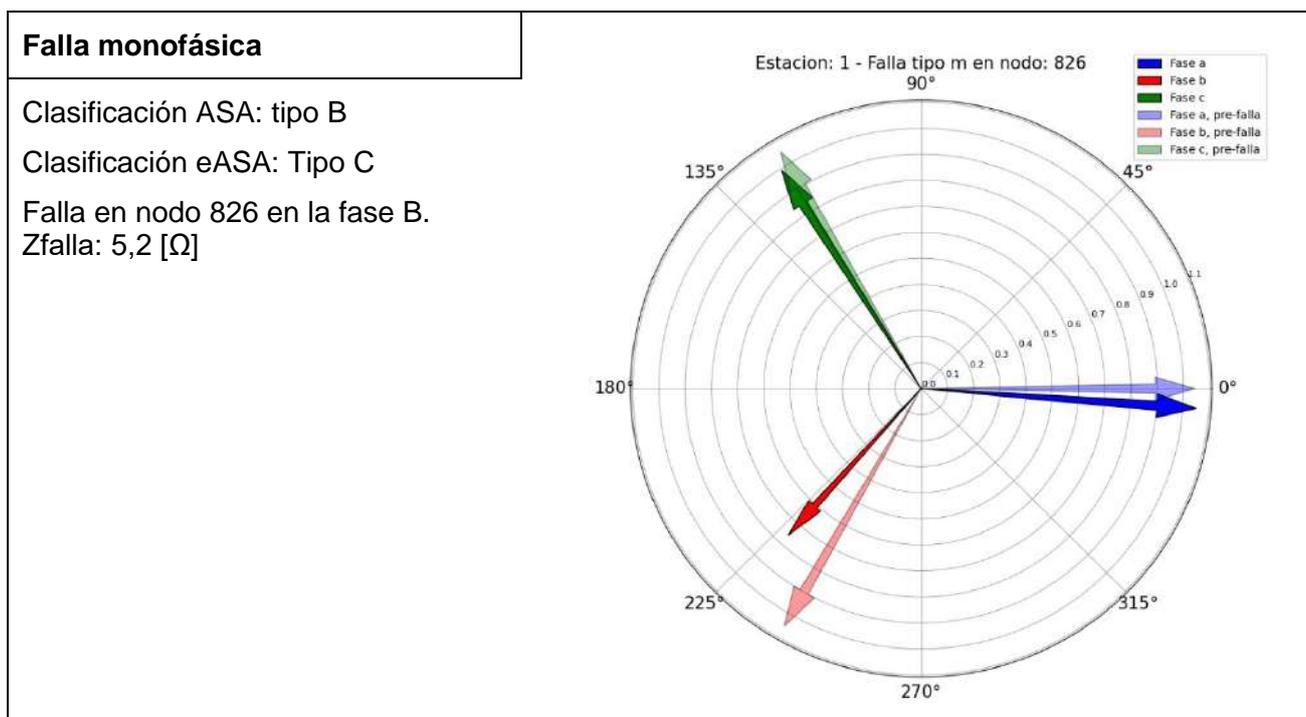


Figura 5-13. Falla monofásica en nodo 826, registrada en estación de medición 1.

Tabla 5-8. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 826, registrada en estación de medición 1.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
1,052	0,757	0,992	0,941	0,809	1,054	0,757	0,066	0,929	0,142	0,097	0,952	0,097	b	ASA
1,052	0,757	0,992	0,941	0,809	1,054	0,757	-0,066	0,929	0,142	0,000	0,903	0,146	c	eASA

Del diagrama fasorial se observa que la fase más afectada es la B, existiendo un corrimiento de las fases no afectadas, en este caso las fases A y C. Pero también se observa que la fase C disminuye en magnitud. Por este motivo, el algoritmo eASA detecta estos huecos de tensión en dos de las fases y clasifica erróneamente como tipo C.

Para el caso del algoritmo ASA, descarta posibles eventos generados por fallas bifásicas luego de evaluar el V_{\min} , que para este caso resulta ser una tensión de fase. Con lo cual, de los eventos restantes el que presenta menor error cuadrático medio es el Tipo B.

Como se verá en el resto de las estaciones, el algoritmo eASA presenta este error al clasificar fallas simuladas en las fases B de la red. Como se había definido en el Capítulo 3, se eligió la fase A para simular la totalidad de las fallas monofásicas. Sin embargo, esta red presenta tres líneas monofásicas en la fase B y es por eso que se simularon fallas en esta fase. Una posible conclusión es que, para este tipo de redes desequilibradas, puede considerarse una buena práctica simular en el mismo nodo fallas monofásicas en sus tres fases. De esta manera analizar si hay cambios significativos en las firmas que producen los huecos de tensión y evaluar de manera completa el desempeño del algoritmo.

5.1.2 Estación de medición 2

Análogamente al caso presentado anteriormente, se analizarán casos extraídos de los resultados obtenidos en la estación de medición 2. En la Figura 5-14 se presentan los resultados de los algoritmos.

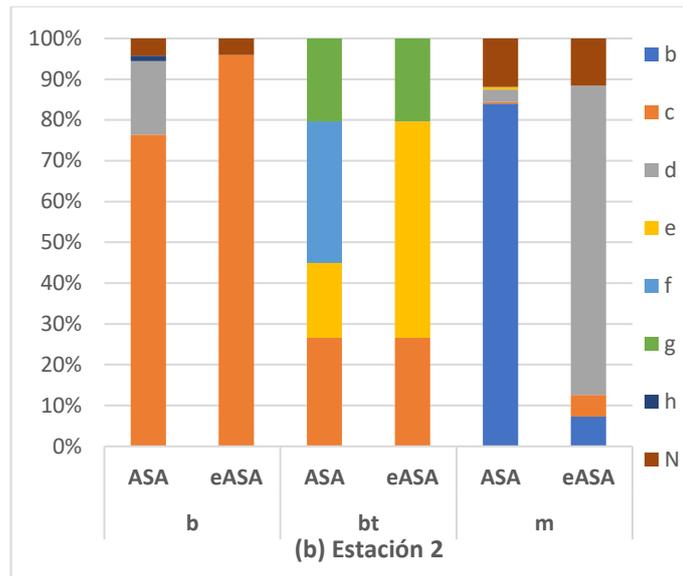


Figura 5-14. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 2.

5.1.2.1 Falla bifásica sin contacto a tierra

Analizando los resultados del algoritmo ASA, el 76,4% de las fallas fueron clasificadas como Tipo C, un 18 % como Tipo D, un 1,3% como Tipo H y el 4,3% restante como tipo N (evento no detectado). Si se compara estos resultados con los resultados de la estación 1 se observa una mejora significativa y se debe a que las fallas antes clasificadas erróneamente como tipo D disminuyen en gran medida. Para los casos donde ahora se clasifican como tipo C la razón es que ahora las $V_{\text{mínima}}$ medidas corresponden a tensiones de línea sobre raíz de tres y no a alguna de las tensiones de fase. En consecuencia, el algoritmo ASA no descarta los eventos generados por fallas bifásicas antes de evaluar el error cuadrático medio mínimo. Esta diferencia se debe a la posición de la estación de medición, teniendo mayor sensibilidad a las fallas del final del circuito.

Analizando los resultados del algoritmo eASA el 95,9% de las fallas fueron clasificadas como Tipo C y el 4,1% restante fue clasificado como Tipo N (evento no detectado).

Se presenta un ejemplo en donde ambos algoritmos clasificaron correctamente a la falla.

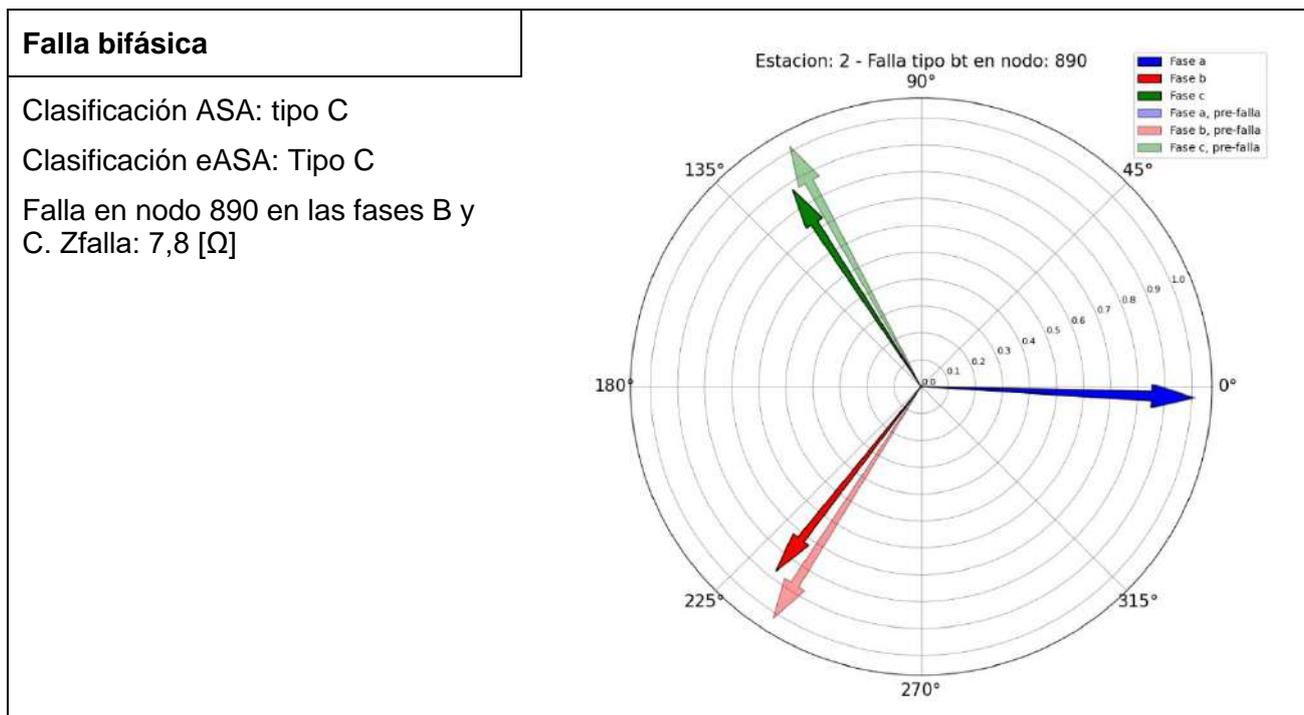


Figura 5-15. Falla bifásica en nodo 890, registrada en estación de medición 2.

Tabla 5-9. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica en nodo 890, registrada en estación de medición 2.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V _{mín}	V ₀₁	V ₊₁	V ₋₁	V ₀₁ mod	V ₊₁ mod	V ₋₁ mod	Evento	Modelo
1,002	0,873	0,876	0,963	0,822	0,964	0,822	0,005	0,914	0,092	0,000	0,921	0,098	c	ASA
1,002	0,873	0,876	0,963	0,822	0,964	0,822	-0,005	0,914	0,092	0,000	0,921	0,098	c	eASA

Analizando la figura anterior, se observa como la fase no afectada no sufrió cambios en su magnitud ni fase, y las fases afectadas B y C sufrieron un cambio en sus magnitudes y ángulo de fases, tendiendo a juntarse. Este es un comportamiento típico de los eventos tipo C donde no hay componente homopolar. En este caso, ambos algoritmos clasificaron con un error bajo a las fallas, aunque el algoritmo que se destaca es el eASA.

5.1.2.2 Falla bifásica con contacto a tierra

En los resultados del algoritmo ASA el 26,6% de las fallas fueron clasificadas como Tipo C, un 18,35 % como Tipo E, un 34,7% como Tipo F y un 20,3% como Tipo G. Es decir, el algoritmo clasificó erróneamente al 34,7% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. Nuevamente, el error de clasificación se debe principalmente al descartar eventos según la $V_{mínima}$.

Analizando los resultados del algoritmo eASA el 26,6% de las fallas fueron clasificadas como Tipo C, un 53% como Tipo E y el 20,4% restante como Tipo G, obteniendo resultados

levemente mejores a los casos analizados en la estación 1 para este tipo de falla. Se presenta un ejemplo para analizar los comportamientos de los algoritmos.

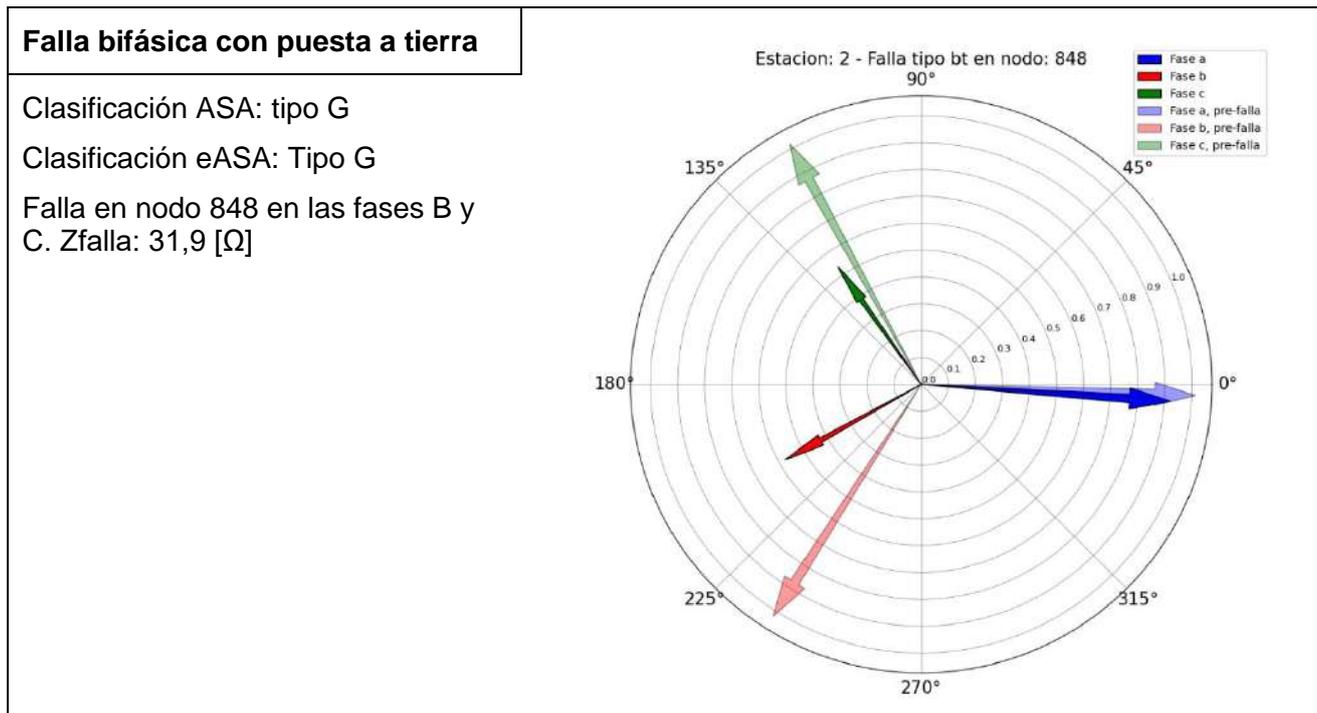


Figura 5-16. Falla bifásica con puesta a tierra en nodo 848, registrada en estación de medición 2.

Tabla 5-10. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla bifásica con puesta a tierra en nodo 848, registrada en estación de medición 2.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
0,924	0,576	0,536	0,832	0,430	0,767	0,430	0,049	0,658	0,235	0,000	0,626	0,196	g	ASA
0,924	0,576	0,536	0,832	0,430	0,767	0,430	0,049	0,658	0,235	0,000	0,626	0,196	g	eASA

Este caso es similar al analizado anteriormente para la falla bifásica sin contacto a tierra, pero con una disminución en la magnitud de la fase no afectada, es decir, la fase A. Esta firma generada por los huecos de tensión es típico en eventos tipo G.

Se concluye que el algoritmo eASA tuvo mejor desempeño al clasificar las fallas bifásicas con contacto a tierra.

5.1.2.3 Falla monofásica

Analizando los resultados del algoritmo ASA, el 84% de las fallas fueron clasificadas como Tipo B, un 3% como Tipo D, un 0,4% como Tipo C, un 0,7% como tipo F y el 11,9% restante como tipo N (evento no detectado). Es decir, el 1,1% de las fallas fueron clasificadas erróneamente como eventos provocados por fallas bifásicas, mejorando levemente su desempeño al analizado en la estación de medición 1.

Analizando los resultados del algoritmo eASA, el 75,85% de las fallas fueron clasificadas como Tipo D, un 7,3% como tipo B, un 5,3% como tipo C y el 11,55% restante como Tipo N (evento no detectado). En este caso el 5,3% de las fallas fueron clasificadas erróneamente como eventos generados por fallas bifásicas.

En ambos algoritmos hubo clasificaciones tipo B y D, siendo ambos eventos aquellos que se generan por fallas del tipo monofásica. Para analizar las causas de estas diferencias se presentarán una serie de caso extraídos de los resultados.

Para este primer ejemplo se presenta una falla monofásica en el nodo 856:

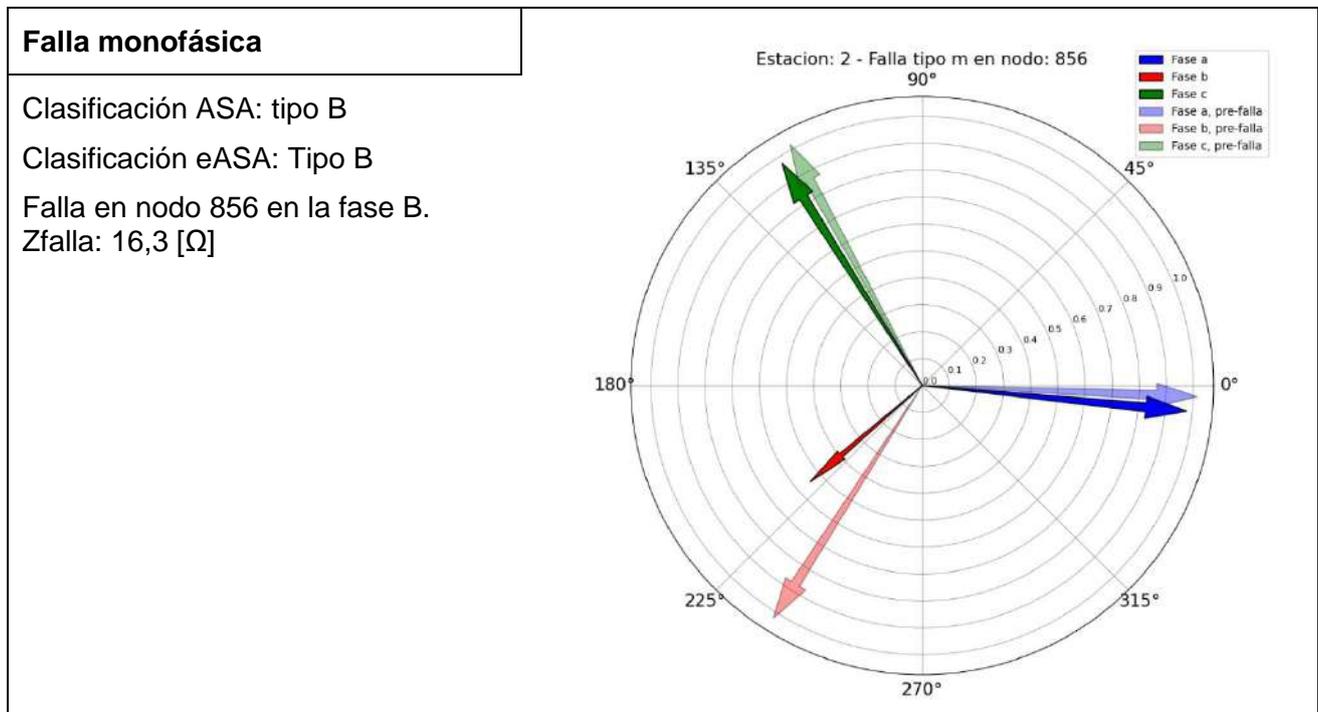


Figura 5-17. Falla monofásica en nodo 856, registrada en estación de medición 2.

Tabla 5-11. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 856, registrada en estación de medición 2.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
0,979	0,546	0,976	0,815	0,686	1,013	0,546	0,125	0,826	0,197	0,158	0,861	0,158	b	ASA
0,979	0,546	0,976	0,815	0,686	1,013	0,546	0,125	0,826	0,197	0,158	0,861	0,158	b	eASA

En este caso, ambos algoritmos detectan que el evento fue del tipo B. Se observa que las tres fases presentan desfasajes, siendo la fase B la más afectada y en la cual se simuló la falla. Las fases A y C tienden a separarse, comportamiento típico en eventos Tipo D. Sin embargo, existe una componente homopolar con magnitud considerable y es la razón por la cual ambos algoritmos clasifican a la falla como Tipo B.

Se presenta una falla en el mismo nodo, pero con una impedancia de falla mayor. Para este caso, el algoritmo eASA clasificó erróneamente como un evento tipo C.

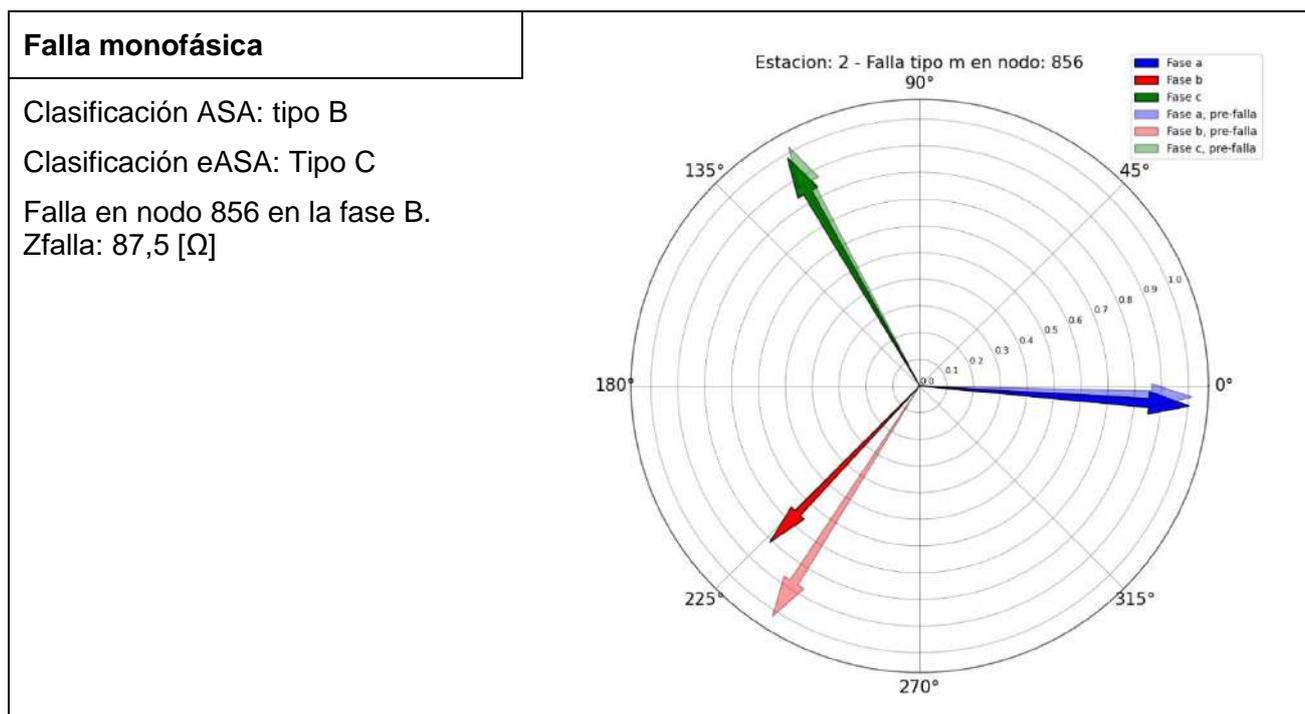


Figura 5-18. Falla monofásica en nodo 856, registrada en estación de medición 2.

Tabla 5-12. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 856, registrada en estación de medición 2.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
1,005	0,809	0,987	0,948	0,833	1,016	0,809	0,067	0,929	0,106	0,070	0,949	0,070	b	ASA
1,005	0,809	0,987	0,948	0,833	1,016	0,809	-0,067	0,929	0,106	0,000	0,914	0,105	c	eASA

Si se comparan los fasores de esta falla y la presentada anteriormente, se observa que el comportamiento es similar. Sin embargo, la fase B afectada cuando la impedancia de falla aumenta, provoca una menor una menor disminución en su magnitud, es decir un hueco de tensión menos profundo. A su vez, la fase C tiene una leve disminución en su magnitud. Si se analizan las componentes simétricas, para el caso del algoritmo eASA se observa que la componente homopolar tiene signo negativo, asignado luego de comparar su desfase con la componente directa. Esto genera que el menor error cuadrático medio sea el de un evento Tipo C. Esto indica que, en ciertos casos, este criterio del algoritmo de asignarle un signo a la componente homopolar puede perjudicar la clasificación del evento, aunque también es cierto que para las fallas bifásicas disminuye notablemente el error de clasificación si se lo compara con el algoritmo ASA.

Finalmente, se muestra un caso en el cual el algoritmo eASA logró clasificar correctamente un evento donde hay presente un corrimiento de las fases no afectadas, B y C, hacia la fase afectada A.

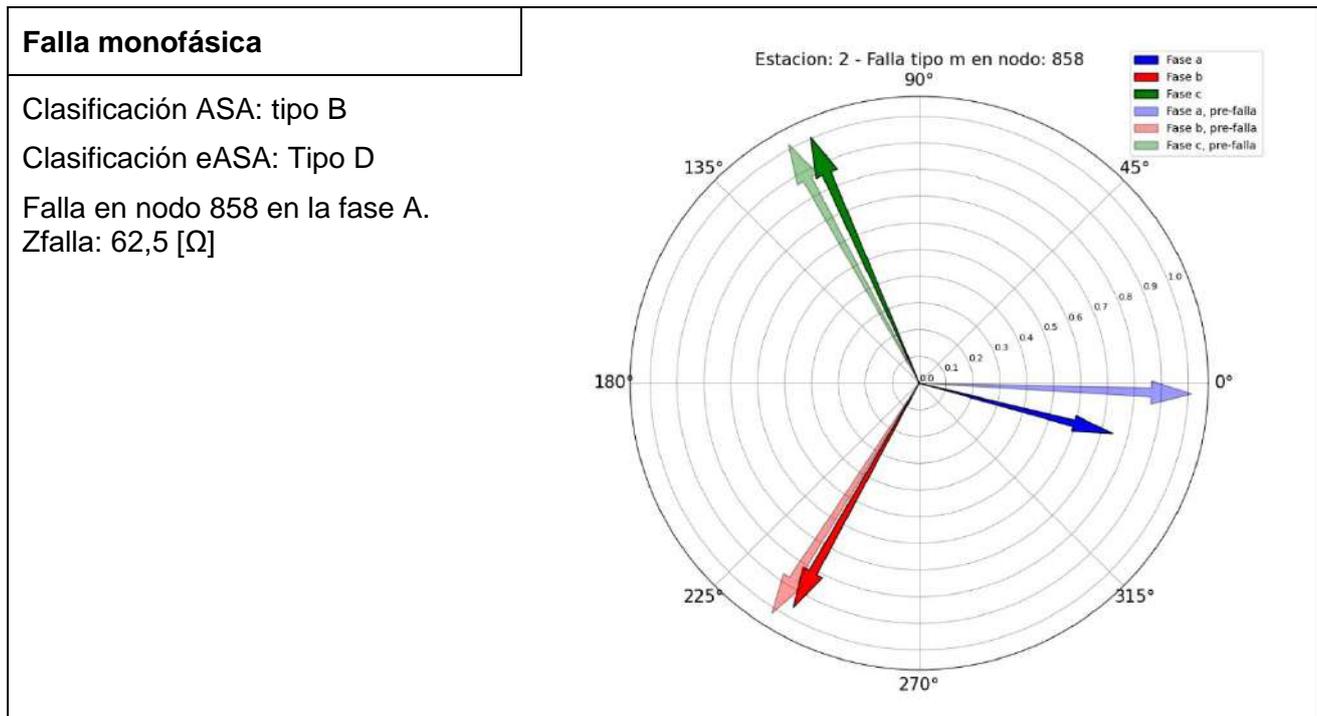


Figura 5-19. Falla monofásica en nodo 858, registrada en estación de medición 2.

Tabla 5-13. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 858, registrada en estación de medición 2.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V _{mín}	V ₀₁	V ₊₁	V ₋₁	V _{01 mod}	V _{+1 mod}	V _{-1 mod}	Evento	Modelo
0,745	0,963	1,007	0,783	1,019	0,913	0,745	0,062	0,900	0,136	0,091	0,928	0,091	b	ASA
0,745	0,963	1,007	0,783	1,019	0,913	0,745	-0,062	0,900	-0,136	0,000	0,882	-0,137	d	eASA

Analizando la Figura 5-16, más precisamente a las componentes simétricas, se observa como el algoritmo eASA corrige el signo en la componente inversa, coincidiendo en signo tanto en la componente inversa del modelo teórico como en el calculado de las mediciones de la simulación. Esto produce que el error cuadrático medio para el evento tipo D disminuya en comparación con los demás posibles eventos.

5.1.3 Estación de medición 5

En la Figura 5-20 se presentan los resultados de los algoritmos para las mediciones efectuadas en la estación 5.

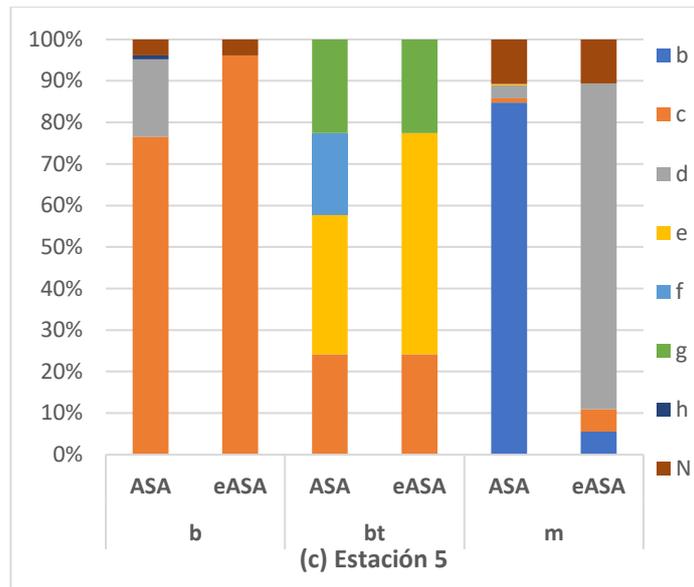


Figura 5-20. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 5.

5.1.3.1 Falla bifásica sin contacto a tierra

Analizando los resultados del algoritmo ASA, el 76,5% de las fallas fueron clasificadas como Tipo C, un 18,6 % como Tipo D, un 1% como Tipo H y el 3,9% restante como tipo N (evento no detectado). Si se compara estos resultados con los resultados de la estación 2 analizada previamente, los resultados para esta falla son prácticamente igual. Es decir, las conclusiones desarrolladas en la estación 2 también aplican a este caso.

Analizando los resultados del algoritmo eASA el 96,1% de las fallas fueron clasificadas como Tipo C y el 3,9% restante fue clasificado como Tipo N (evento no detectado). Los resultados resultaron ser prácticamente iguales a los arribados en la estación 2.

5.1.3.2 Falla bifásica con contacto a tierra

En los resultados del algoritmo ASA el 24,2% de las fallas fueron clasificadas como Tipo C, un 33,7% como Tipo E, un 19,5% como Tipo F y un 22,6% como Tipo G. Es decir, el algoritmo clasificó erróneamente al 19,5% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. Si se compara con los resultados obtenidos en la estación de medición 2, el desempeño del algoritmo mejoró aumentando las clasificaciones del tipo E. Esto se debe a que las $V_{\text{mínima}}$ de las fallas que en esta ocasión fueron clasificadas correctamente resultaron ser tensiones de línea sobre raíz de tres y no tensiones de fase como sucedía previamente. Es decir, el algoritmo no descartó previamente los eventos generados por fallas bifásicas. Para evaluar la sensibilidad del algoritmo en función de la distancia de la falla se realizará un análisis particular en la próxima sección del presente

capítulo. Sin embargo, estos resultados dan indicios de que este algoritmo presenta cambios al variar el sitio de medición de la falla.

Analizando los resultados del algoritmo eASA el 24,2% de las fallas fueron clasificadas como Tipo C, un 53,2% como Tipo E y el 22,6% restante como Tipo G, obteniendo resultados prácticamente idénticos a los obtenidos en la estación analizada previamente para este tipo de falla.

5.1.3.3 Falla monofásica

Analizando los resultados del algoritmo ASA, el 84,8% de las fallas fueron clasificadas como Tipo B, un 1% como Tipo C, un 3,16% como Tipo D, un 0,4% como tipo E y el 10,64% restante como tipo N (evento no detectado). Es decir, el 1,4% de las fallas fueron clasificadas como eventos generados por fallas bifásicas, siendo prácticamente el mismo error que el obtenido para la estación de medición 2.

Analizando los resultados del algoritmo eASA, el 78,4% de las fallas fueron clasificadas como Tipo D, un 5,5% como tipo B, un 5,4% como tipo C y el 10,7% restante como Tipo N (evento no detectado). En este caso el 5,4% de las fallas fueron clasificadas erróneamente como eventos provocados por fallas bifásicas.

El desempeño de los algoritmos es prácticamente el mismo que para el analizado en la estación 2, con lo cual se arriban a las mismas conclusiones que las desarrolladas previamente.

5.1.4 Estación de medición 6

En la Figura 5-21 se presentan los resultados de los algoritmos para las mediciones efectuadas en la estación 6.

A partir de la estación 6, es decir también para las estaciones 7, 8, 9 y 10, se observó que las muestras de fallas bifásicas sin y con contacto a tierra disminuyeron en cantidad. Se recuerda que, para realizar los análisis presentados, se comenzó filtrando aquellas fallas con una tensión mínima mayor a 0.1 [p.u], y descartando aquellas con una $V_{\text{mínima}}$ debajo de ese valor. Esto significa que, a partir de este punto de la red, los huecos de tensión generados por fallas bifásicas tuvieron una profundidad mayor al criterio impuesto. Este comportamiento aparece cuando las fallas bifásicas se produjeron aguas arriba a la estación de medición.

Con esto se quiere dejar el precedente de que los análisis siguientes para las fallas bifásicas tienen un muestreo de datos significativamente menor, y que se manifiesta en mayor medida para las fallas bifásicas con contacto a tierra. Como conclusión, debería analizarse aplicar un cambio en la forma en que se simularon estas fallas, por ejemplo, agregando impedancias

entre las líneas antes de la impedancia que conecta a tierra, y tal vez aumentar los valores de la impedancia de falla a tierra.

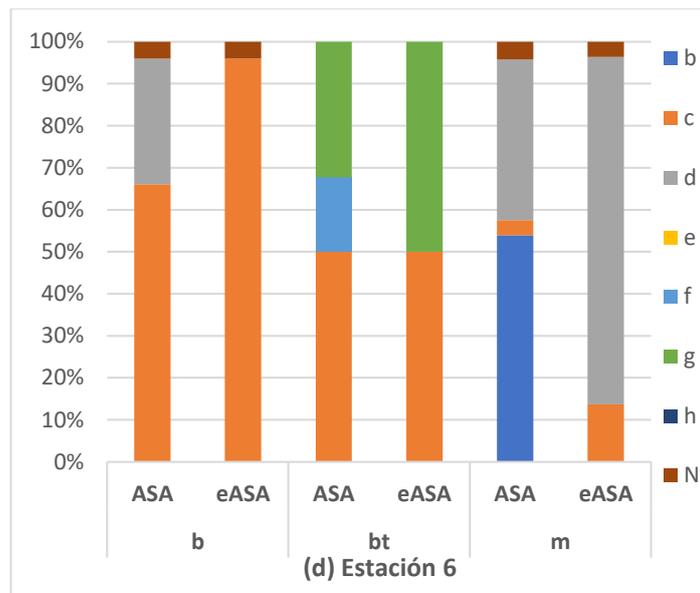


Figura 5-21. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 6.

5.1.4.1 Falla bifásica sin contacto a tierra

Analizando los resultados del algoritmo ASA, el 66 % de las fallas fueron clasificadas como Tipo C, un 30 % como Tipo D y el 4% restante como tipo N (evento no detectado). El algoritmo clasifica erróneamente al 30% de la muestra considerando que el evento fue provocado por una falla monofásica. Las conclusiones desarrolladas en las estaciones de medición previas también aplican a este caso.

Analizando los resultados del algoritmo eASA el 96 % de las fallas fueron clasificadas como Tipo C y el 4% restante fue clasificado como Tipo N (evento no detectado). Los resultados resultaron ser prácticamente iguales a los arribados en la estación 5. Se destaca el desempeño del algoritmo eASA para este tipo de falla.

5.1.4.2 Falla bifásica sin contacto a tierra

Analizando los resultados del algoritmo ASA, el 50% de las fallas fueron clasificadas como Tipo C, un 17,8% como Tipo F y un 32,2% como Tipo G. Es decir, el algoritmo clasificó erróneamente al 17,8% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. La razón de este error en la clasificación es la misma que las desarrolladas previamente para este tipo de falla.

Analizando los resultados del algoritmo eASA el 50% de las fallas fueron clasificadas como Tipo C y el otro 50% como tipo G. Si bien los eventos C y G son provocados por distintas

razones, ambos diagramas fasoriales son muy similares, y en los dos casos, se generan a partir de una falla bifásica. Teniendo en cuenta esto, el algoritmo eASA presenta un mejor desempeño que el ASA.

5.1.4.3 Falla monofásica

El algoritmo ASA clasificó como evento Tipo B al 54% de las fallas monofásicas, un 38,2% fueron clasificadas como tipo D, un 3,6% fueron clasificadas como tipo C, y el 4,2% restante como tipo N (evento no detectado). Estos resultados muestran que el algoritmo ASA aumentó las clasificaciones tipo D a comparación de las estaciones de mediciones previas. En las próximas estaciones de medición esta tendencia aumenta.

Por otro lado, el algoritmo eASA clasificó como evento Tipo D al 82,7% de las fallas monofásicas, un 13,7% fueron clasificadas como tipo C y el 3,6% restante como Tipo N (evento no detectado). Estos resultados fueron iguales a los obtenidos para la estación 5.

Comparando ambos algoritmos, el eASA fue el que presentó mayor error de clasificación, clasificando erróneamente el 13,7% de los casos como eventos generados por una falla bifásica.

Para intentar sacar conclusiones del comportamiento de estos algoritmos en este punto de la red se analiza una falla monofásica en el nodo 888 donde ambos métodos detectaron que la falla fue del tipo D:

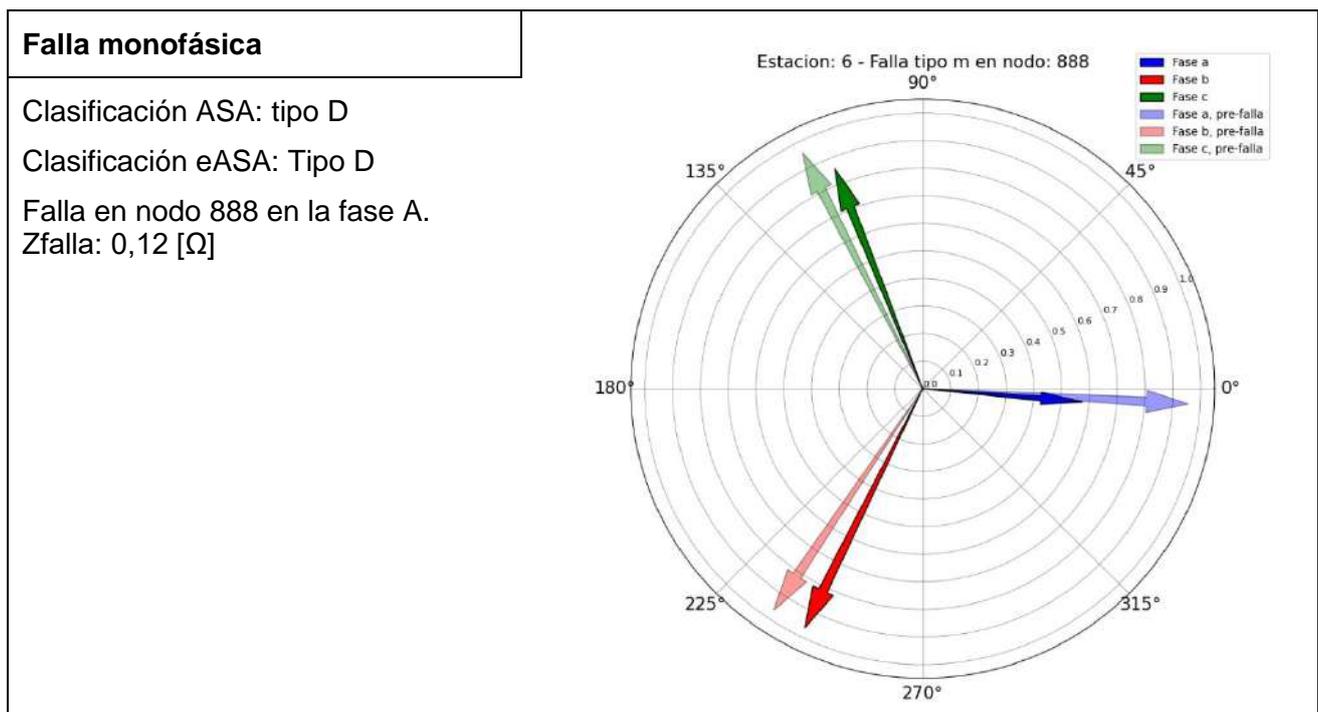


Figura 5-22. Falla monofásica en nodo 888, registrada en estación de medición 6.

Tabla 5-14. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 888, registrada en estación de medición 6.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
0,578	0,966	0,858	0,747	0,964	0,709	0,578	0,067	0,797	0,168	0,000	0,769	0,192	d	ASA
0,578	0,966	0,858	0,747	0,964	0,709	0,578	-0,067	0,797	-0,168	0,000	0,769	-0,192	d	eASA

Observando la Figura 5-22, esta falla genera una firma típica de un evento tipo D, y ambos algoritmos lo detectan correctamente. Si se analiza la Tabla 5-14, particularmente los valores de las componentes simétricas, se observa que el algoritmo eASA corrige el signo de las componente inversa y homopolar calculadas de las mediciones de la simulación, y coinciden en signo con las componentes simétricas del modelo teórico. Para el caso del algoritmo ASA esto no sucede ya que trabaja con el módulo de las componentes simétricas, pero aun así el evento más probable coincide con el del eASA.

Si se analiza la misma falla, pero con una mayor impedancia de falla, el resultado es diferente ya que el algoritmo ASA lo clasifica como un tipo B y el algoritmo eASA como tipo D.

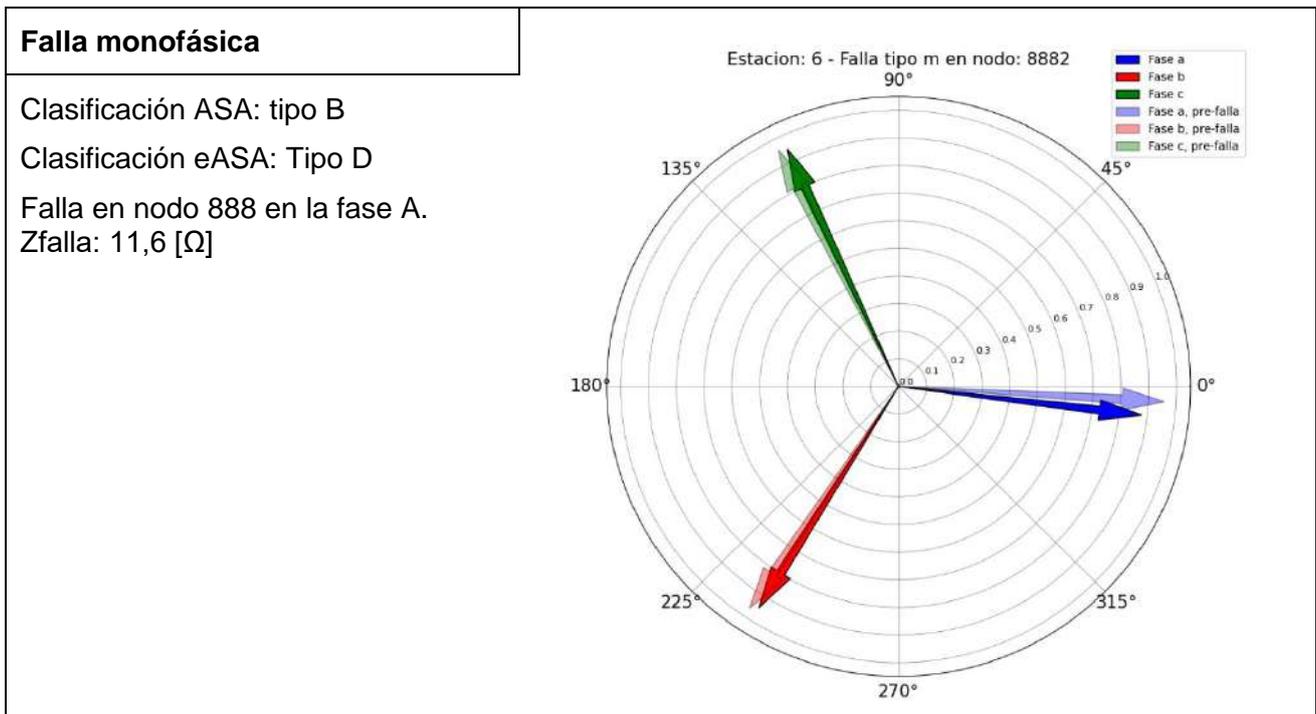


Figura 5-23. Falla monofásica en nodo 888, registrada en estación de medición 6.

Tabla 5-15. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 888, registrada en estación de medición 6.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
0,879	0,948	0,948	0,892	0,961	0,922	0,879	0,019	0,925	0,041	0,027	0,934	0,027	b	ASA
0,879	0,948	0,948	0,892	0,961	0,922	0,879	-0,019	0,925	-0,041	0,000	0,920	-0,041	d	eASA

Nuevamente la diferencia principal entre los algoritmos es el signo de las componentes simétricas inversas y homopolares. Para el algoritmo eASA el error entre el valor del modelo teórico y el calculado de la simulación para la componente inversa resulta nula.

Este comportamiento se repite en otras fallas y en otras estaciones de medición, y es la razón principal por la cual para los casos monofásicos el algoritmo ASA clasifica en mayor porcentaje a los eventos tipo B y el algoritmo eASA clasifica en mayor medida a los eventos como tipo D. Se concluye que el algoritmo eASA distingue con un mejor desempeño la presencia de desfasajes en las fases que no sufrieron la falla, en este caso los desfasajes de las fases B y C.

5.1.5 Estación de medición 7

Esta estación de medición se ubica en el lado secundario al transformador de baja tensión. La distribución de los eventos de tensión detectados por los algoritmos ASA y eASA se observa a continuación:

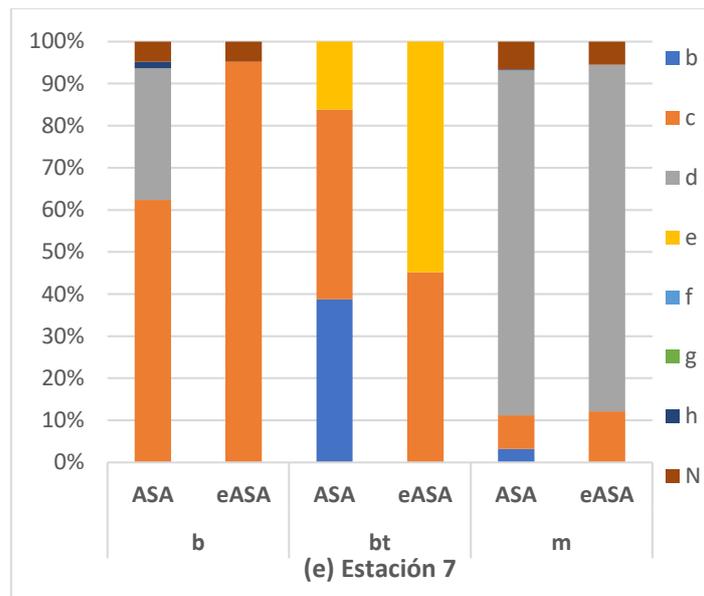


Figura 5-24. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 7.

Al igual que lo sucedido en la estación de medición 6, el muestreo de las fallas bifásicas se vio disminuido drásticamente por generar huecos de tensión menores a 0.1 [p.u]. En la Figura 5-25 se ejemplifica el comportamiento observado en este tipo de fallas, que se ve incrementado al disminuir las impedancias de falla.

5.1.5.1 Falla bifásica sin contacto a tierra.

Analizando los resultados del algoritmo ASA, el 62,4 % de las fallas fueron clasificadas como Tipo C, un 31,2 % como Tipo D, un 1,6% como tipo H y el 4,8% restante como tipo N (evento no detectado). El algoritmo clasifica erróneamente al 32,8% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. Las conclusiones desarrolladas en las estaciones de medición previas también aplican a este caso.

Analizando los resultados del algoritmo eASA el 95,2 % de las fallas fueron clasificadas como Tipo C y el 4,8% restante fue clasificado como Tipo N (evento no detectado).

Los resultados son prácticamente iguales a los arribados en la estación 6. Esto significa que el transformador no generó cambios en la firma de la falla. Este comportamiento es esperado por tratarse de un transformador $Y_n Y_n$, con lo cual no se producen cambios en las fases como si sucedería por ejemplo con un transformador Dy. Se destaca el desempeño del algoritmo eASA para este tipo de falla.

5.1.5.2 Falla bifásica con contacto a tierra

En los resultados del algoritmo ASA el 38,7% de las fallas fueron clasificadas como Tipo B, un 45,2% como Tipo C y un 16,2% como Tipo E. Es decir, el algoritmo clasificó erróneamente al 38,7% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. La razón de este error en la clasificación es la misma que las desarrolladas previamente para este tipo de falla.

Analizando los resultados del algoritmo eASA el 45,2% de las fallas fueron clasificadas como Tipo C y el otro 54,8% como tipo E. Si bien los eventos C y E son provocados por distintas razones, ambos diagramas fasoriales son muy similares, y en los dos casos, se generan a partir de una falla bifásica.

El problema detectado en las otras estaciones de medición, correspondiente a los grandes desfases en las fases afectadas por este tipo de fallas, se vio incrementado en este punto de la red. Para ejemplificar esto, se analiza el siguiente caso donde se analiza una falla en el nodo 834.

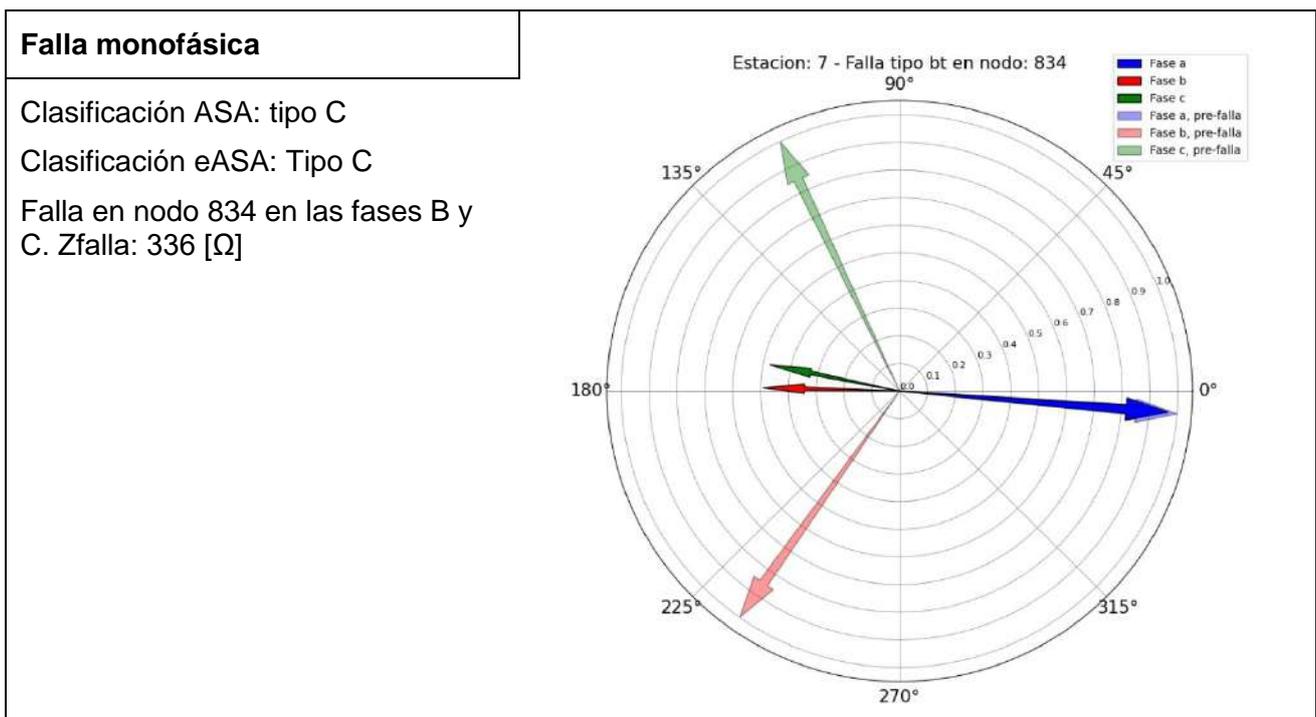


Figura 5-25. Falla monofásica en nodo 834, registrada en estación de medición 7.

Tabla 5-16. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 834, registrada en estación de medición 7.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
0,967	0,492	0,476	0,842	0,050	0,832	0,050	0,011	0,508	0,459	0,000	0,525	0,475	c	ASA
0,967	0,492	0,476	0,842	0,050	0,832	0,050	0,011	0,508	0,459	0,000	0,525	0,475	c	eASA

Este comportamiento de grandes desfasajes en las fases afectadas por fallas bifásicas con puesta a tierra aumenta, como se comentó previamente, a partir de las mediciones realizadas a partir de la estación 6 en adelante, incluyendo a las estaciones 7, 8, 9 y 10. Esto se ve potenciado particularmente en la estación de medición 7, principalmente en las magnitudes de las tensiones afectadas producto del cambio de tensión generado por el transformador. Nuevamente se remarca la importancia de generar fallas bifásicas que provoquen huecos de tensión de menor profundidad.

Las fallas simuladas en el mismo nivel de tensión y aguas abajo a la estación de medición, es decir en el nodo 890, no tuvieron este comportamiento de grandes desfasajes y magnitudes de tensiones muy pequeñas. A modo de ejemplo se presenta el siguiente caso:

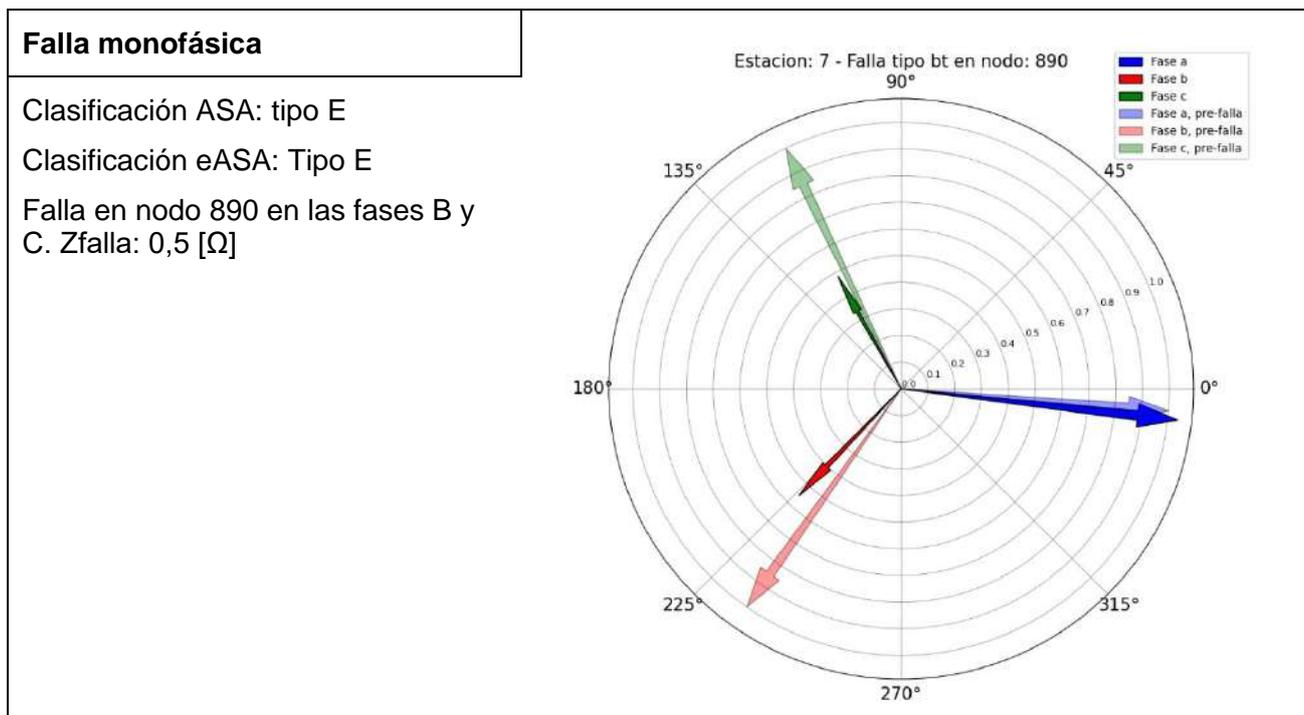


Figura 5-26. Falla monofásica en nodo 890, registrada en estación de medición 7.

Tabla 5-17. Valores de tensión RMS, V_{\min} , componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 890, registrada en estación de medición 7.

V_a	V_b	V_c	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V_{\min}	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
1,037	0,553	0,483	0,832	0,482	0,795	0,482	0,141	0,689	0,209	0,172	0,655	0,172	e	ASA
1,037	0,553	0,483	0,832	0,482	0,795	0,482	0,141	0,689	0,209	0,172	0,655	0,172	e	eASA

Se observa una rotación simétrica en las tres fases, y además las fases afectadas B y C presentan rotaciones en sus fases provocando que tiendan a acercarse. En este caso ambos algoritmos clasificaron correctamente el evento de tensión.

5.1.5.3 Falla monofásica

El algoritmo ASA clasificó como evento Tipo D al 82% de la muestra, un 3,2% fueron clasificadas como tipo B, un 8% fueron clasificadas como tipo C y el 6,8% restante como tipo N (evento no detectado). Estos resultados muestran un cambio significativo en la clasificación efectuada por el algoritmo ASA, manteniendo la tendencia observada en la estación de medición 6 donde la clasificación de los eventos como tipo D aumentó.

Por otro lado, el algoritmo eASA clasificó como evento Tipo D al 82,4% de las fallas monofásicas, un 12,1% fueron clasificadas como tipo C y el 5,5% restante como Tipo N (evento no detectado). Estos resultados fueron prácticamente iguales a los obtenidos para la estación 6.

Comparando ambos algoritmos, el eASA fue el que presentó mayor error al clasificar erróneamente el 12,1% de las fallas, seleccionando un tipo de evento que se genera a raíz de una falla bifásica.

Se analizarán dos casos de fallas monofásicas generadas en una misma región del circuito, pero aplicadas en distintas fases. Por un lado, la falla en el nodo 838 se simuló en la falla B dado que la línea es monofásica en dicha fase. También se analiza el caso de la falla monofásica en el nodo 862, nodo encargado de alimentar el nodo 838, simulada en la fase A. Se eligieron estos casos porque los algoritmos los clasificaron erróneamente como un evento Tipo C para el nodo 838 y Tipo D para el caso 862.

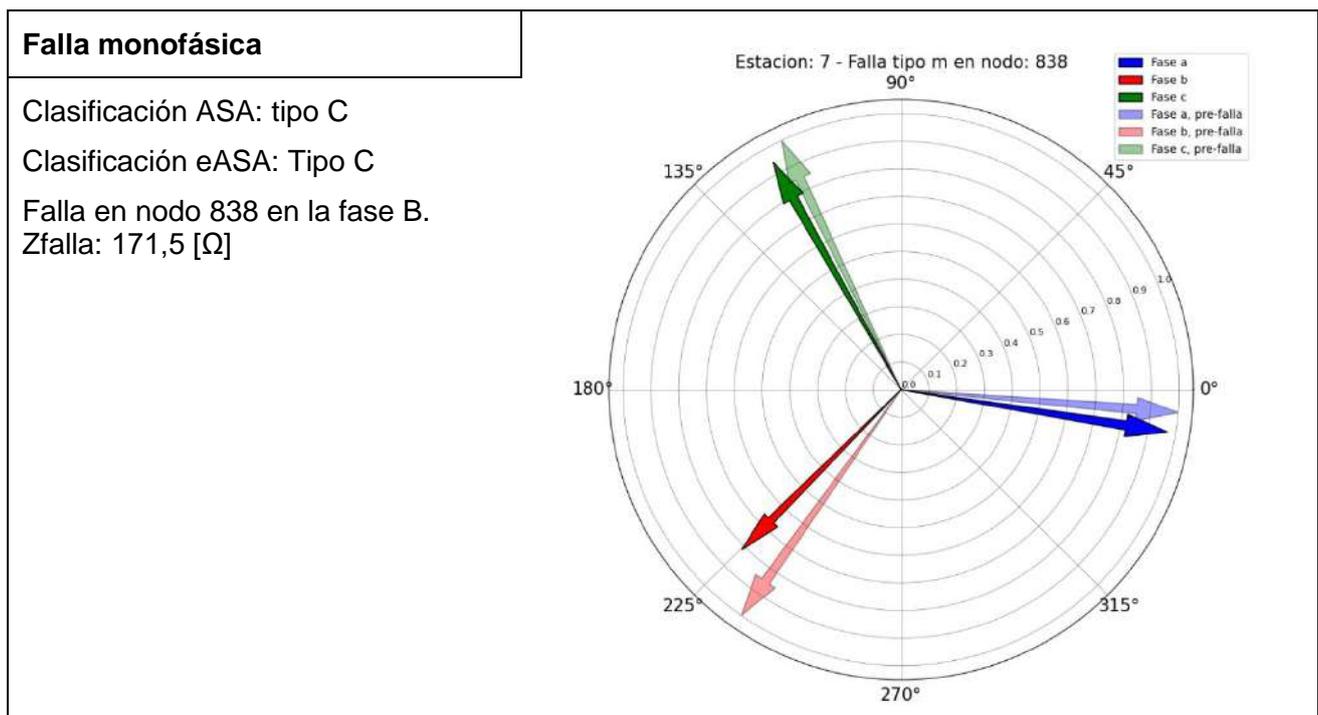


Figura 5-27. Falla monofásica en nodo 838, registrada en estación de medición 7.

Tabla 5-18. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 838, registrada en estación de medición 7.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	V _{mín}	V ₀₁	V ₊₁	V ₋₁	V ₀₁ mod	V ₊₁ mod	V ₋₁ mod	Evento	Modelo
0,971	0,816	0,944	0,918	0,814	0,995	0,814	0,039	0,906	0,105	0,000	0,907	0,093	c	ASA
0,971	0,816	0,944	0,918	0,814	0,995	0,814	-0,039	0,906	0,105	0,000	0,907	0,093	c	eASA

En este caso la fase más afectada fue la B, producto de la falla en dicha fase. Si se analiza la

Tabla 5-8, para el algoritmo eASA la componente homopolar calculada a partir de la simulación tiene signo negativo ya que dicha componente presenta un desfase con la componente directa mayor a $\pm 90^\circ$. Esta es la razón principal por la cual el error cuadrático mínimo corresponde con un evento Tipo C. Si se calcula la componente homopolar teórica del modelo para un evento tipo B, el valor es 0,062 (positivo) y como la componente homopolar que considera el algoritmo eASA es negativa, la diferencia entre esta componente y la del modelo teórico aumenta, incrementado el error cuadrático medio.

Hay que tener en cuenta que la componente homopolar es prácticamente nula, con lo cual esta clasificación errónea puede ser producto de una suma de errores, desde redondeos hasta errores propios de los métodos de medición, entre otros. Aun así, se decidió presentar este caso porque este error puede significar que lo más conveniente para analizar el desempeño de los algoritmos en redes desequilibradas sea probarlo en fallas generadas en sus tres fases y analizar cómo influye el desequilibrio propio de la red en el desempeño del algoritmo.

Si se analiza la falla monofásica en el nodo 862 para la misma impedancia de falla, pero ahora implementada en la fase A se obtienen los siguientes resultados.

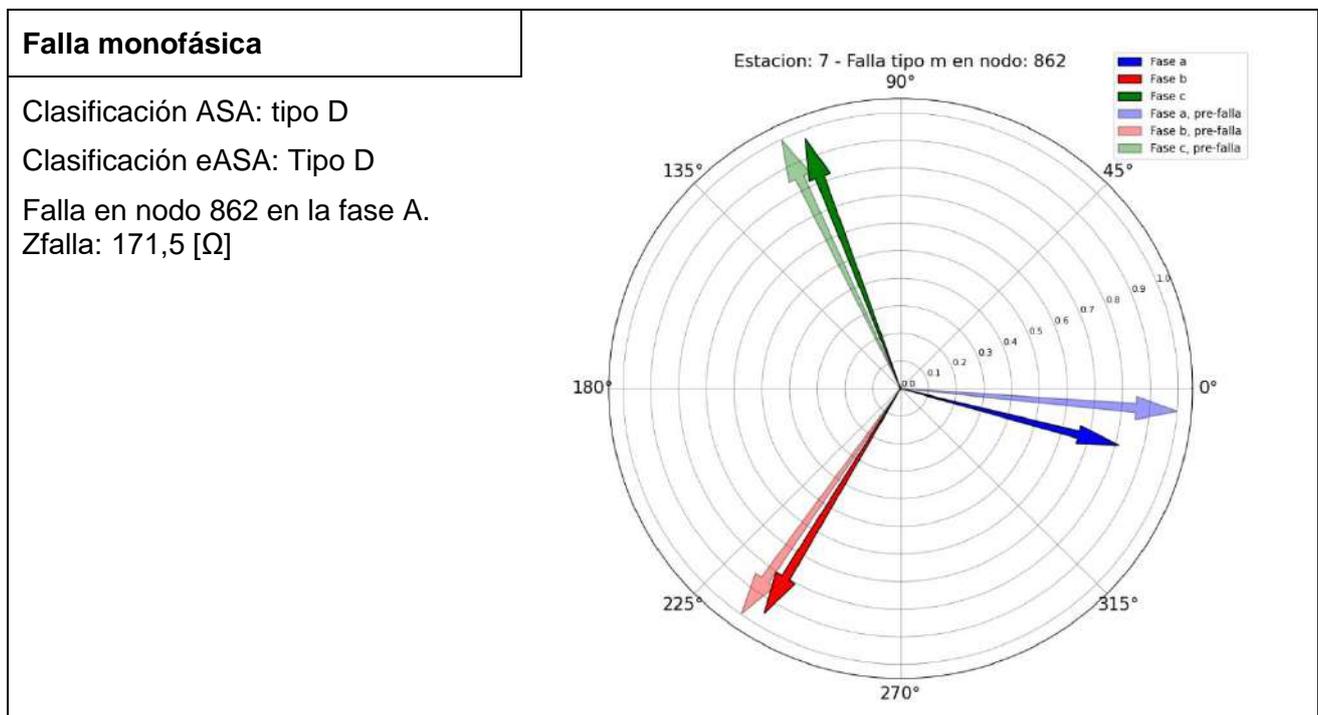


Figura 5-28. Falla monofásica en nodo 862, registrada en estación de medición 7.

Tabla 5-19. Valores de tensión RMS, $V_{mín}$, componentes simétricas reales y del modelo, junto con el tipo de evento detectado. Falla monofásica en nodo 862, registrada en estación de medición 7.

Va	Vb	Vc	$\frac{V_{ab}}{\sqrt{3}}$	$\frac{V_{bc}}{\sqrt{3}}$	$\frac{V_{ca}}{\sqrt{3}}$	$V_{mín}$	V_{01}	V_{+1}	V_{-1}	V_{01} mod	V_{+1} mod	V_{-1} mod	Evento	Modelo
----	----	----	---------------------------	---------------------------	---------------------------	-----------	----------	----------	----------	-----------------	-----------------	-----------------	--------	--------

0,813	0,953	0,970	0,818	0,999	0,916	0,813	0,042	0,908	0,104	0,000	0,906	0,093	d	ASA
0,813	0,953	0,970	0,818	0,999	0,916	0,813	-0,042	0,908	-0,104	0,000	0,906	-0,093	d	eASA

En este caso se observa en la Figura 5-28 que la fase más afectada es la fase A. La primera diferencia observada aparece en la componente inversa calculada a partir de la simulación, donde para el caso del algoritmo eASA tiene el signo negativo, evidencia de que la componente inversa y la componente directa tienen un desfase mayor a $\pm 90^\circ$. A su vez, la componente inversa del modelo teórico también es negativa, presentando una menor diferencia entre componentes inversas, que resulta en un error cuadrático medio más pequeño.

Si se analiza el algoritmo ASA se llega a la misma conclusión, es decir, clasifica como tipo D, pero lo hace recorriendo un camino distinto. El algoritmo ASA descarta que pueda tratarse de una falla bifásica en su primera toma de decisión, cuando evalúa si el valor de tensión mínimo es de fase o de línea sobre raíz de tres. Si no hiciera este primer descarte, no podría decidir entre un evento tipo C y uno tipo D ya que sus componentes simétricas teóricas del modelo utilizado son idénticas.

Si bien la diferencia es mínima, se analizó este caso porque nos indica varios aspectos importantes. Por un lado, implementar estos algoritmos en sistemas desequilibrados resulta interesante debido a que fallas en un mismo sitio en función de su fase pueden generar eventos de tensión similares, pero con diferencias suficientes para que el algoritmo de clasificación las considere diferentes. También este caso sirvió para evidenciar que asignarles un signo a las componentes simétricas tiene muy buenos resultados para clasificar fallas bifásicas, pero en el caso de fallas monofásicas, en casos puntuales, suele introducir algunos errores.

5.1.6 Estación de medición 8

En la Figura 5-29 se presentan los resultados de los algoritmos al detectar fallas en la estación de medición 8.

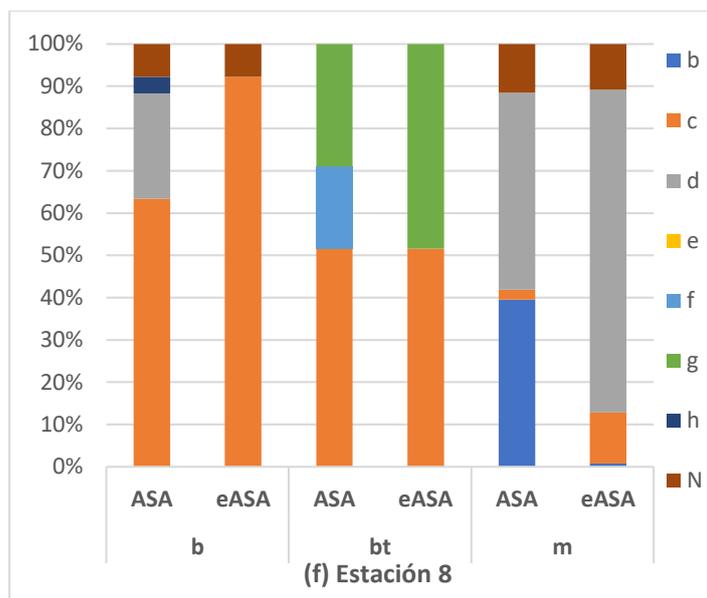


Figura 5-29. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 8.

5.1.6.1 Fallas bifásicas sin contacto a tierra

Analizando los resultados del algoritmo ASA, el 63,4 % de las fallas fueron clasificadas como Tipo C, un 24,9 % como Tipo D, un 3,9% como tipo H y el 7,8% restante como tipo N (evento no detectado). El algoritmo clasifica erróneamente al 28,8% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica, mostrando resultados muy similares a los presentados en la estación de medición 7. Las conclusiones desarrolladas en las estaciones de medición previas también aplican a este caso.

Analizando los resultados del algoritmo eASA el 92,2 % de las fallas fueron clasificadas como Tipo C y el 7,8% restante fue clasificado como Tipo N (evento no detectado). Los resultados resultaron ser prácticamente iguales a los arribados en la estación 7.

5.1.6.2 Fallas bifásicas con contacto a tierra

En los resultados del algoritmo ASA el 51,6% de las fallas fueron clasificadas como Tipo C, un 19,4% como Tipo F y el 29% restante como Tipo G. Es decir, el algoritmo clasificó erróneamente al 19,4% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. La razón de este error en la clasificación es la misma que las desarrolladas previamente para este tipo de falla.

Analizando los resultados del algoritmo eASA el 51,6% de las fallas fueron clasificadas como Tipo C y el otro 48,4% como tipo G.

Los resultados obtenidos son prácticamente igual a los obtenidos en la estación de medición 6. Nuevamente se remarca que la muestra de datos se redujo en gran medida para este tipo de falla.

5.1.6.3 Fallas monofásicas

El algoritmo ASA clasificó como evento Tipo B al 39,5% de las fallas, un 46,6% fueron clasificadas como tipo D, un 2,35% fueron clasificadas como tipo C, y el 11,55% restante como tipo N (evento no detectado). Estos resultados muestran que el algoritmo ASA aumentó las clasificaciones tipo D, con resultados similares a los obtenidos en la estación de medición 6.

Por otro lado, el algoritmo eASA clasificó como evento Tipo D al 76,3% de las fallas monofásicas, un 1% como tipo B, un 12% fueron clasificadas como tipo C y el 10,7% restante como Tipo N (evento no detectado).

Comparando ambos algoritmos, el eASA fue el que presentó mayor error de clasificación, clasificando erróneamente el 12% de los casos como eventos generados por una falla bifásica. Estos resultados son similares a los obtenidos en la estación de medición 6, por lo tanto, también se aplican sus conclusiones.

5.1.7 Estación de medición 9

En la Figura 5-30 se presentan las clasificaciones de eventos de los algoritmos para mediciones en la estación 9.

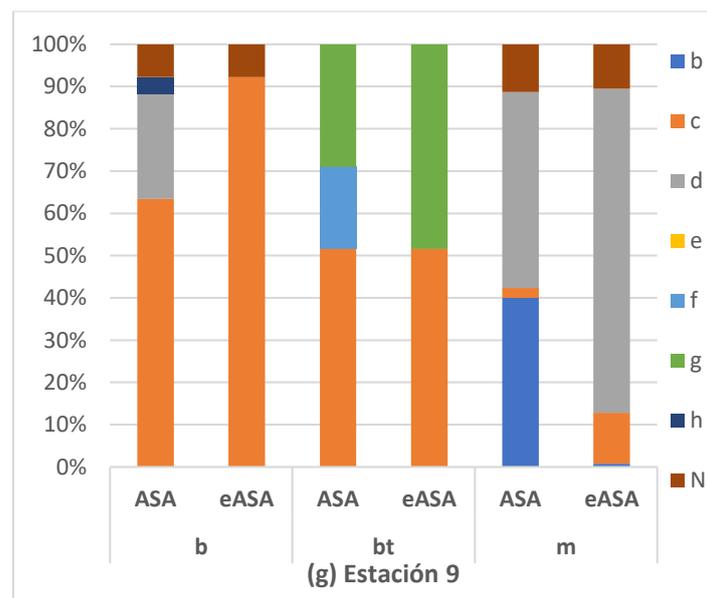


Figura 5-30. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 9.

Los resultados obtenidos son prácticamente iguales a los resultados de la estación de medición 8. Por lo tanto, el análisis realizado previamente también se aplica a este caso.

5.1.7.1 Fallas bifásicas sin contacto a tierra

Analizando los resultados del algoritmo ASA, el 63,4 % de las fallas fueron clasificadas como Tipo C, un 24,9 % como Tipo D, un 3,9% como tipo H y el 7,8% restante como tipo N (evento no detectado). El algoritmo clasifica erróneamente al 28,8% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica, mostrando resultados iguales a los presentados en la estación de medición 8.

Analizando los resultados del algoritmo eASA el 92,2 % de las fallas fueron clasificadas como Tipo C y el 7,8% restante fue clasificado como Tipo N (evento no detectado). Los resultados son idénticos a los arribados en la estación 8.

5.1.7.2 Fallas bifásicas con contacto a tierra

En los resultados del algoritmo ASA el 51,6% de las fallas fueron clasificadas como Tipo C, un 19,4% como Tipo F y el 29% restante como Tipo G. Es decir, el algoritmo clasificó erróneamente al 19,4% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. La razón de este error en la clasificación es la misma que las desarrolladas previamente para este tipo de falla.

Analizando los resultados del algoritmo eASA el 51,6% de las fallas fueron clasificadas como Tipo C y el otro 48,4% como tipo G.

Los resultados obtenidos son iguales a los obtenidos en la estación de medición 8. Nuevamente se remarca que la muestra de datos se redujo en gran medida para este tipo de falla.

5.1.7.3 Fallas monofásicas

El algoritmo ASA clasificó como evento Tipo B al 40% de las fallas, un 46,4% fueron clasificadas como tipo D, un 2,35% fueron clasificadas como tipo C, y el 11,25% restante como tipo N (evento no detectado). Los resultados son prácticamente iguales a los obtenidos en la estación de medición 8.

Por otro lado, el algoritmo eASA clasificó como evento Tipo D al 76,7% de las fallas monofásicas, un 0,8% como tipo B, un 12% fueron clasificadas como tipo C y el 10,5% restante como Tipo N (evento no detectado).

Comparando ambos algoritmos, el eASA fue el que presentó mayor error de clasificación, clasificando erróneamente el 12% de los casos como eventos generados por una falla bifásica. Estos resultados son iguales a los obtenidos en la estación de medición 8, por lo tanto, también se aplican sus conclusiones.

5.1.8 Estación de medición 10

En la Figura 5-31 se muestran los resultados de las clasificaciones de eventos de los algoritmos en la estación de medición 10.

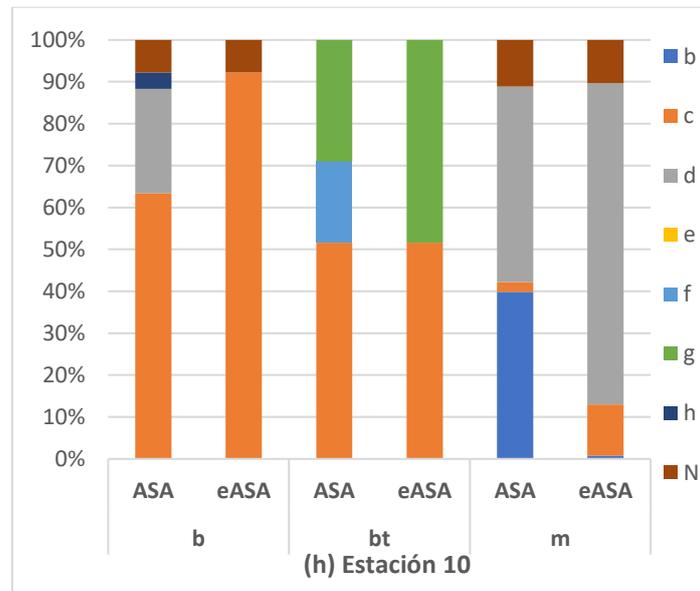


Figura 5-31. Distribución de eventos de tensión detectados por los algoritmos ASA y eASA para las tres fallas simuladas (bifásica, bifásica con puesta a tierra y monofásica) en la estación de medición 10.

Los resultados obtenidos son prácticamente iguales a los resultados de la estación de medición 8 y 9.

5.1.8.1 Fallas bifásicas sin contacto a tierra

Analizando los resultados del algoritmo ASA, el 63,4 % de las fallas fueron clasificadas como Tipo C, un 24,9 % como Tipo D, un 3,9% como tipo H y el 7,8% restante como tipo N (evento no detectado). El algoritmo clasifica erróneamente al 28,8% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica, mostrando resultados iguales a los presentados en la estación de medición 8.

Analizando los resultados del algoritmo eASA el 92,2 % de las fallas fueron clasificadas como Tipo C y el 7,8% restante fue clasificado como Tipo N (evento no detectado). Los resultados resultaron ser iguales a los arribados en la estación 8 y 9.

5.1.8.2 Fallas bifásicas con contacto a tierra

En los resultados del algoritmo ASA el 51,6% de las fallas fueron clasificadas como Tipo C, un 19,4% como Tipo F y el 29% restante como Tipo G. Es decir, el algoritmo clasificó erróneamente al 19,4% de la muestra al seleccionar como evento más probable uno provocado por una falla monofásica. La razón de este error en la clasificación es la misma que las desarrolladas previamente para este tipo de falla.

Analizando los resultados del algoritmo eASA el 51,6% de las fallas fueron clasificadas como Tipo C y el otro 48,4% como tipo G.

Los resultados obtenidos son iguales a los obtenidos en la estación de medición 8 y 9. Nuevamente se remarca que la muestra de datos se redujo en gran medida para este tipo de falla.

5.1.8.3 Fallas monofásicas

El algoritmo ASA clasificó como evento Tipo B al 39,8% de las fallas, un 44% fueron clasificadas como tipo D, un 2,35% fueron clasificadas como tipo C, y el 13,85% restante como tipo N (evento no detectado). Los resultados son prácticamente iguales a los obtenidos en la estación de medición 8 y 9.

Por otro lado, el algoritmo eASA clasificó como evento Tipo D al 76,6% de las fallas monofásicas, un 0,8% como tipo B, un 12,2% fueron clasificadas como tipo C y el 10,4% restante como Tipo N (evento no detectado).

Comparando ambos algoritmos, el eASA fue el que presentó mayor error de clasificación, clasificando erróneamente el 12,2% de los casos como eventos generados por una falla bifásica. Estos resultados son iguales a los obtenidos en la estación de medición 8 y 9.

5.2 Análisis de resultados sobre muestra representativa

Para realizar un análisis particular de los datos obtenidos con el fin de extraer conclusiones más significativas, se realizó un filtro de los resultados de las simulaciones manteniendo aquellos cuyas tensiones residuales se encontraban entre el 50% y el 90% del valor de tensión nominal.

Para tomar este criterio se utilizó la información proporcionada por *DPQ Statistical Summary Report* [61] del Instituto de Investigación de Energía Eléctrica (del inglés *Electric Power Research Institute, EPRI*). El reporte *DPQ* tiene como objetivo principal recopilar, analizar e informar sobre datos de calidad de energía del sistema de distribución de EE.UU con un grado de importancia estadística. En la Figura 5-32 se indica, mediante un histograma, la frecuencia acumulada de los *dips* y las interrupciones de tensión en función de la tensión residual del hueco de tensión medido. En la Tabla 5-20 se resume esta información.

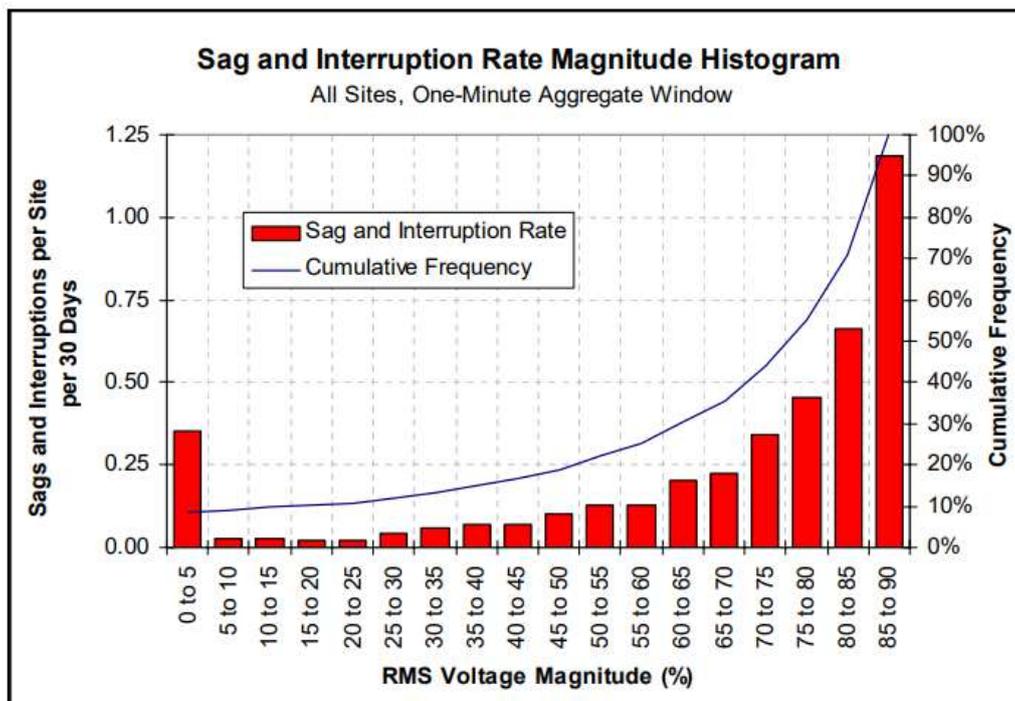


Figura 5-32. Histograma de magnitud de tasa de *dips* e interrupciones de tensión en todos los sitios analizados en el proyecto DPQ. [61]

Voltage (%)	Sag and Interruption Rate	Frequency	Cumulative Frequency
0 to 10	0.38	9.19%	9.19%
10 to 20	0.04	1.08%	10.26%
20 to 30	0.06	1.53%	11.80%
30 to 40	0.13	3.08%	14.87%
40 to 50	0.17	4.11%	18.98%
50 to 60	0.26	6.23%	25.22%
60 to 70	0.43	10.42%	35.63%
70 to 80	0.80	19.39%	55.02%
80 to 90	1.84	44.98%	100.00%

Tabla 5-20. *Dips* e interrupciones en todos los sitios analizados en el proyecto DPQ. [61]

A partir de esta información fue que se decidió realizar el filtro de datos mencionado previamente para comparar el desempeño de los algoritmos ASA y eASA, siendo una muestra que representa el 81,02% de los huecos de tensión detectados en el reporte DPQ.

Una vez efectuado el filtro sobre la muestra se analizaron los resultados nuevamente, pero esta vez se tuvo en cuenta la distancia del nodo de medición al resto de las fallas de la red para determinar si la distancia tiene un efecto sobre el desempeño de los algoritmos y si es necesario modificar la distribución de los puntos de medición en la red.

A partir del análisis realizado en la Sección 5.1, se optó por analizar los resultados de cuatro estaciones de medición que representen en buena medida a la totalidad de los datos. Este criterio se aplicó luego de notar que hubo estaciones de medición que presentaron un comportamiento similar en las clasificaciones realizadas. Para este segundo análisis se

consideraron los resultados extraídos en la estación de medición 1 (Nodo 800), la estación de medición 2 (Nodo 850), la estación de medición 7 (Nodo 888 en Baja tensión), la estación de medición 8 (Nodo 834).

Para este segundo análisis de los resultados de clasificación de los algoritmos, se agregó como información adicional la distancia entre el nodo en el cual se realiza la medición de tensión y corriente y el nodo donde se efectúa la falla. Este nuevo dato es mostrado en los siguientes gráficos como una línea continua superpuesta con los histogramas de resultados. El objetivo es el de analizar cómo afecta la distancia entre la falla y la medición de dicha falla a lo largo de la red y obtener conclusiones acerca de la propagación del evento de tensión.

5.2.1 Estación de medición 1: Nodo 800

Se presentan los resultados de los Algoritmos *ASA* y *eASA* para las fallas monofásicas, bifásicas con y sin contacto a tierra.

5.2.1.1 Fallas monofásicas

Para las fallas monofásicas se observa una tendencia variable en los resultados del algoritmo *ASA*. Inicialmente, el algoritmo clasificó erróneamente el 60% de la muestra como Tipo E, es decir como un evento provocado por una falla bifásica. Como se comentó previamente, este error aparece porque el algoritmo descarta los eventos provocados por fallas monofásicas debido a que la $V_{\text{mínima}}$ es una tensión de línea sobre raíz de tres y no una tensión de fase. Este error disminuye, pero se mantiene presente hasta el nodo 810 inclusive, que se ubica a unos 20 km de la medición. A partir de este punto, comienza a clasificar las fallas como eventos tipo B en la casi totalidad de los casos. Finalmente, a partir del nodo 830 que se encuentra a unos 43 km del punto medido, empieza a clasificar a la falla como tipo D.

Por el contrario, el algoritmo *eASA* detecta a la falla como un evento del tipo D en la totalidad de los nodos salvo las excepciones antes mencionadas en los nodos 810, 856 y 838. En estos nodos se simulaban fallas en la fase B, a diferencia del resto de las fallas monofásicas que fueron simuladas en la fase A.

Se observó que estas fallas, además de afectar la magnitud de la fase fallada, tenían una disminución en la magnitud de la fase C, motivo por el cual el algoritmo determinaba como evento más probable el de tipo C (bifásica).

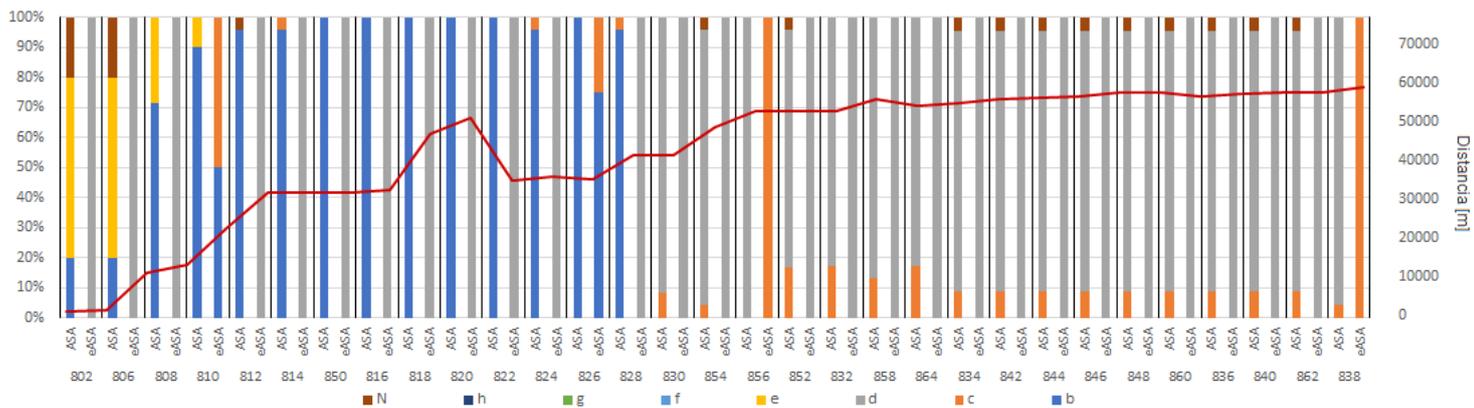


Figura 5-33. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 800 para fallas monofásicas. La línea roja indica la distancia entre la falla y el lugar de medición.

Se concluye que el algoritmo ASA muestra un comportamiento que varía según la distancia entre la medición y la falla, teniendo en cuenta que al hablar de distancias se hace referencia implícitamente a la impedancia de la red. En base a lo analizado en las secciones previas, donde se analizaron los fasores y las componentes simétricas de casos particulares, el algoritmo ASA no detecta en la totalidad de los casos los corrimientos de las fases, clasificando a un evento como Tipo B cuando la firma del evento de tensión es típica de un Tipo D.

Por el contrario, el algoritmo eASA sí detecta los corrimientos en las fases mencionados previamente. Aunque, también, es cierto que clasificó de forma incorrecta las fallas provocadas en los nodos 810, 856 y 838.

En general, el algoritmo eASA tuvo un mejor desempeño en la clasificación de los eventos provocados por fallas monofásicas.

5.2.1.2 Fallas bifásicas sin contacto a tierra

Nuevamente se observa que el algoritmo ASA es más sensible a la ubicación de la falla. En los primeros nodos clasifica a este evento como tipo H debido a que la $V_{m\acute{a}xima}$ resulta ser mayor a 1.1 [p.u], razón por la cual descarta a todos los eventos a excepción del tipo H e I. Al incrementarse la distancia a la falla, comienza a clasificar correctamente al evento como tipo C hasta el nodo 828 ubicado a unos 35 km. A partir de este punto clasifica al evento como Tipo D, es decir, un evento del tipo monofásico.

Como se analizó previamente, este error es cometido porque la $V_{m\acute{i}nima}$ pertenece a una tensión de fase, descartando a todos aquellos eventos que no sean generados por una falla monofásica. Si se analiza al algoritmo eASA, las clasificaciones en la totalidad de los nodos resultaron del tipo C, no presentando errores en la clasificación.

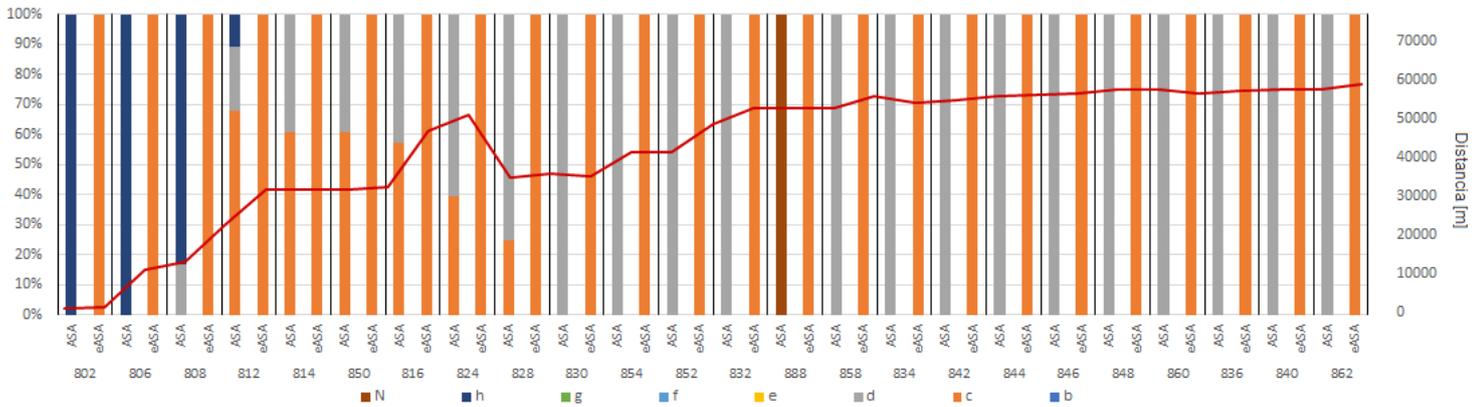


Figura 5-34. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 800 para fallas bifásicas sin contacto a tierra. La línea roja indica la distancia entre la falla y el lugar de medición.

Puede concluirse que la falla principal en el algoritmo ASA es provocado por no evaluar a la totalidad de los eventos luego de realizar un primer descarte basado en la $V_{\text{mínima}}$ y $V_{\text{máxima}}$.

Por el contrario, el algoritmo eASA sí evalúa la totalidad de los eventos y selecciona aquel con menor error cuadrático medio, con la diferencia de que no trabaja con los módulos de las componentes simétricas, sino que tiene en cuenta su signo.

Se concluye que el algoritmo eASA mostró mejor desempeño que el algoritmo ASA al clasificar los eventos generados por las fallas bifásicas con contacto a tierra.

5.2.1.3 Fallas bifásicas con contacto a tierra

Para este tipo de fallas el algoritmo ASA presenta diferencias notables según el lugar de la falla en la red.

Desde los primeros nodos hasta el nodo 854 ubicado a 41 km de la medición el algoritmo clasifica a la falla de tres maneras: Tipo B, Tipo C, Tipo E y en un pequeño porcentaje como Tipo G. Al igual que para las otras fallas analizadas, la razón de clasificar erróneamente como Tipo B es debido a que la $V_{\text{mínima}}$ pertenece a una tensión de fase, descartando todos los eventos provocados por una falla bifásica. Si bien la clasificación correcta sería un evento tipo E, se observó previamente que en muchos casos existía un corrimiento de las fases afectadas, provocando firmas en la red que pueden pertenecer a un evento Tipo C e incluso tipo G. A partir del nodo 854 la clasificación predominante es la Tipo B y también aparece la Tipo D, es decir, el error se incrementa.

De forma similar, desde el inicio de la red hasta el nodo 854 el algoritmo eASA clasifica correctamente, predominando el evento Tipo E y en menor medida el evento Tipo C. A partir del nodo 852 el algoritmo clasifica al 70% de las fallas como eventos Tipo G.

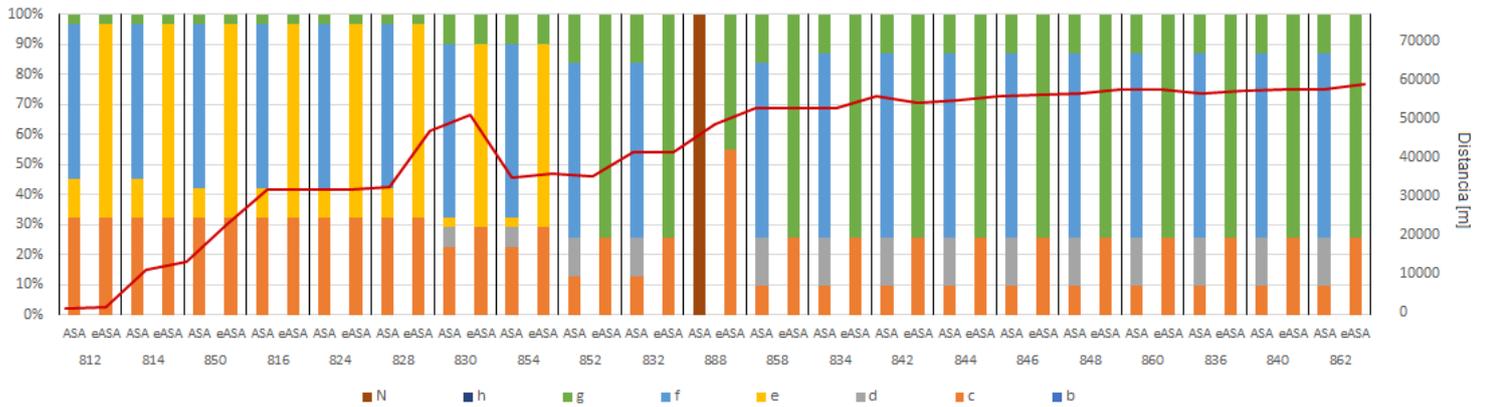


Figura 5-35. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 800 para fallas bifásicas con contacto a tierra. La línea roja indica la distancia entre la falla y el lugar de medición.

Se concluye que el algoritmo eASA mostró un mejor desempeño al clasificar los eventos generados por las fallas bifásicas sin contacto a tierra.

5.2.2 Estación de medición 2: Nodo 850

Se presentan los resultados de los Algoritmos ASA y eASA al clasificar los eventos generados por las fallas monofásicas, bifásicas con y sin contacto a tierra medidos desde el nodo 850 de la red. Se decidió mostrar este caso porque la estación de medición 2 se encuentra en la mitad de la red, a unos 30 km del nodo 800 donde se ubica la alimentación de la red. Con esto se buscará analizar el comportamiento de los algoritmos cuando clasifican eventos de tensión que se produjeron tanto aguas arriba de donde se ubica la estación de medición como aguas abajo.

5.2.2.1 Fallas monofásicas

De igual manera que para la estación de medición 1, el algoritmo eASA clasifica como evento tipo D a la totalidad de las fallas con la excepción de las fallas en los nodos 810, 826, 856 y 838. Como se comentó previamente, este comportamiento aparece en las fallas monofásicas generadas en la fase B de la red. Para el caso del nodo 810, el error aumentó considerablemente si se lo compara con la estación de medición 1. Lo cual implica que el error se incrementó al clasificar un evento que sucedió aguas arriba al punto de medición.

El algoritmo ASA mantiene el comportamiento mostrado en la estación de medición 1, clasificando a los eventos como un tipo B y variando a un tipo D al final de la línea, que coincide con el nodo 838 ubicado a unos 27 km del medidor. Los errores en los nodos al inicio de la red son los mismos que para la estación de medición 1.

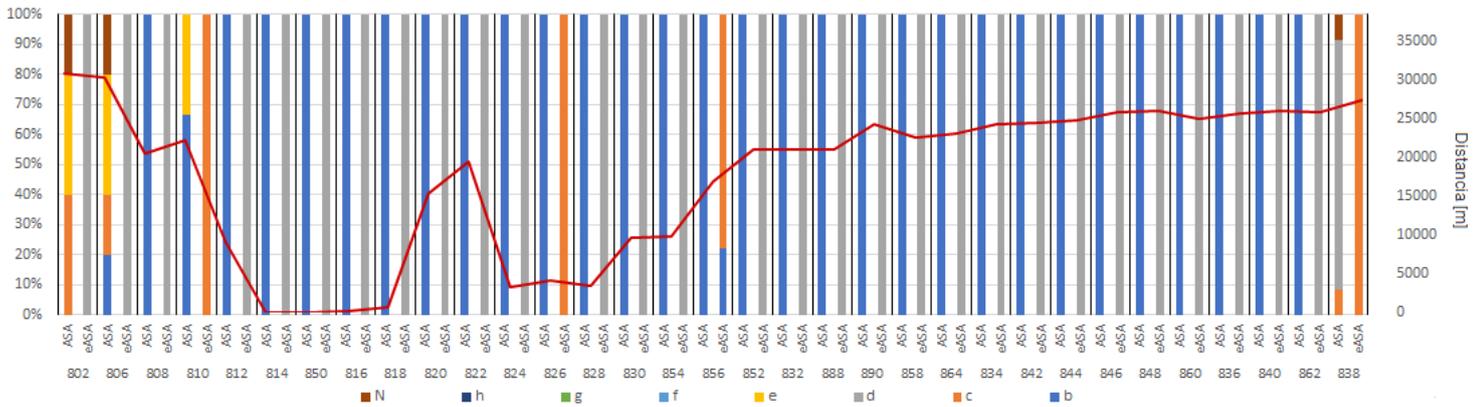


Figura 5-36. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 850 para fallas monofásicas. La línea roja indica la distancia entre la falla y el lugar de medición.

Se concluye que el algoritmo ASA tiene menor sensibilidad a los corrimientos de fase clasificando a la totalidad de los eventos como Tipo B, a excepción del último nodo de la red que lo clasifica como Tipo D. Un comportamiento similar se apreció en la estación 1, donde el algoritmo ASA clasifica como tipo B y luego a partir del nodo 830 comienza a clasificar como tipo D.

Por el contrario, el algoritmo eASA logra detectar a lo largo de toda la red los desfases de las fases no afectadas, clasificando como Tipo D a estos eventos. En este caso la influencia de la ubicación de la falla, es decir, si se ubica aguas arriba o abajo del punto de medición, no provoca alteraciones en los resultados obtenidos por el algoritmo.

Al igual que para la estación de medición 1, los errores de clasificación del algoritmo eASA para las fallas monofásicas provocadas en la fase B son evidentes.

5.2.2.2 Fallas bifásicas sin contacto a tierra

Para este tipo de fallas el algoritmo ASA presenta algunas diferencias según el lugar de la falla en la red. Desde los primeros nodos hasta el nodo 854 ubicado a 41 km de la medición el algoritmo clasifica principalmente al evento como Tipo D. Este error es producto del valor de $V_{\text{mínima}}$. A partir del nodo 854 la clasificación se divide de manera casi equitativa en eventos de Tipo C y D.

Por el contrario, el algoritmo eASA clasifica correctamente en la totalidad de los nodos, siendo eventos de Tipo C.

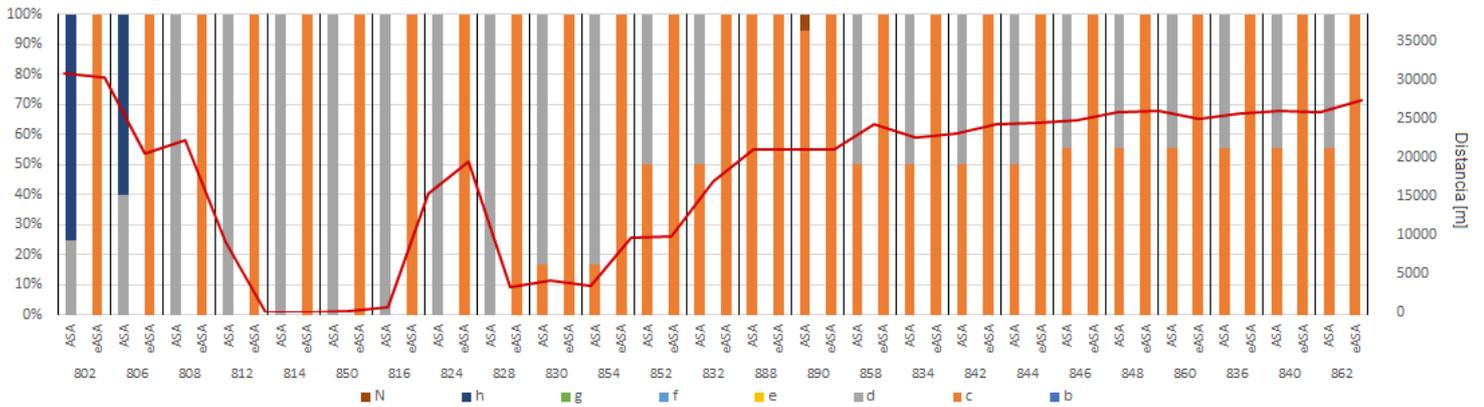


Figura 5-37. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 850 para fallas bifásicas sin contacto a tierra. La línea roja indica la distancia entre la falla y el lugar de medición.

5.2.2.3 Fallas bifásicas con contacto a tierra

Para esta estación de medición no se pudieron analizar las fallas bifásicas con contacto a tierra porque generaron huecos de tensión con tensiones residuales inferiores a 0.5 [p.u] en la totalidad de los nodos. Se concluye que la manera de simular la falla de tipo bifásica con contacto a tierra debería modificarse, ya sea aumentando los valores de impedancia de falla a tierra o agregando impedancias en las fases previo a la impedancia de falla a tierra.

5.2.3 Estación de medición 7: Nodo 888

Se decidió analizar esta estación de medición por tratarse de mediciones en baja tensión. Con esto se buscará extraer conclusiones sobre cómo se propagan los eventos de tensión en la red al cambiar los niveles de tensión.

5.2.3.1 Fallas monofásicas

Analizando los resultados del algoritmo ASA se observa que las clasificaciones varían según la posición de la falla. Las fallas provocadas aguas arriba presentan un mayor error, siendo eventos clasificados como tipo C. La razón es la misma que la comentada previamente, es decir producto de los valores de $V_{\text{mínima}}$. Este error se incrementa al aumentar la distancia.

Las fallas ubicadas en la región final de la red presentan menor error de clasificación. También se observa que el algoritmo falla al clasificar los eventos provocados por las fallas en los nodos 810, 858 y 838, igual que como le sucede al algoritmo eASA. Se observa que, a diferencia de las estaciones de medición 1 y 2, el algoritmo ASA no clasifica ningún evento como tipo B a excepción de las fallas generadas en el mismo nivel de tensión, en los nodos 888 y 890.

Por otro lado, el algoritmo eASA clasifica a la totalidad de los eventos como tipo D, a excepción de las clasificaciones erróneas definidas como tipo C, generadas por las fallas en los nodos 810, 856 y 838 que corresponden a las fallas monofásicas en la fase B.

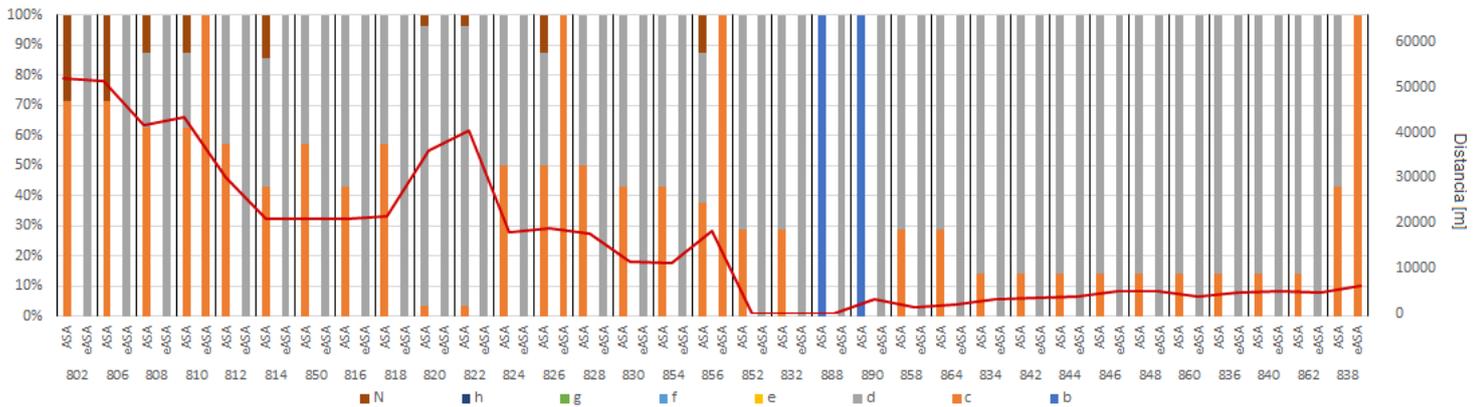


Figura 5-38. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 888 para fallas monofásicas. La línea roja indica la distancia entre la falla y el lugar de medición.

Se concluye que el algoritmo ASA tiene mayor error al clasificar eventos provocados por fallas ubicadas aguas arriba al punto de medición. Al clasificar fallas en otros niveles de tensión, el algoritmo logró detectar los corrimientos de las fases producto de una componente homopolar casi nula, clasificando correctamente a estos eventos como tipo D.

Para mostrar esta diferencia, se adjuntan los gráficos de los fasores y los valores que utiliza el algoritmo para clasificar el mismo evento, medido desde la estación de medición 7 (baja tensión) y la estación de medición 6 (media tensión). También se muestran los valores que utiliza el algoritmo eASA para poder observar la diferencia al clasificar el evento. La falla seleccionada fue la generada en el nodo 860, para una impedancia de falla de 171,5 [Ω]

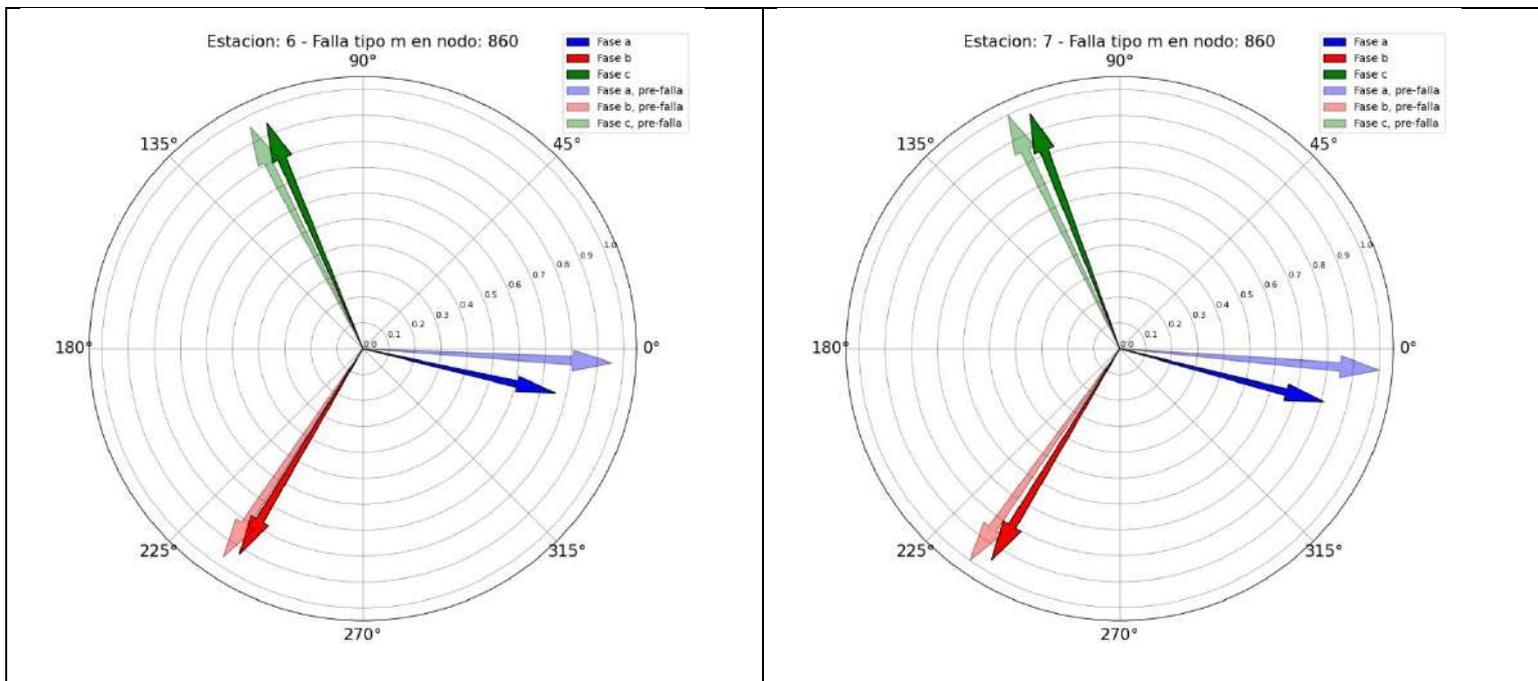


Figura 5-39. Fasores de las tensiones medidas en las estaciones 6 (izquierda) y 7 (derecha) para la falla monofásica en el nodo 860.

Tabla 5-21. Valores utilizados por los algoritmos al clasificar el evento generado por una falla monofásica en el nodo 860.

Estación	Va	Vb	Vc	$\frac{Vab}{\sqrt{3}}$	$\frac{Vbc}{\sqrt{3}}$	$\frac{Vca}{\sqrt{3}}$	V ₀₁	V ₊₁	V ₋₁	V ₀₁ mod	V ₊₁ mod	V ₋₁ mod	Evento	Modelo
6	0,758	0,923	0,947	0,786	0,963	0,878	0,046	0,873	0,102	0,068	0,893	0,068	b	ASA
	0,758	0,923	0,947	0,786	0,963	0,878	-0,046	0,873	-0,102	0,000	0,860	-0,101	d	eASA
7	0,813	0,953	0,970	0,817	0,999	0,916	0,042	0,908	0,105	0,000	0,906	0,093	d	ASA
	0,813	0,953	0,970	0,817	0,999	0,916	-0,042	0,908	-0,105	0,000	0,906	-0,093	d	eASA

Se observa como en la estación de medición 7, ambos algoritmos clasifican al evento como tipo D, mientras que para la estación 6 sólo el algoritmo eASA clasifica como tipo D, y el ASA como tipo B.

Resulta interesante notar que, para esta estación de medición, el algoritmo ASA también falla al clasificar eventos generados por fallas en los nodos 810, 826, 856 y 838, siendo fallas monofásicas en la fase B.

El algoritmo eASA logra detectar a lo largo de toda la red los desfases de las fases no afectadas, clasificando a estos eventos como Tipo D. Al igual que para la estación de medición 1 y 2, los errores de clasificación del algoritmo eASA para las fallas monofásicas provocadas en la fase B son evidentes.

5.2.3.2 Fallas bifásicas sin contacto a tierra

Para esta falla, el algoritmo ASA presenta mucho error en su clasificación. Las fallas aguas arriba a la estación de medición 7 fueron clasificadas en su totalidad como tipo D, es decir evento generado por una falla monofásica. Las clasificaciones de las fallas en la zona final de la red mejoraron hasta aproximadamente un 50% de clasificación correcta.

En cambio, el algoritmo eASA logró clasificar correctamente a la totalidad de los eventos, seleccionando al evento tipo C. El resultado no se vio afectado por la distancia entre la falla y el sitio de medición.

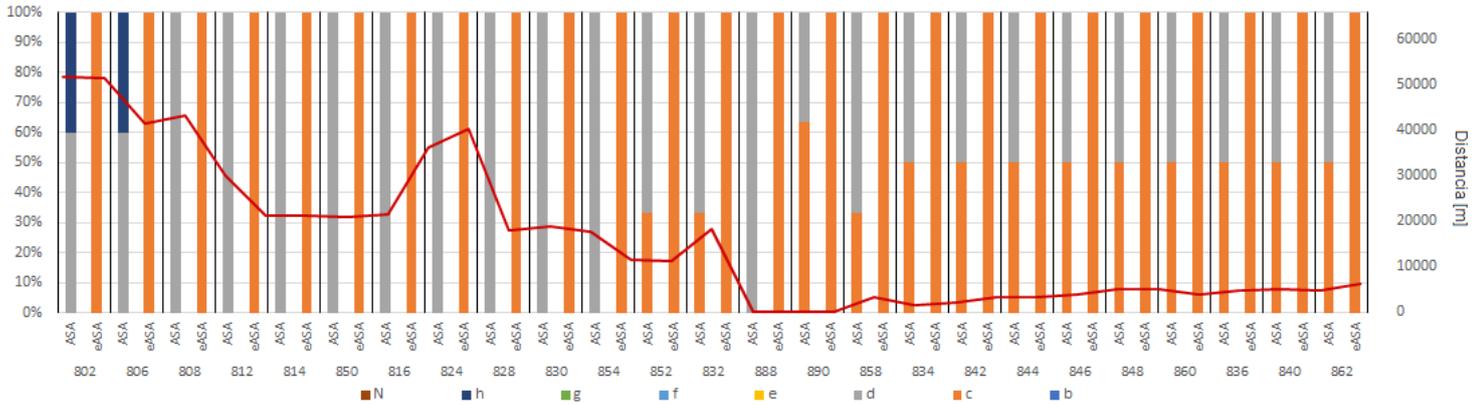


Figura 5-40. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 888 para fallas bifásicas sin contacto a tierra. La línea roja indica la distancia entre la falla y el lugar de medición.

Se concluye que el algoritmo eASA presentó un mejor desempeño al clasificar los eventos provocados por una falla bifásica sin contacto a tierra.

5.2.3.3 Fallas bifásicas con contacto a tierra

Para esta estación de medición no se pudieron analizar las fallas bifásicas con contacto a tierra porque generaron huecos de tensión con tensiones residuales inferiores a 0.5 [p.u.] en la totalidad de los nodos. Se concluye que la manera de simular la falla de tipo bifásica con contacto a tierra debería modificarse, ya sea aumentando los valores de impedancia de falla a tierra o agregando impedancias en las fases previo a la impedancia de falla a tierra.

5.2.4 Estación de medición 8: Nodo 834

Se decidió analizar esta estación de medición por tratarse de mediciones en la zona más alejada al nodo 800, es decir el punto de alimentación de la red. Con esto se buscará extraer conclusiones acerca de cómo se desempeñan los algoritmos al clasificar eventos ocurridos aguas arriba al sitio de medición.

5.2.4.1 Fallas monofásicas

Analizando los resultados del algoritmo ASA, se observa que hasta el nodo 830, unos 3200 [m] desde el sitio de medición, clasificó a los eventos como tipo B. A partir de este punto, empezó a clasificar al evento como Tipo D y las fallas en los nodos del inicio del circuito fueron clasificados como tipo C. Si se compara este comportamiento con el obtenido en la estación de medición 1 para las fallas monofásicas (ver Figura 5-33) se observa que presenta un mismo patrón de clasificación, pero invertido. Esto indica que tanto para fallas aguas arriba como aguas abajo al sitio de medición, la distancia con el lugar de la falla provoca distintas clasificaciones.

Por el contrario, el algoritmo eASA clasificó a la totalidad de las fallas como tipo D, con la excepción antes comentada para los nodos 812, 856 y 838, sin mostrar dependencia con la distancia entre el sitio de medición y la falla.

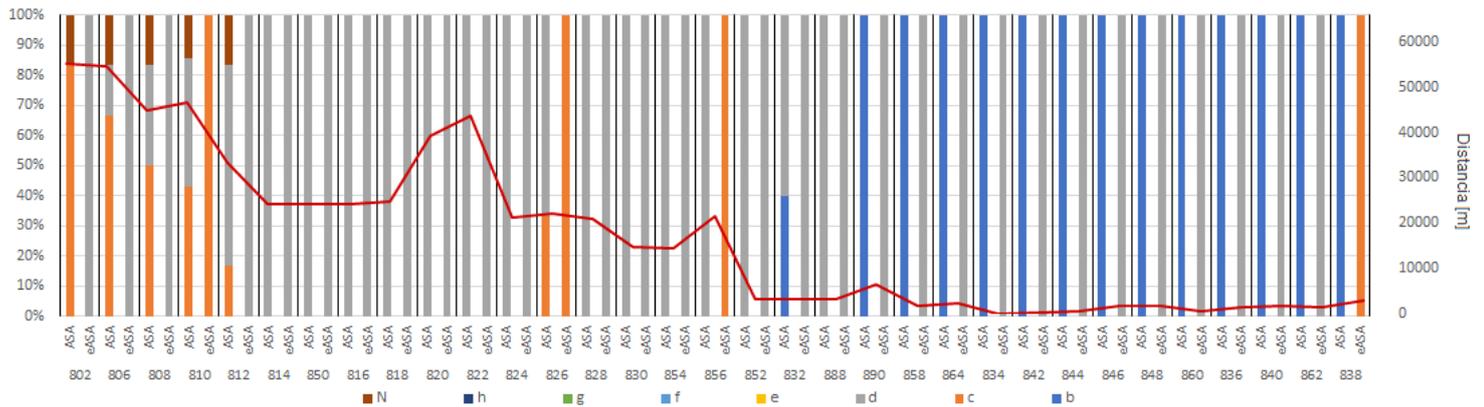


Figura 5-41. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 834 para fallas monofásicas. La línea roja indica la distancia entre la falla y el lugar de medición.

5.2.4.2 Fallas bifásicas sin contacto a tierra

El algoritmo ASA para este tipo de falla muestra una clasificación errónea en un gran porcentaje de los eventos al clasificarlos como tipo D. Este comportamiento se acrecienta al aumentar la distancia entre el sitio de medición y la falla. La excepción aparece al clasificar los eventos generados en baja tensión, nodos 888 y 890. Esto sucede porque las fallas detectadas en estos nodos tienen una $V_{mínima}$ que corresponde a la tensión de línea sobre raíz de tres afectada por la falla (V_{bc}) y por lo tanto el algoritmo descarta que el evento fue provocado por una falla monofásica.

Por el contrario, el algoritmo eASA clasifica correctamente a la totalidad de los eventos, sin importar la distancia entre el sitio de medición y la falla.

El algoritmo que tuvo mejor desempeño en este tipo de fallas fue el eASA.

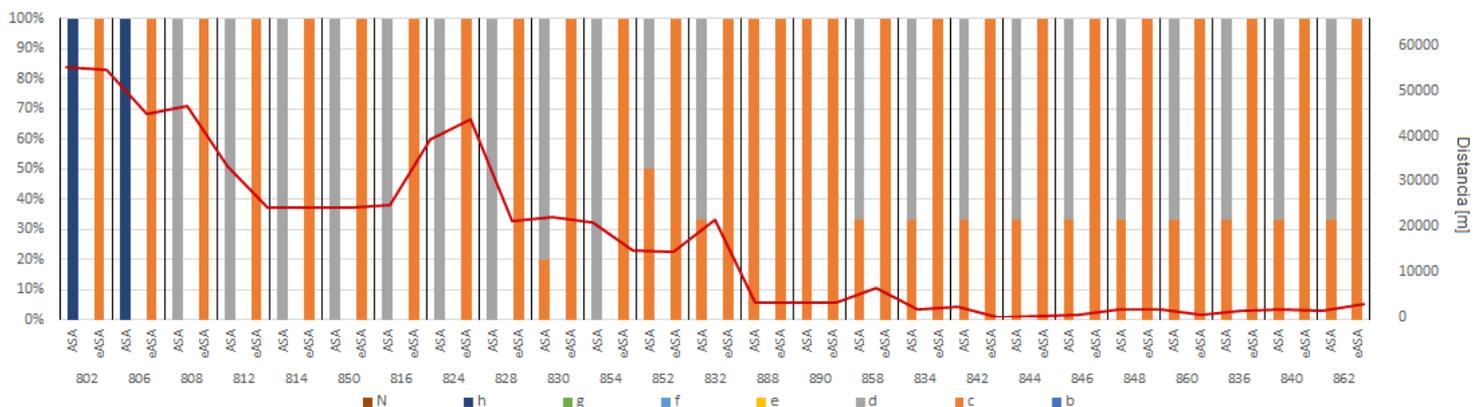


Figura 5-42. Resultados de los algoritmos ASA y eASA de las mediciones en el nodo 834 para fallas bifásicas sin contacto a tierra. La línea roja indica la distancia entre la falla y el lugar de medición.

5.2.4.3 Fallas bifásicas con contacto a tierra

Para esta estación de medición no se pudieron analizar las fallas bifásicas con contacto a tierra porque generaron huecos de tensión con tensiones residuales inferiores a 0.5 [p.u] en la totalidad de los nodos. Se concluye que la manera de simular la falla de tipo bifásica con contacto a tierra debería modificarse, ya sea aumentando los valores de impedancia de falla a tierra o agregando impedancias en las fases previo a la impedancia de falla a tierra.

5.2.5 Desempeño general

Para resumir los resultados obtenidos, se presentan una serie de gráficos de torta. En ellos se evalúan los eventos que fueron detectados correctamente junto con el porcentaje de error de cada algoritmo.

Para las fallas monofásicas se consideraron como eventos detectados correctamente a aquellos de tipo B y D.

Para el caso de las fallas bifásicas sin contacto a tierra se consideró como correctos a aquellos eventos clasificados como Tipo C.

Finalmente, para las fallas bifásicas con contacto a tierra se consideró como correctos a aquellos eventos clasificados como Tipo E y G.

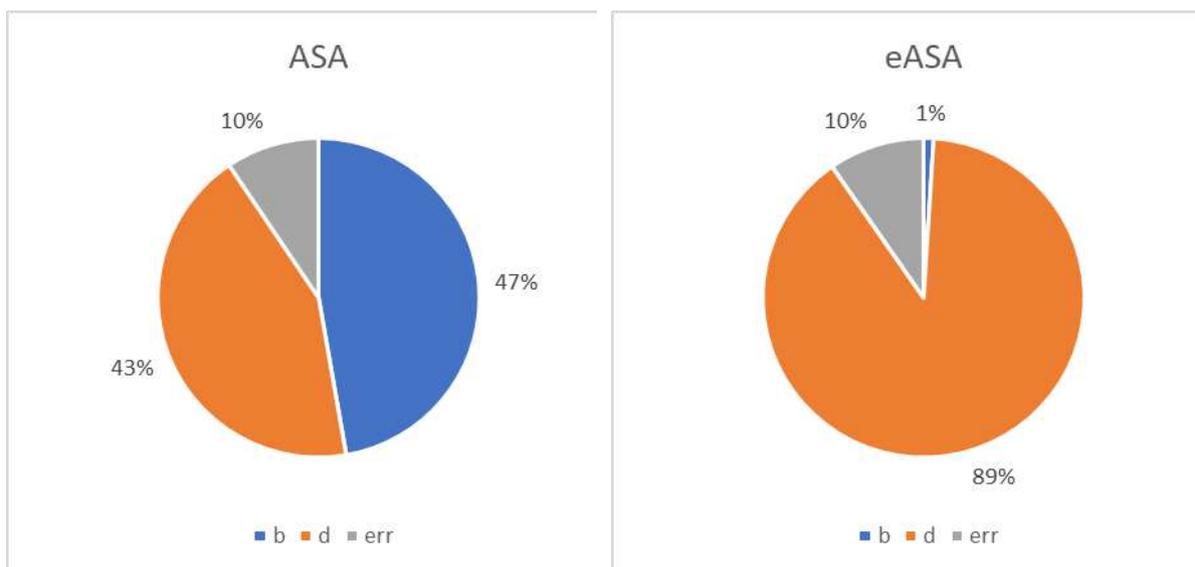


Figura 5-43. Distribución de resultados obtenidos por ambos algoritmos para una falla monofásica.

Se observa que ambos algoritmos presentan el mismo error de clasificación y se diferencian por el tipo de evento detectado. Como se analizó previamente, el evento que mejor representa a las fallas monofásicas fue el tipo D por presentar una componente homopolar de valor muy pequeña y un corrimiento de sus fases, siendo este el comportamiento típico de dicho tipo de evento.

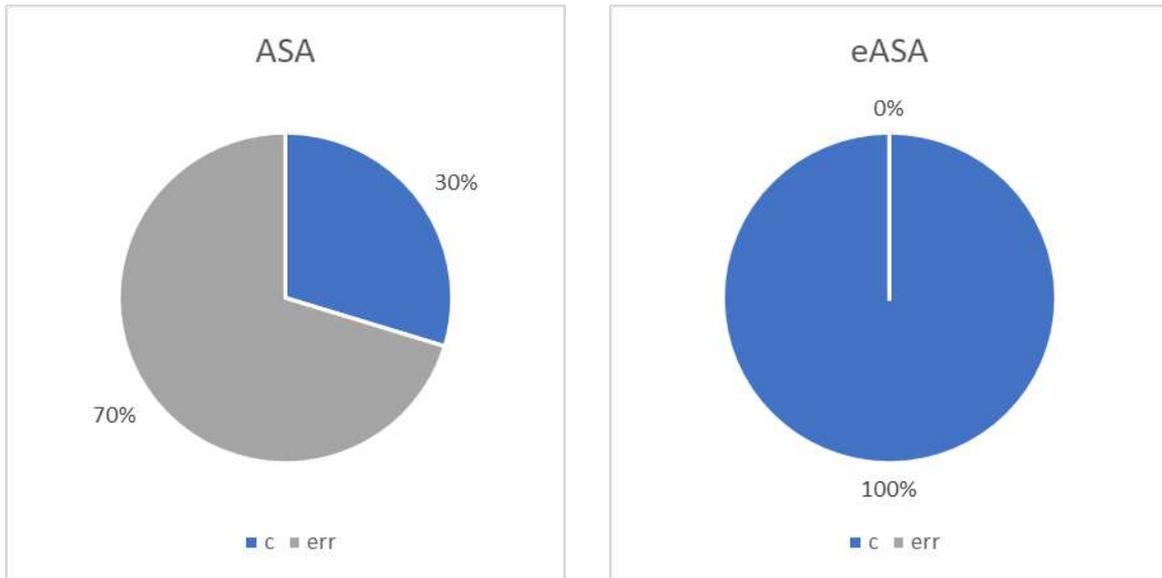


Figura 5-44. Distribución de resultados obtenidos por ambos algoritmos para una falla bifásica sin contacto a tierra.

Se observa que el algoritmo ASA presenta un porcentaje elevado de error. Como se analizó previamente, este error radica en el filtro del evento más probable a partir del análisis de las tensiones mínimas y máximas. Por el contrario, el algoritmo eASA no presentó errores en las clasificaciones realizadas en esta muestra.

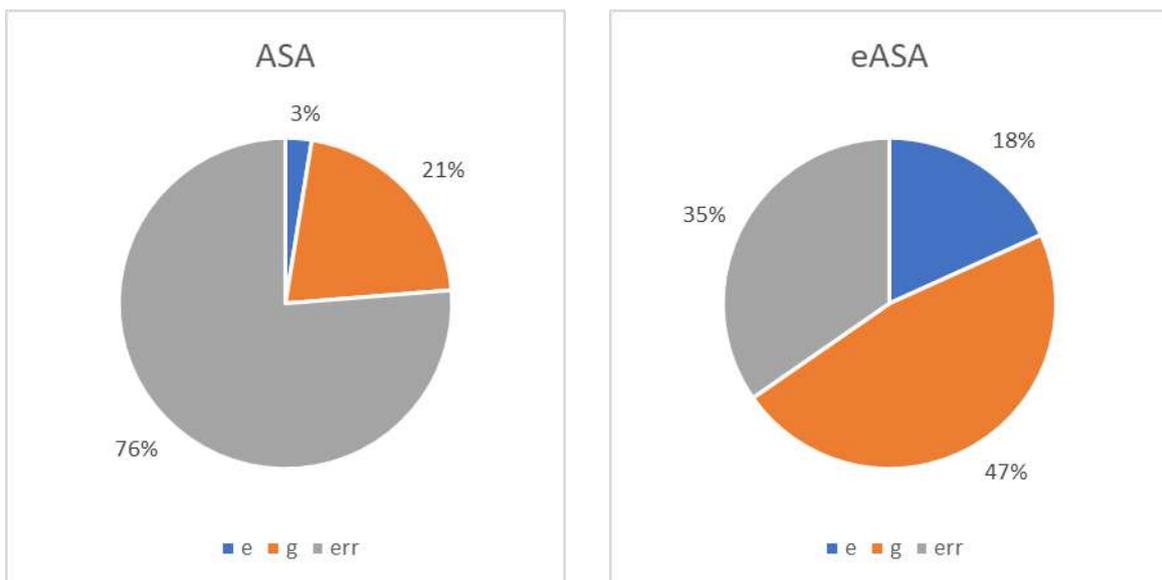


Figura 5-45. Distribución de resultados obtenidos por ambos algoritmos para una falla bifásica con contacto a tierra.

Se observa que el algoritmo ASA es el que presenta mayor error en las clasificaciones. Similar a lo que sucede para las fallas bifásicas sin contacto a tierra, el error principal radica en el filtro del evento más probable luego de analizar las tensiones mínimas y máximas. El algoritmo eASA presenta un error menor, aunque sigue siendo elevado. Como se mencionó previamente, el desempeño de los algoritmos para las fallas bifásicas a tierra se vio afectado

por haberse generado huecos de tensión muy profundos debido a la manera en que se modeló este tipo de falla.

A modo de resumen, se presenta una comparación del recuento de coincidencias entre ambos algoritmos respecto a su clasificación de evento de tensión para una misma falla. Esto refiere a la comparación de eventos sobre una falla en el mismo nodo, misma impedancia de falla y estación de medición.

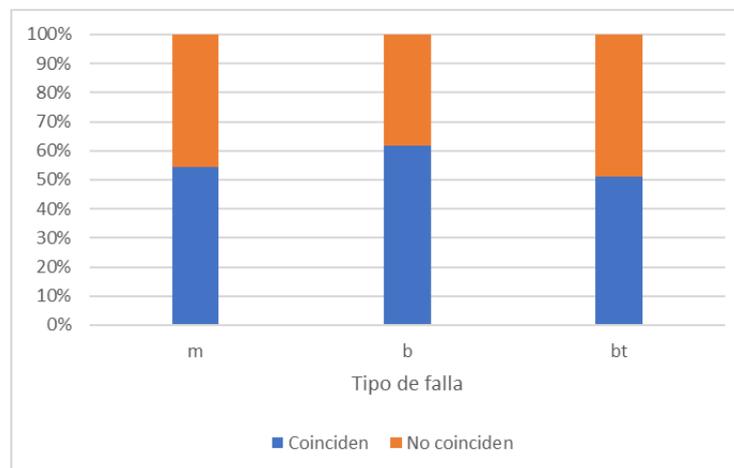


Figura 5-46. Porcentaje de coincidencia entre los resultados obtenidos por cada algoritmo.

En promedio, se observa que el porcentaje de coincidencia entre ambos algoritmos ronda el 55%. Esto denota el impacto que tienen los pequeños cambios efectuados sobre los criterios de clasificación aun cuando ambos algoritmos comparten la misma arquitectura.

5.3 Conclusión

Se cumplió con el objetivo de utilizar la red eléctrica de 34 nodos validada para implementar los algoritmos de clasificación de eventos *ASA* y *eASA* y evaluar su desempeño en fallas generadas en varios sitios de la red.

Se obtuvo una gran cantidad de información de tensiones y corrientes producto de las fallas generadas en cada nodo de la red, lo cual permitió observar las diferencias de estos algoritmos al clasificar eventos de tensión. También pudieron analizarse aquellos casos en donde los algoritmos clasificaban erróneamente al evento, logrando determinar la raíz del problema.

Se observó que el algoritmo *eASA* presentó clasificaciones erróneas al evaluar fallas generadas en la fase B. Esto introduce la idea de que, para este tipo de redes eléctricas desequilibradas, puede ser una buena práctica generar fallas en la totalidad de las fases de manera de evaluar un espectro mayor de firmas generadas por los huecos de tensión.

Se observó, a su vez, que hubo muchas fallas en diferentes nodos de la red que generaron firmas prácticamente idénticas entre sí. Esto se observó principalmente en la zona final de la

red, entre los nodos 858 y 838, donde la distancia entre los mismos no supera los 10 [km]. Este comportamiento era esperado debido a la corta distancia entre los nodos donde fueron generadas las fallas.

Algo similar sucedió con las estaciones de medición. Como se observó en los análisis previos, las estaciones de medición 8, 9 y 10 no aportaron información adicional a los resultados. A raíz de este análisis, se propone considerar a futuro una reducción en la cantidad de sitios de medición definidos en la red.

Por el contrario, las estaciones de medición 1, 2 y 6 sí obtuvieron resultados diferentes entre sí, siendo estaciones que se encuentran alejadas entre ellas: 30 km de distancia entre las estaciones de medición 1 y 2; y 21 km de distancia entre las estaciones 2 y 6.

La estación de medición 7 ubicada en baja tensión aportó información relevante, principalmente en el estudio del comportamiento del algoritmo *ASA* al clasificar eventos provocados por fallas monofásicas en niveles de tensión diferentes. Como se comentó previamente, las fallas bifásicas provocaron huecos de tensión muy profundos, comportamiento que se potenció en este sitio de medición de baja tensión.

Respecto a la manera de simular las fallas, se concluye que el modelo seleccionado de falla bifásica con contacto a tierra no permitió obtener información relevante del comportamiento de la red a causa de los huecos de tensión muy profundos generados, a tal punto que un gran porcentaje de estos datos se encontraban por debajo de 0.1 [p.u] de tensión residual. Se propone, en consecuencia, simular estas fallas incorporando impedancias entre las fases previo a la ubicación de la impedancia de falla a tierra, y utilizar valores de impedancia de falla mayores a 0.7 p.u.

En cuanto al desempeño de los algoritmos, se concluye que el algoritmo *eASA* tuvo un mejor desempeño que el *ASA*, presentando porcentajes de error significativamente inferiores. Se logró evaluar el desempeño frente a saltos de fase, rotaciones de fases simétricas y desviaciones de la tensión de pre-falla respecto de su valor nominal, siendo estos fenómenos algunos de los que suelen introducir errores en la clasificación de los eventos. En ambos algoritmos el desempeño frente a estos fenómenos fue satisfactorio.

Como propuesta a futuro, sería interesante agregar transformadores a la red, por ejemplo, con un conexionado Dyn para generar eventos como el Tipo F. Adicionalmente, podrían incluirse a la simulación una mayor cantidad de líneas de baja tensión para lograr analizar en mayor profundidad el desempeño de estos algoritmos aplicados sobre eventos propagados en otros niveles de tensión. Por último, en vista del auge de la generación distribuida y de la migración de las redes tradicionales a las redes inteligentes, podrían incorporarse generadores distribuidos en la red de manera de evaluar el desempeño de los algoritmos frente a cambios en la topología de la red.

En lo respectivo a los algoritmos considero que sería interesante combinar los algoritmos de detección y clasificación de eventos con algoritmos de localización de fallas con el objetivo de tener disponible mayor información sobre la red como por ejemplo su topología, datos de impedancias, entre otros. Esto podría mejorar de manera significativa los resultados de clasificación de eventos debido a que los algoritmos tendrían mayor información sobre cómo se está propagando el evento a lo largo de la red.

Finalmente, se considera que, a partir del trabajo desarrollado a lo largo del presente proyecto final, se ha demostrado el cumplimiento de los objetivos propuestos con un desempeño satisfactorio.

5.4 Bibliografía

[61] *DPQ Statistical Summary Report*, ser. Id. 1017341. EPRI - Electric Power Research Institute. Dec 31, 2003. Recuperado de: <https://www.epri.com/research/products/1017341>

[62] Jorge L. Strack, Ignacio Carugati, Carlos M. Orallo, Sebastián O. Maestri, Patricio G. Donato, Marcos A. Funes. (2019). *Three-phase voltage events classification algorithm based on an adaptive threshold*. Electric Power Systems Research Volume 172, July 2019, Pages 167-176.

<https://www.sciencedirect.com/science/article/abs/pii/S037877961930104X?via%3Dihub>

[63] Ignacio Carugati, Jorge L. Strack, Patricio G. Donato, Carlos M. Orallo, Marcos A. Funes, Matias Hadad. (2020). *Voltage event classification method based on symmetrical components and extended ABC classification criterion*. Preprint submitted to Electric Power Systems Research April 2, 2020.

[64] Resultados de fallas eléctricas obtenidos en las simulaciones del alimentador de prueba de 34 nodos. https://drive.google.com/drive/folders/1abyF1z-847SfKxje2DR--DnWtNDzxjyE?usp=share_link

6 Anexo

6.1.1 Algoritmo de inicialización

```
“inicialización.py”
import math
#Lectura archivo .atp y creacion de vector con nombres de var
g = open("circuito.atp", "r+") #Este path debe modificarse según el nombre y
ubicación del archivo .atp
atp = g.readlines()
inicio=atp.index("$VINTAGE,1\n")
fin=atp.index("$VINTAGE,0\n")
j=inicio+1
vatp=list()
a2=list()
for i in range(j,fin,1):
    cadena=atp[i]
    n=cadena[2:14]
    if n!="          ":
        a2=n.split()
        vatp.append(a2[0])
    else:
        vatp.append(" ")
print(vatp) #la lista vatp almacena el nombre de la carga junto con los nombres
de los nodos
#lectura de archivo .csv para inicializar ATP
c=open("formato_importacion.csv", "r+")
datos=c.readlines()
m_datos=list()
#m_datos.append(datos[0].split(";")) esto es para leer la primera linea con los
nombres de las columnas
for i in range(1,len(datos),1):
    b=datos[i].split(";") #en caso que el csv separe con coma se tiene que cambiar
";" por ","
    for i in range(5,len(b),1): #arranco en 2 porque el 0 corresponde al nombre
de la carga y el 1 al tipo (m o t)
        if b[i]!="":
            b[i]=float(b[i])
        m_datos.append(b)
#calculo R y L y lo almaceno en una matriz
m_rxl=list()
for i in range(0,len(m_datos),1):
    fila=m_datos[i]
    if fila[3]=="y":
        vf=(fila[11])/math.sqrt(3)
    else:
        vf=fila[11]
```

```

if fila[1]=="m":
    if fila[5]!="":
        cphi=math.cos(math.atan((fila[6]/fila[5])))
        sphi=math.sin(math.atan((fila[6]/fila[5])))
        z=((vf**2)*cphi)/fila[5]
        ri=z*cphi
        xli=z*sphi
        b2=[fila[0],fila[1],round(ri,3),round(xli,3),"", "", "", "", ""]
    elif fila[7]!="":
        cphi=math.cos(math.atan((fila[8]/fila[7])))
        sphi=math.sin(math.atan((fila[8]/fila[7])))
        z=((vf**2)*cphi)/fila[7]
        ri=z*cphi
        xli=z*sphi
        b2=[fila[0],fila[1], "", "", round(ri,3),round(xli,3), "", ""]
    else:
        cphi=math.cos(math.atan((fila[10]/fila[9])))
        sphi=math.sin(math.atan((fila[10]/fila[9])))
        z=((vf**2)*cphi)/fila[9]
        ri=z*cphi
        xli=z*sphi
        b2=[fila[0],fila[1], "", "", "", "", round(ri,3),round(xli,3)]
    m_rxl.append(b2)
elif fila[1]=="t":
    cphi_a=math.cos(math.atan((fila[6]/fila[5])))
    sphi_a=math.sin(math.atan((fila[6]/fila[5])))
    z_a=((vf**2)*cphi_a)/fila[5]
    ri_a=z_a*cphi_a
    xli_a=z_a*sphi_a
    cphi_b=math.cos(math.atan((fila[8]/fila[7])))
    sphi_b=math.sin(math.atan((fila[8]/fila[7])))
    z_b=((vf**2)*cphi_b)/fila[7]
    ri_b=z_b*cphi_b
    xli_b=z_b*sphi_b
    cphi_c=math.cos(math.atan((fila[10]/fila[9])))
    sphi_c=math.sin(math.atan((fila[10]/fila[9])))
    z_c=((vf**2)*cphi_c)/fila[9]
    ri_c=z_c*cphi_c
    xli_c=z_c*sphi_c
    b2=[fila[0],fila[1],round(ri_a,3),round(xli_a,3),round(ri_b,3),round(xli_b,3),round(ri_c,3),round(xli_c,3)]
    m_rxl.append(b2)
else:
    print("El tipo de carga indicado no es correcto. Revisar los datos .csv de la linea ",i+1)
print(m_rxl)
#armo un vector v nuevo con la información de los nuevos valores de R, L, C con el formato de la tarjeta .ATP

```

```

carga=list()
q="      0" #corresponde a la salida de la carga, es la información que va a
aparecer en el archivo .lis. Puede ser 0, 1, 2, 3 y 4. Ver en ATP.
linc="          " #esto corresponde al valor de C, definido vacío por ser
las cargas mayoritariamente de tipo R y L
#ubico primero en el vector vatp la posición del nombre de la carga
for i in range(0,len(m_rxl),1):
    b3=m_rxl[i]
    ind=vatp.index(b3[0])
    print(ind)
    print(b3)
    if b3[1]=="m":
        linea_carga="C "+str(b3[0])+"\n"
        carga.append(linea_carga)
        if len(vatp[ind+1])<12: #armo el formato del nombre del nodo
            espn=12-len(vatp[ind+1])
            linn=vatp[ind+1]
            for i in range(0,espn,1):
                linn=linn+" "
        if b3[2]!="":
            r=b3[2]
            l=b3[3]
        elif b3[4]!="":
            r=b3[4]
            l=b3[5]
        else:
            r=b3[6]
            l=b3[7]
        linr=format(r, '.8E')
        linl=format(l, '.8E')
        espr=16-len(linr)
        espl=16-len(linl)
        for i in range(0,espr,1):
            linr=" "+linr
        for i in range(0,espl,1):
            linl=" "+linl
        lcarga=" "+linn+"          "+linr+linl+linc+q+"\n" #los espacios vacíos
corresponden a ref1 y ref2 que aparecen en la tarjeta pero no se utilizan
        carga.append(lcarga)

    if b3[1]=="t":
        linea_carga="C "+str(b3[0])+"\n"
        carga.append(linea_carga)
        for i in range(1,4,1):
            if len(vatp[ind+i])<12: #armo el formato del nombre del nodo
                espn=12-len(vatp[ind+i])
                linn=vatp[ind+i]
                for j in range(0,espn,1):

```

```

        linn=linn+" "
        r=b3[i*2] #esto es asumiendo que en la tarjeta ATP figura la fase
A, B y C en ese orden
        l=b3[i*2+1] #esta relación es para que a la carga A (i=1) use el valor
de la posición 2 para r y 3 para l. Para B i=2 y para C i=3
        linr=format(r, '.8E')
        linl=format(l, '.8E')
        espr=16-len(linr)
        espl=16-len(linl)
        for i in range(0,espr,1):
            linr=" "+linr
        for i in range(0,espl,1):
            linl=" "+linl

        lcarga=" "+linn+" "+linr+linl+linc+q+"\n" #los espacios
vacíos corresponden a ref1 y ref2 que aparecen en la tarjeta pero no se utilizan
        carga.append(lcarga)
print(carga)

#sobreescribo la información de las cargas con los nuevos valores R, L, C
g.close()
j=inicio+1
h=open("circuito.atp", "w") #lo defino en modo escritura
for i in range(0,inicio+1,1): #el rango del for incluye el valor inicial pero el
final no, es decir, i tomo los valores dentro del conjunto [0,inicio+1). Para
este caso parte desde el valor 0 y llega al valor inicio.
    h.write(atp[i]) #escribo línea por línea hasta que comienza la lista de
cargas (VINTAGE1)
while j<fin: #revisar con el vector de R, L, C. En este caso, las últimas tres
líneas antes de fin corresponden a la misma línea de atp[i]
    try:
        indice=carga.index(atp[j])
        cargai=carga[indice]
        for i in range(0,len(m_datos),1):
            baux=m_datos[i]
            nombre="C "+baux[0)+"\n"
            if nombre==cargai:
                tipo=baux[1]
        if tipo=="m":
            h.write(atp[j]) #escribe primero la línea del comentario con el nombre
de la carga
            h.write(carga[indice+1])
            j=j+2
        else:
            h.write(atp[j])
            h.write(carga[indice+1])
            h.write(carga[indice+2])
            h.write(carga[indice+3])

```

```

        j=j+4
    except ValueError:
        h.write(atp[j])
        j=j+1
for i in range(fin,len(atp),1):
    h.write(atp[i])
h.close()
from subprocess import Popen
p = Popen("inicio.bat", cwd=r"C:\ATP_soft")
stdout, stderr = p.communicate()
print("Proceso de inicialización finalizado")

```

6.1.2 Algoritmo de convergencia

```

“convergencia.py”
import math
import re
import os
#####
##### A P A R T A D O D E F U N C I O N E S #####
#####
def calculo_rxlp(b,matriz_vrms,indice):
    if b[1]=="m":
        v=matriz_vrms[indice+1]
        if b[3]=="y": #la tension de entrada que lee el models es respecto a
tierra, por lo que la tension que devuelve para el caso de conexion en estrella
es igual a Vfase
            vf=v
        else:
            vf=v*math.sqrt(3)
    if b[5]!="":
        cphi=math.cos(math.atan((b[6]/b[5])))
        sphi=math.sin(math.atan((b[6]/b[5])))
        z=((vf**2)*cphi)/b[5]
        rc=z*cphi
        xlc=z*sphi
        resultado=[b[0],b[1],round(rc,3),round(xlc,3),"", "", "", ""]
    elif b[7]!="":
        cphi=math.cos(math.atan((b[8]/b[7])))
        sphi=math.sin(math.atan((b[8]/b[7])))
        z=((vf**2)*cphi)/b[7]
        rc=z*cphi
        xlc=z*sphi
        resultado=[b[0],b[1], "", "", round(rc,3),round(xlc,3), "", ""]
    else:
        cphi=math.cos(math.atan((b[10]/b[9])))
        sphi=math.sin(math.atan((b[10]/b[9])))
        z=((vf**2)*cphi)/b[9]

```

```

        rc=z*cphi
        xlc=z*sphi
        resultado=[b[0],b[1], "", "", "", "", round(rc,3),round(xlc,3)]
elif b[1]=="t":
    if b[3]=="y":
        vfa=matriz_vrms[indice+1]
        vfb=matriz_vrms[indice+2]
        vfc=matriz_vrms[indice+3]
    else:
        vfa=matriz_vrms[indice+1]/math.sqrt(3)
        vfb=matriz_vrms[indice+2]/math.sqrt(3)
        vfc=matriz_vrms[indice+3]/math.sqrt(3)
    cphi_a=math.cos(math.atan((b[6]/b[5])))
    sphi_a=math.sin(math.atan((b[6]/b[5])))
    zc_a=((vfa**2)*cphi_a)/b[5]
    rc_a=zc_a*cphi_a
    xlc_a=zc_a*sphi_a
    cphi_b=math.cos(math.atan((b[8]/b[7])))
    sphi_b=math.sin(math.atan((b[8]/b[7])))
    zc_b=((vfb**2)*cphi_b)/b[7]
    rc_b=zc_b*cphi_b
    xlc_b=zc_b*sphi_b
    cphi_c=math.cos(math.atan((b[10]/b[9])))
    sphi_c=math.sin(math.atan((b[10]/b[9])))
    zc_c=((vfc**2)*cphi_c)/b[9]
    rc_c=zc_c*cphi_c
    xlc_c=zc_c*sphi_c
    resultado=[b[0],b[1],round(rc_a,3),round(xlc_a,3),round(rc_b,3),round(xlc_b,3),round(rc_c,3),round(xlc_c,3)]
    return(resultado)
def calculo_rxli(b,matriz_vrms,indice):
    if b[1]=="m":
        v=matriz_vrms[indice+1]
        if b[3]=="y": #la tension de entrada que lee el modelo es respecto a tierra, por lo que la tension que devuelve para el caso de conexion en estrella es igual a Vfase
            vf=v
            vnominal=b[11]/math.sqrt(3)
        else:
            vf=v*math.sqrt(3)
            vnominal=b[11]
        if b[5]!="":
            cphi=math.cos(math.atan((b[6]/b[5])))
            sphi=math.sin(math.atan((b[6]/b[5])))
            s=math.sqrt((b[5]**2)+(b[6]**2))
            ifase=s/vnominal
            z=vf/ifase
            rc=z*cphi

```

```

        xlc=z*sphi
        resultado=[b[0],b[1],round(rc,3),round(xlc,3),"", "", "", "", ""]
elif b[7]!="":
    cphi=math.cos(math.atan((b[8]/b[7])))
    sphi=math.sin(math.atan((b[8]/b[7])))
    s=math.sqrt((b[7]**2)+(b[8]**2))
    ifase=s/vnominal
    z=vf/ifase
    rc=z*cphi
    xlc=z*sphi
    resultado=[b[0],b[1], "", "", round(rc,3),round(xlc,3), "", ""]
else:
    cphi=math.cos(math.atan((b[10]/b[9])))
    sphi=math.sin(math.atan((b[10]/b[9])))
    s=math.sqrt((b[9]**2)+(b[10]**2))
    ifase=s/vnominal
    z=vf/ifase
    rc=z*cphi
    xlc=z*sphi
    resultado=[b[0],b[1], "", "", "", "", round(rc,3),round(xlc,3)]
elif b[1]=="t":
    if b[3]=="y":
        vfa=matriz_vrms[indice+1]
        vfb=matriz_vrms[indice+2]
        vfc=matriz_vrms[indice+3]
        vnominal=b[11]/math.sqrt(3)
    else:
        vfa=matriz_vrms[indice+1]*math.sqrt(3)
        vfb=matriz_vrms[indice+2]*math.sqrt(3)
        vfc=matriz_vrms[indice+3]*math.sqrt(3)
        vnominal=b[11]
    cphi_a=math.cos(math.atan((b[6]/b[5])))
    sphi_a=math.sin(math.atan((b[6]/b[5])))
    sa=math.sqrt((b[5]**2)+(b[6]**2))
    ifasea=sa/vnominal
    zc_a=vfa/ifasea
    rc_a=zc_a*cphi_a
    xlc_a=zc_a*sphi_a
    cphi_b=math.cos(math.atan((b[8]/b[7])))
    sphi_b=math.sin(math.atan((b[8]/b[7])))
    sb=math.sqrt((b[7]**2)+(b[8]**2))
    ifaseb=sb/vnominal
    zc_b=vfb/ifaseb
    rc_b=zc_b*cphi_b
    xlc_b=zc_b*sphi_b
    cphi_c=math.cos(math.atan((b[10]/b[9])))
    sphi_c=math.sin(math.atan((b[10]/b[9])))
    sc=math.sqrt((b[9]**2)+(b[10]**2))

```

```

    ifasec=sc/vnominal
    zc_c=vfc/ifasec
    rc_c=zc_c*cphi_c
    xlc_c=zc_c*sphi_c
    resultado=[b[0],b[1],round(rc_a,3),round(xlc_a,3),round(rc_b,3),round(xl
c_b,3),round(rc_c,3),round(xlc_c,3)]
    return(resultado)
def datos_tarjeta(b3,vatp,ind):
    carga=list()
    q="    0" #corresponde a la salida de la carga, es la info que va a
aparecer en el .lis. Puede ser 0, 1, 2, 3 y 4. Ver en ATP.
    linc="                " #esto corresponde al valor de C, definido vacio por
ser las cargas mayoritariamente de tipo R,L
    if b3[1]=="m":
        linea_carga="C "+str(b3[0])+"\n"
        carga.append(linea_carga)
        if len(vatp[ind+1])<12: #armo el formato del nombre del nodo
            espn=12-len(vatp[ind+1])
            linn=vatp[ind+1]
            for i in range(0,espn,1):
                linn=linn+" "
    if b3[2]!="":
        r=b3[2]
        l=b3[3]
    elif b3[4]!="":
        r=b3[4]
        l=b3[5]
    else:
        r=b3[6]
        l=b3[7]
    linr=format(r, '.8E')
    linl=format(l, '.8E')
    espr=16-len(linr)
    espl=16-len(linl)
    for i in range(0,espr,1):
        linr=" "+linr
    for i in range(0,espl,1):
        linl=" "+linl
    lcarga=" "+linn+"                "+linr+linl+linc+q+"\n" #los espacios
vacios corresponden a ref1 y ref2 que aparecen en la tarjeta pero no se utilizan
    carga.append(lcarga)

    if b3[1]=="t":
        linea_carga="C "+str(b3[0])+"\n"
        carga.append(linea_carga)
        for i in range(1,4,1):
            if len(vatp[ind+i])<12: #armo el formato del nombre del nodo
                espn=12-len(vatp[ind+i])

```

```

        linn=vatp[ind+i]
        for j in range(0,espn,1):
            linn=linn+" "
            r=b3[i*2] #esto es asumiendo que en la tarjeta ATP figura la fase
A, B y C en ese orden
            l=b3[i*2+1] #esta relacion es para que a la carga A (i=1) use el
valor de la posicion 2 para r y 3 para l. Para B i=2 y para C i=3
            linr=format(r, '.8E')
            linl=format(l, '.8E')
            espr=16-len(linr)
            espl=16-len(linl)
            for i in range(0,espr,1):
                linr=" "+linr
            for i in range(0,espl,1):
                linl=" "+linl

            lcarga=" "+linn+" "+linr+linl+linc+q+"\n" #los espacios
vacios corresponden a ref1 y ref2 que aparecen en la tarjeta pero no se utilizan
            carga.append(lcarga)
        return(carga)
#####
##### A P A R T A D O D E C A L C U L O S #####
#####

#lectura de archivo .csv para obtener los datos de P, Q y Vnominal
c=open("formato_importacion.csv", "r+")
datos=c.readlines()
m_datos=list()
#m_datos.append(datos[0].split(";")) esto es para leer la primera linea con los
nombres de las columnas
for i in range(1,len(datos),1):
    b=datos[i].split(";") #en caso que el csv separe con coma se tiene que
cambiar ";" por ","
    for i in range(5,len(b)-1,1): #arranco en 2 porque el 0 corresponde al
nombre de la carga y el 1 al tipo (m o t)
        if b[i]!="":
            b[i]=float(b[i])
    m_datos.append(b)
#####
#Lectura archivo .lis y construccion de vectores de variables
#####
# abro el archivo .lis para leer los valores de tension
f = open("circuitoprueba7.lis", "r+")
# cada linea del archivo .lis se almacena en un elemento de la lista l
l = f.readlines()

#recorro las lineas de .lis buscando los comentarios de los nombres de los nodos
y junto con los resultados del models de Vrms

```

```

#creo un vector vrms con los nombres de los nodos y sus valores de tension rms
vrms=list()
for i in range(0,len(m_datos),1):
    aux=m_datos[i]
    if aux[12]=="s\n":
        nodo=aux[0]
        aux1=l.index("V "+nodo+"\n")
        if aux[1]=="m":
            vrms.append(l[aux1])
            avrms=l[aux1+1]
            avrms=re.sub("\n", "", avrms)
            vrms.append(float(avrms))
        if aux[1]=="t":
            vrms.append(l[aux1])
            avrms=l[aux1+1]
            avrms=re.sub("\n", "", avrms)
            vrms.append(float(avrms))
            bvrms=l[aux1+2]
            bvrms=re.sub("\n", "", bvrms)
            vrms.append(float(bvrms))
            cvrms=l[aux1+3]
            cvrms=re.sub("\n", "", cvrms)
            vrms.append(float(cvrms))

print(vrms)
#####
#Lectura archivo .atp y creacion de vector con nombres de var
#####
g = open("circuitoprueba7.atp", "r+")
atp = g.readlines()
inicio=atp.index("$VINTAGE,1\n") #ver si conviene poner un comentario con
inicio/fin
fin=atp.index("$VINTAGE,0\n")
j=inicio+1
vatp=list()
a2=list()
for i in range(j,fin,1):
    cadena=atp[i]
    n=cadena[2:14]
    if n!=" ":
        a2=n.split()
        vatp.append(a2[0])
    else:
        vatp.append(" ")
print(vatp) #la lista vatp me almacena el nombre de la carga junto con los
nombres de los nodos
#####
#Calculo de R y L a partir de P e I - C O N V E R G E N C I A
#####

```

```

m_conv=list()
for i in range (0,len(m_datos),1):
    b4=m_datos[i]
    if b4[12]=="s\n":
        indvar=vrms.index("V "+b4[0)+"\n") #criterio: "V "+nodo
        if b4[4]=="p": #estos calculos son validos para el caso de una carga de
            tipo Potencia constante
                m_conv.append(calculo_rxlp(b4,vrms,indvar))
        elif b4[4]=="i":
            m_conv.append(calculo_rxli(b4,vrms,indvar))
print(m_conv)
#armo un vector tarjeta con la informacion de los nuevos valores de R, L, C con
el formato de la tarjeta .ATP
tarjeta=list()
#ubico primero en el vector vatp la posicion del nombre de la carga
for i in range(0,len(m_conv),1):
    btarjeta=m_conv[i]
    indtarjeta=vatp.index(btarjeta[0])
    tarjeta=tarjeta+datos_tarjeta(btarjeta,vatp,indtarjeta)
print(tarjeta)
#sobreescribo la informacion de las cargas con los nuevos valores R, L, C
g.close()
j=inicio+1
h=open("circuitoprueba7.atp", "w") #lo defino en modo escritura
for i in range(0,inicio+1,1): #el rango del for incluye el valor inicial pero el
final no, es decir, i tomo los valores dentro del conjunto [0,inicio+1). Para
este caso parte desde el valor 0 y llega al valor inicio.
    h.write(atp[i]) #escribo linea por linea hasta que comienza la lista de
cargas (VINTAGE1)
while j<fin: #revisar con el vector de R,L,C. En este caso las ultimas tres
lineas antes de fin corresponden a la misma linea de atp[i]
    try:
        indice=tarjeta.index(atp[j])
        carga=tarjeta[indice]
        #tengo que agregar una linea que me busque el nombre de la carga en la
matriz m_datos y me devuelva si es m o t
        for i in range(0,len(m_datos),1):
            baux=m_datos[i]
            nombre="C "+baux[0)+"\n"
            if nombre==carga:
                tipo=baux[1]
        if tipo=="m":
            h.write(atp[j]) #me escribe primero la linea del comentario con el
nombre de la carga
            h.write(tarjeta[indice+1])
            j=j+2
        else:
            h.write(atp[j])

```

```

        h.write(tarjeta[indice+1])
        h.write(tarjeta[indice+2])
        h.write(tarjeta[indice+3])
        j=j+4
    except ValueError:
        h.write(atp[j])
        j=j+1
for i in range(fin,len(atp),1):
    h.write(atp[i])
h.close()
from subprocess import Popen
p = Popen("inicio.bat", cwd=r"C:\ATP_soft")
stdout, stderr = p.communicate()
#Almaceno los valores de vrms, r y l en un .txt
convergencia=list()
for i in range(0, len(m_conv),1):
    aux2=m_conv[i]
    if aux2[1]=="m":
        aux3=vrms.index("V "+b4[0]+"\\n")
        if aux2[2]!="":
            aux4=aux2[0]+";"+aux2[1]+";"+str(aux2[2])+";"+str(aux2[3])+";"+aux2[
4]+";"+aux2[5]+";"+aux2[6]+";"+aux2[7]+";"+str(vrms[aux3+1])+";"+";"+";"+\\n"
        elif aux2[4]!="":
            aux4=aux2[0]+";"+aux2[1]+";"+aux2[2]+";"+aux2[3]+";"+str(aux2[4])+";
"+str(aux2[5])+";"+aux2[6]+";"+aux2[7]+";"+";"+";"+";"+str(vrms[aux3+1])+";"+";"+\\n"
        else:
            aux4=aux2[0]+";"+aux2[1]+";"+aux2[2]+";"+aux2[3]+";"+aux2[4]+";"+aux
2[5]+";"+str(aux2[6])+";"+str(aux2[7])+";"+";"+";"+";"+str(vrms[aux3+1])+";"+\\n"
            convergencia.append(aux4)
    if aux2[1]=="t":
        aux3=vrms.index("V "+b4[0]+"\\n")
        aux4=aux2[0]+";"+aux2[1]+";"+str(aux2[2])+";"+str(aux2[3])+";"+str(aux2[
4])+";"+str(aux2[5])+";"+str(aux2[6])+";"+str(aux2[7])+";"+str(vrms[aux3+1])+";"
+str(vrms[aux3+2])+";"+str(vrms[aux3+3])+";"+\\n"
        convergencia.append(aux4)
print(convergencia)
#Almaceno los valores de R, Xl y Vrms con el formato:
id_carga;tipo;ra;sla;rb;slb;rc;slc;vrmsa;vrmsb;vrmsc
k=open("resultados_convergencia.txt", "a+")
k.write("\\n")
for i in range(0,len(convergencia),1):
    k.write(convergencia[i])
k.close()
linvrms=list()
for i in range(0,len(m_datos),1):
    aux=m_datos[i]
    nodo=aux[0]
    aux1=l.index("V "+nodo+"\\n")

```

```

if aux[1]=="m":
    avrms=l[aux1+1]
    avrms=re.sub("\n", "", avrms)
    if aux[5]!="":
        lvrms=aux[0]+";"+str(avrms)+";"+";"+"\n"
    elif aux[7]!="":
        lvrms=aux[0]+";"+";"+";"+str(avrms)+";"+";"+"\n"
    else:
        lvrms=aux[0]+";"+";"+";"+";"+str(avrms)+"\n"
    linvrms.append(lvrms)
if aux[1]=="t":
    vrms.append(l[aux1])
    avrms=l[aux1+1]
    avrms=re.sub("\n", "", avrms)
    bvrms=l[aux1+2]
    bvrms=re.sub("\n", "", bvrms)
    cvrms=l[aux1+3]
    cvrms=re.sub("\n", "", cvrms)
    lvrms=aux[0]+";"+str(avrms)+";"+str(bvrms)+";"+str(cvrms)+"\n"
    linvrms.append(lvrms)
y=open("valores_vrms.txt", "a+")
y.write("\n")
for i in range(0,len(linvrms),1):
    y.write(linvrms[i])
y.close()
print("Proceso de convergencia finalizado")

```

6.1.3 Algoritmo de fallas

6.1.3.1 Modificación de valor de resistencia

```

"fallas.py"
#el objetivo es modificar los valores de la resistencia de falla, llamar al .atp
para simular
#primero leo el valor anterior de resistencia desde la tarjeta .atp
h=open("C:/SimulacionesATP/FALLAS/simulacion/falla.atp", "r")
hl=h.readlines()
h.close()
ind=hl.index("C r_falla\n") #busco donde se encuentra el comentario que
corresponde a la resistencia
vres=hl[ind+1].split() #a la linea de valores de la resistencia la divido por
espacios
res_ant=float(vres[1]) #almaceno el valor de resistencia anterior
res=res_ant*1.1 #aumento un 5% el valor de la resistencia respecto al valor
anterior
res="{:.5f}".format(res)
linr=str(res)

```

```

espr=16-len(linr)
if len(vres[0])<12: #armo el formato del nombre del nodo
    espn=12-len(vres[0])
    linn=vres[0]
    for j in range(0,espn,1):
        linn=linn+" "
for i in range(0,espr,1): #armo el formato del valor de r, esto es considerando
que tiene MENOS de 16 caracteres
    linr=" "+linr
lres=" "+linn+"          "+linr+"          "+"          "+"          "
+vres[2]+"\\n"
#sobreescribo la tarjeta .atp reemplazando el valor nuevo de resistencia
k=open("C:/SimulacionesATP/FALLAS/simulacion/falla.atp","w")
for i in range(0,len(hl),1):
    if i!=(ind+1):
        k.write(hl[i])
    else:
        k.write(lres)
k.close()

```

6.1.3.2 Lectura de resultados RMS y almacenamiento

```

"resultados_fallas.py"
#el objetivo es leer los resultados desde el archivo .lis, almacenarlos en un
.csv.
import re
f = open("C:/SimulacionesATP/FALLAS/simulacion/lista_nodos_fallas.csv", "r+")
#MODIFICAR PATH CON EL ARCHIVO FINAL DE LISTA DE NODOS
fl=f.readlines()
m_nodos=list()

for i in range(1,len(fl),1): #creo la matriz con la informacion de los nodos a
los que quiero buscar en el .lis
    aux=fl[i].split(";")
    m_nodos.append(aux)
f.close()
#Leo el valor de resistencia actual
z=open("C:/SimulacionesATP/FALLAS/simulacion/falla.atp", "r+")
zl=z.readlines()
ind=zl.index("C r_falla\\n")
lres=zl[ind+1].split()
res=lres[1]
z.close()
#DEFINO LA CANTIDAD DE TIEMPOS DE MUESTREO
nro_muestreo=3
g = open("C:/SimulacionesATP/FALLAS/simulacion/falla.lis", "r+") #MODIFICAR AL
NOMBRE DE ARCHIVO .LIS FINAL
lis = g.readlines()

```

```

m_resultados=list()
resultado=list()
for i in range(0,nro_muestreo,1):
    n=i+1
    for i in range(0,len(m_nodos),1):
        aux1=m_nodos[i]
        indv=lis.index("RMS "+str(n)+".0          "+aux1[2]+".\n") #encabezado
de la medida Vrms
        indf=lis.index("FASOR V "+str(n)+".0          "+aux1[2]+".\n") #encabezado
de la medida del fasor de V
        indi=lis.index("RMS "+str(n)+".0          "+aux1[0]+".\n") #encabezado de
la medida Irms
        indfi=lis.index("ESTACION DE
MEDICION "+aux1[0]+"          "+str(n)+".0\n") #encabezado de la medida del
fasor de I
        if aux1[1]=="m":
            v=lis[indv+1].split()
            t=re.sub("\n", "", v[1])
            vrms=v[0]
            il=lis[indi+1].split()
            irms=il[0]
            f=lis[indf+1].split()
            fase=f[1]
            fi=lis[indfi+1].split()
            fasei=fi[1]
            if aux1[3]=="a\n":
                resultado=[aux1[0],t,float(vrms), "", "", float(fase), "", "", float(i
rms), "", "", float(fasei), "", ""]
            elif aux1[3]=="b\n":
                resultado=[aux1[0],t, "", float(vrms), "", "", float(fase), "", "", floa
t(irms), "", "", float(fasei), ""]
            else:
                resultado=[aux1[0],t, "", "", float(vrms), "", "", float(fase), "", "", f
loat(irms), "", "", float(fasei)]
        elif aux1[1]=="t":
            va=lis[indv+1].split()
            t=re.sub("\n", "", va[1])
            avrms=va[0]
            vb=lis[indv+2].split()
            bvrms=vb[0]
            vc=lis[indv+3].split()
            cvrms=vc[0]
            fa=lis[indf+1].split()
            fasea=fa[1]
            fb=lis[indf+2].split()
            faseb=fb[1]
            fc=lis[indf+3].split()
            fasec=fc[1]

```

```

        ia=lis[indi+1].split()
        airms=ia[0]
        ib=lis[indi+2].split()
        birms=ib[0]
        ic=lis[indi+3].split()
        cirms=ic[0]
        fia=lis[indfi+1].split()
        faseia=fia[1]
        fib=lis[indfi+2].split()
        faseib=fib[1]
        fic=lis[indfi+3].split()
        faseic=fic[1]
        resultado=[aux1[0],t,float(avrms),float(bvrms),float(cvrms),float(fa
sea),float(faseb),float(fasec),float(airms),float(birms),float(cirms),float(fase
ia),float(faseib),float(faseic)]
    else:
        print("no se ingreso t de trifasica o m de monofasica")
    m_resultados.append(resultado)

#####
#Almacenamiento de resultados de vrms y fases en archivo .txt
#####
h=open("C:/SimulacionesATP/FALLAS/simulacion/resultados_fallas.csv", "r+")
hl=h.readlines()
h.close()
encabezado="estacion;resistencia;tiempo;vrms a;vrms b;vrms c;fase a;fase b;fase
c;irms a;irms b;irms c;fasei a;fasei b;fasei c\n"
y=open("C:/SimulacionesATP/FALLAS/simulacion/resultados_fallas.csv", "w")
if hl[0]!=encabezado:
    y.write(encabezado)
else:
    for i in range(0,len(hl),1):
        y.write(hl[i])
j=0
for i in range(0,len(m_resultados),1):
    a=m_resultados[i]
    if j<len(m_nodos):
        b=m_nodos[j]
        j=j+1
    else:
        j=1
        b=m_nodos[0]
    if b[1]=="m":
        if b[3]=="a\n":
            linea=a[0]+";"+res+";"+a[1]+";"+str(a[2])+";"+""+";"+""+";"+str(a[5]
)+";"+""+";"+""+";"+str(a[8])+";"+""+";"+""+";"+str(a[11])+";"+""+";"+""+""\n"
        elif b[3]=="b\n":

```

```

        linea=a[0]+";"+res+";"+a[1]+";"+""+";"+str(a[3])+";"+""+";"+""+";"+s
tr(a[6])+";"+""+";"+""+";"+str(a[9])+";"+""+";"+""+";"+str(a[12])+";"+""+""\n"
    else:
        linea=a[0]+";"+res+";"+a[1]+";"+""+";"+""+";"+str(a[4])+";"+""+";"+
"+";"+str(a[7])+";"+""+";"+""+";"+str(a[10])+";"+""+";"+""+";"+str(a[13])+""\n"
    else:
        linea=a[0]+";"+res+";"+a[1]+";"+str(a[2])+";"+str(a[3])+";"+str(a[4])+";
"+str(a[5])+";"+str(a[6])+";"+str(a[7])+";"+str(a[8])+";"+str(a[9])+";"+str(a[10
])+";"+str(a[11])+";"+str(a[12])+";"+str(a[13])+""\n"
    y.write(linea)
y.close()

```

6.1.4 Códigos de generación de gráficos

6.1.4.1 Gráfico de magnitud y desfase de tensión RMS vs. impedancia

```

“grafico_xy.py”
import matplotlib
import numpy as np
import matplotlib.pyplot as plt

print("matplotlib.__version__ = ", matplotlib.__version__)
print("matplotlib.get_backend()= ", matplotlib.get_backend())
#Usado para graficar el mismo nodo en distintos tiempos. hasta ahora muestra 2
tiempos
#####
#Para cada estacion corregir SOLO la linea 12
est=1 #INDICAR EL NRO DE ESTACION
k = open("C:/SimulacionesATP/graficos/resultados_fallas.csv", "r+")
lk=k.readlines()
a=lk[est].split(";") #almaceno los valores pre-falla
k.close()
#ALMACENO LAS FASES ANTES DE LA FALLA
a[6]=float(a[6])
a[7]=float(a[7])
a[8]=float(a[8])
if a[6]!="":
    if a[7]!="":
        if a[8]!="":
            if a[6]<0:
                fap=360+float(a[6])
            else:
                fap=float(a[6])
            if a[7]<0:
                fbp=360+float(a[7])
            else:
                fbp=float(a[7])

```

```

        if a[8]<0:
            fcp=360+float(a[8])
        else:
            fcp=float(a[8])
    else:
        if a[6]<0:
            fap=360+float(a[6])
        else:
            fap=float(a[6])
else:
    if a[7]!="":
        if a[7]<0:
            fbp=360+float(a[7])
        else:
            fbp=float(a[7])
    else:
        if a[8]<0:
            fcp=360+float(a[8])
        else:
            fcp=float(a[8])
##### MONOFASICA #####
titulo1="Vf y argVf vs Z estacion "+str(est)+" m"
h = open("C:/SimulacionesATP/graficos/m.csv","r+")
rl=h.readlines()
i=est+10
va=list()
vb=list()
vc=list()
fa=list()
fb=list()
fc=list()
z=list()
while i<len(rl):
    b=rl[i].split(";")
    b[6]=float(b[6])
    b[7]=float(b[7])
    b[8]=float(b[8])
    if a[3]!="":
        if a[4]!="":
            if a[5]!="":
                va.append(float(b[3]))
                vb.append(float(b[4]))
                vc.append(float(b[5]))
                if b[6]<0:
                    fa2=360+float(b[6])
                    dfa=fa2-fap
                else:
                    dfa=float(b[6])-fap

```

```

        if b[7]<0:
            fb2=360+float(b[7])
            dfb=fb2-fbp
        else:
            dfb=float(b[7])-fbp
        if b[8]<0:
            fc2=360+float(b[8])
            dfc=fc2-fcp
        else:
            dfc=float(b[8])-fcp
        fa.append(dfa)
        fb.append(dfb)
        fc.append(dfc)
    else:
        va.append(float(b[3]))
        if b[6]<0:
            fa2=360+float(b[6])
            dfa=fa2-fap
        else:
            dfa=float(b[6])-fap
        fa.append(dfa)
else:
    if a[4]!="":
        vb.append(float(b[4]))
        if b[7]<0:
            fb2=360+float(b[7])
            dfb=fb2-fbp
        else:
            dfb=float(b[7])-fbp
        fb.append(dfb)
    else:
        vc.append(float(b[5]))
        if b[8]<0:
            fc2=360+float(b[8])
            dfc=fc2-fcp
        else:
            dfc=float(b[8])-fcp
        fc.append(dfc)
res=float(b[1])/248.0 #248 es el valor base de la impedancia resistiva
z.append(res)
i=i+20
figure, axis = plt.subplots(2)
if a[3]!="":
    axis[0].plot(z, va, label = "Fase a", color='tab:blue')
if a[4]!="":
    axis[0].plot(z, vb, label = "Fase b", color='tab:red')
if a[5]!="":
    axis[0].plot(z, vc, label = "Fase c", color='tab:green')

```

```

axis[0].set(ylabel='Vf
[pu]',xlim=[0,1.35],xticks=[0,0.15,0.30,0.45,0.60,0.75,0.9,1.05,1.20,1.35],
          yticks=[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2])
axis[0].legend(loc='lower right')
axis[0].grid(True, linestyle='-.', lw=0.4, color='grey')
if a[3]!="":
    axis[1].plot(z, fa, label = "Fase a", color='tab:blue')

if a[4]!="":
    axis[1].plot(z, fb, label = "Fase b", color='tab:red')

if a[5]!="":
    axis[1].plot(z, fc, label = "Fase c", color='tab:green')
axis[1].set(xlabel='Zf [pu]', ylabel='arg(Vf)
[°]',xlim=[0,1.35],xticks=[0,0.15,0.30,0.45,0.60,0.75,0.9,1.05,1.20,1.35])
axis[1].legend(loc='lower right')
axis[1].grid(True, linestyle='-.', lw=0.4, color='grey')
plt.savefig(titulo1)
plt.clf()
plt.close()
##### BIFASICA #####
titulo2="Vf y argVf vs Z estacion "+str(est)+" b"
h = open("C:/SimulacionesATP/graficos/b.csv","r+")
rl=h.readlines()
i=est+10
va=list()
vb=list()
vc=list()
fa=list()
fb=list()
fc=list()
z=list()
while i<len(rl):
    b=rl[i].split(";")
    b[6]=float(b[6])
    b[7]=float(b[7])
    b[8]=float(b[8])
    if a[3]!="":
        if a[4]!="":
            if a[5]!="":
                va.append(float(b[3]))
                vb.append(float(b[4]))
                vc.append(float(b[5]))
                if b[6]<0:
                    fa2=360+float(b[6])
                    dfa=fa2-fap
                else:
                    dfa=float(b[6])-fap

```

```

        if b[7]<0:
            fb2=360+float(b[7])
            dfb=fb2-fbp
        else:
            dfb=float(b[7])-fbp
        if b[8]<0:
            fc2=360+float(b[8])
            dfc=fc2-fcp
        else:
            dfc=float(b[8])-fcp
        fa.append(dfa)
        fb.append(dfb)
        fc.append(dfc)
    else:
        va.append(float(b[3]))
        if b[6]<0:
            fa2=360+float(b[6])
            dfa=fa2-fap
        else:
            dfa=float(b[6])-fap
        fa.append(dfa)
else:
    if a[4]!="":
        vb.append(float(b[4]))
        if b[7]<0:
            fb2=360+float(b[7])
            dfb=fb2-fbp
        else:
            dfb=float(b[7])-fbp
        fb.append(dfb)
    else:
        vc.append(float(b[5]))
        if b[8]<0:
            fc2=360+float(b[8])
            dfc=fc2-fcp
        else:
            dfc=float(b[8])-fcp
        fc.append(dfc)
res=float(b[1])/248.0 #248 es el valor base de la impedancia resistiva
z.append(res)
i=i+20
figure, axis = plt.subplots(2)
if a[3]!="":
    axis[0].plot(z, va, label = "Fase a", color='tab:blue')

if a[4]!="":
    axis[0].plot(z, vb, label = "Fase b", color='tab:red')

```

```

if a[5]!="":
    axis[0].plot(z, vc, label = "Fase c", color='tab:green')
axis[0].set(ylabel='Vf
[pu]',xlim=[0,1.35],xticks=[0,0.15,0.30,0.45,0.60,0.75,0.9,1.05,1.20,1.35],
    yticks=[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2])
axis[0].legend(loc='lower right')
axis[0].grid(True, linestyle='-.', lw=0.4, color='grey')
if a[3]!="":
    axis[1].plot(z, fa, label = "Fase a", color='tab:blue')
if a[4]!="":
    axis[1].plot(z, fb, label = "Fase b", color='tab:red')
if a[5]!="":
    axis[1].plot(z, fc, label = "Fase c", color='tab:green')
axis[1].set(xlabel='Zf [pu]', ylabel='arg(Vf)
[°]',xlim=[0,1.35],xticks=[0,0.15,0.30,0.45,0.60,0.75,0.9,1.05,1.20,1.35])
axis[1].legend(loc='lower right')
axis[1].grid(True, linestyle='-.', lw=0.4, color='grey')
plt.savefig(titulo2)
plt.clf()
plt.close()
##### BIFASICA CON PUESTA A TIERRA #####
titulo3="Vf y argVf vs Z estacion "+str(est)+" bt"
h = open("C:/SimulacionesATP/graficos/bt.csv", "r+")
rl=h.readlines()
i=est+10
va=list()
vb=list()
vc=list()
fa=list()
fb=list()
fc=list()
z=list()
while i<len(rl):
    b=rl[i].split(";")
    b[6]=float(b[6])
    b[7]=float(b[7])
    b[8]=float(b[8])
    if a[3]!="":
        if a[4]!="":
            if a[5]!="":
                va.append(float(b[3]))
                vb.append(float(b[4]))
                vc.append(float(b[5]))
                if b[6]<0:
                    fa2=360+float(b[6])
                    dfa=fa2-fap
                else:
                    dfa=float(b[6])-fap

```

```

        if b[7]<0:
            fb2=360+float(b[7])
            dfb=fb2-fbp
        else:
            dfb=float(b[7])-fbp
        if b[8]<0:
            fc2=360+float(b[8])
            dfc=fc2-fcp
        else:
            dfc=float(b[8])-fcp
        fa.append(dfa)
        fb.append(dfb)
        fc.append(dfc)
    else:
        va.append(float(b[3]))
        if b[6]<0:
            fa2=360+float(b[6])
            dfa=fa2-fap
        else:
            dfa=float(b[6])-fap
        fa.append(dfa)
else:
    if a[4]!="":
        vb.append(float(b[4]))
        if b[7]<0:
            fb2=360+float(b[7])
            dfb=fb2-fbp
        else:
            dfb=float(b[7])-fbp
        fb.append(dfb)
    else:
        vc.append(float(b[5]))
        if b[8]<0:
            fc2=360+float(b[8])
            dfc=fc2-fcp
        else:
            dfc=float(b[8])-fcp
        fc.append(dfc)
res=float(b[1])/248.0 #248 es el valor base de la impedancia resistiva
z.append(res)
i=i+20
figure, axis = plt.subplots(2)
if a[3]!="":
    axis[0].plot(z, va, label = "Fase a", color='tab:blue')
if a[4]!="":
    axis[0].plot(z, vb, label = "Fase b", color='tab:red')
if a[5]!="":
    axis[0].plot(z, vc, label = "Fase c", color='tab:green')

```

```

axis[0].set(ylabel='Vf
[pu]',xlim=[0,1.35],xticks=[0,0.15,0.30,0.45,0.60,0.75,0.9,1.05,1.20,1.35],
          yticks=[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2])
axis[0].legend(loc='lower right')
axis[0].grid(True, linestyle='-.', lw=0.4, color='grey')
if a[3]!="":
    axis[1].plot(z, fa, label = "Fase a", color='tab:blue')
if a[4]!="":
    axis[1].plot(z, fb, label = "Fase b", color='tab:red')
if a[5]!="":
    axis[1].plot(z, fc, label = "Fase c", color='tab:green')
axis[1].set(xlabel='Zf [pu]', ylabel='arg(Vf)
[°]',xlim=[0,1.35],xticks=[0,0.15,0.30,0.45,0.60,0.75,0.9,1.05,1.20,1.35])
axis[1].legend(loc='lower right')
axis[1].grid(True, linestyle='-.', lw=0.4, color='grey')
plt.savefig(titulo3)
plt.clf()
plt.close()
print("Los graficos fueron almacenados en la carpeta")

```

6.1.4.2 Gráfico fasorial

```

"grafico_fasores.py"
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
print("matplotlib.__version__= ", matplotlib.__version__)
print("matplotlib.get_backend()= ", matplotlib.get_backend())
#Usado para graficar el mismo nodo en distintos tiempos. hasta ahora muestra 2
tiempos
#####
est=1 #INDICAR EL NRO DE ESTACION
ref=(-33.904469)/180.*np.pi #FASE DE REFERENCIA
k = open("C:/SimulacionesATP/graficos/resultados_fallas.csv", "r+")
lk=k.readlines()
a=lk[est].split(";") #almaceno los valores pre-falla
k.close()
##### MONOFASICA #####
h = open("C:/SimulacionesATP/graficos/m.csv", "r+")
rl=h.readlines()
b=rl[est+10].split(";") #solo grafico en estado estacionario
titulo1="Resultados estación "+str(est)+" en estado estacionario"
titulo2=" Resultados estacion "+str(est)+" estado estacionario m"
# radar green, solid grid lines
plt.rc('grid', color='grey', linewidth=0.5, linestyle='-') #'#316931'
plt.rc('xtick', labelsz=15) #tamaño de los numeros de los angulos
plt.rc('ytick', labelsz=8) #tamaño de los numeros de los circulos de las
divisiones 0.25, 0.5, 0.75 etc

```

```

# force square figure and square axes looks better for polar, IMO
width, height = matplotlib.rcParams['figure.figsize']
size = min(width, height)+4
# make a square figure
fig = plt.figure(figsize=(size, size))
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8], polar=True)
plt.yticks([0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2])
ax.plot(0,1,color='#ee8d18', lw=0.4)
ax.set_rmax(2.0)
plt.grid(True)
ax.set_title(titulo1, fontsize=15)

if a[3]!="":
    if a[4]!="":
        if a[5]!="":
            va=float(a[3])
            fa=float(a[6])
            vb=float(a[4])
            fb=float(a[7])
            vc=float(a[5])
            fc=float(a[8])
            va1=float(b[3])
            fa1=float(b[6])
            vb1=float(b[4])
            fb1=float(b[7])
            vc1=float(b[5])
            fc1=float(b[8])
            arr1 = plt.arrow((fa/180.*np.pi)-ref, 0, 0.0, va, alpha
= 0.5, width = 0.04,
                                length_includes_head=True , head_width=0.1 ,
            edgecolor = 'black',facecolor = 'blue',lw = 1, zorder = 5, label ="Fase a, pre-
falla") #desde el inicio, 0, 0.5 (angulo, inicio del modulo )es el inicio del
vector, y
                # el prox 0, 1 es lo que se le suma al inicio. entonces el
vector quedaría derecho desde 0 a 0 grados, y va desde 0.5 a (0.5+1) 1.5 de
largo
                # zorder es el nivel de la linea o en este caso flecha que se
esté graficando. significa que si hay 5 lineas superpuestas, la de mas arriba
sera la de zorder 5
                # arrow at 45 degree
            arr2 = plt.arrow((fb/180.*np.pi)-ref, 0, 0.0, vb, alpha
= 0.5, width = 0.04,
                                length_includes_head=True , head_width=0.1 ,
            edgecolor = 'black', facecolor = 'red', lw = 1, zorder = 5, label ="Fase b, pre-
falla")

            arr3 = plt.arrow((fc/180.*np.pi)-ref, 0, 0.0, vc, alpha
= 0.5, width = 0.04,

```

```

        length_includes_head=True , head_width=0.1 ,
edgecolor = 'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c,
pre-falla")

        arr4 = plt.arrow((fa1/180.*np.pi)-ref, 0, 0, va1, alpha
= 1, width = 0.04,
        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black',facecolor = 'blue',lw = 1, zorder = 5, label ="Fase a, post-
falla") #desde el inicio, 0, 0.5 (angulo, inicio del modulo )es el inicio del
vector, y
        # el prox 0, 1 es lo que se le suma al inicio. entonces el
vector quedaría derecho desde 0 a 0 grados, y va desde 0.5 a (0.5+1) 1.5 de
largo
        # zorder es el nivel de la linea o en este caso flecha que se
esté graficando. significa que si hay 5 lineas superpuestas, la de mas arriba
sera la de zorder 5
        # arrow at 45 degree
        arr5 = plt.arrow((fb1/180.*np.pi)-ref, 0, 0, vb1, alpha
= 1, width = 0.04,
        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black', facecolor = 'red', lw = 1, zorder = 5, label ="Fase b,
post-falla")

        arr6 = plt.arrow((fc1/180.*np.pi)-ref, 0, 0, vc1, alpha
= 1, width = 0.04,
        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c,
post-falla")

    else:
        va=float(a[3])
        fa=float(a[6])
        va1=float(b[3])
        fa1=float(b[6])
        arr1 = plt.arrow((fa/180.*np.pi)-ref, 0, 0, va, alpha = 0.5,
width = 0.04,
        length_includes_head=True , head_width=0.1 , edgecolor =
'black',facecolor = 'blue',lw = 1, zorder = 5, label="Fase a")
        arr4 = plt.arrow((fa1/180.*np.pi)-ref, 0, 0, va1, alpha = 1,
width = 0.04,
        length_includes_head=True , head_width=0.05 , edgecolor
= 'black',facecolor = 'blue',lw = 1, zorder = 5, label="Fase a, r0")

elif a[4]!="":
    vb=float(a[4])
    fb=float(a[7])
    vb1=float(b[4])
    fb1=float(b[7])

```

```

arr2 = plt.arrow((fb/180.*np.pi)-ref, 0, 0, vb, alpha = 0.5, width =
0.04,
                length_includes_head=True , head_width=0.1 , edgecolor =
'black', facecolor = 'red', lw = 1, zorder = 5, label="Fase b")

arr5 = plt.arrow((fb1/180.*np.pi)-ref, 0, 0, vb1, alpha = 1, width =
0.04,
                length_includes_head=True , head_width=0.05 , edgecolor =
'black', facecolor = 'red', lw = 1, zorder = 5, label="Fase b, r0")

else:
    vc=float(a[5])
    fc=float(a[8])
    vc1=float(b[5])
    fc1=float(b[8])
    arr3 = plt.arrow((fc/180.*np.pi)-ref, 0, 0, vc, alpha = 0.5, width =
0.04,
                    length_includes_head=True , head_width=0.1 , edgecolor =
'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c")

    arr6 = plt.arrow((fc1/180.*np.pi)-ref, 0, 0, vc1, alpha = 1, width =
0.04,
                    length_includes_head=True , head_width=0.05 , edgecolor =
'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c, r0")

i=est+10
while i<len(r1): #va de la primera linea del resultado de la estacion para el t2
(estado estacionario) saltando 20 filas
    aux=r1[i].split(";")
    d=1-(0.05*(i/100))
    if aux[3]!="":
        fa=float(aux[6])
        va=float(aux[3])
        c = plt.scatter((fa/180.*np.pi)-ref, va,s=2,color='blue',
alpha=d)
    if aux[4]!="":
        fb=float(aux[7])
        vb=float(aux[4])
        c = plt.scatter((fb/180.*np.pi)-ref, vb,s=2,color='red',
alpha=d)
    if aux[5]!="":
        fc=float(aux[8])
        vc=float(aux[5])
        c = plt.scatter((fc/180.*np.pi)-ref, vc,s=2,color='green',
alpha=d)
    i=i+20

```

```

angle = np.deg2rad(30)
ax.legend(loc='upper right', bbox_to_anchor=(0.67 + np.cos(angle)/2, 0.84 +
np.sin(angle)/2))
plt.savefig(titulo2)
plt.clf()
plt.close()
##### BIFASICA #####
h = open("C:/SimulacionesATP/graficos/b.csv", "r+")
rl=h.readlines()
b=rl[est+10].split(";") #solo grafico en estado estacionario
titulo3="Resultados estación "+str(est)+" en estado estacionario"
titulo4=" Resultados estacion "+str(est)+" estado estacionario b"
# radar green, solid grid lines
plt.rc('grid', color='grey', linewidth=0.5, linestyle='-') #'#316931'
plt.rc('xtick', labels=15) #tamaño de los numeros de los angulos
plt.rc('ytick', labels=8) #tamaño de los numeros de los circulos de las
divisiones 0.25, 0.5, 0.75 etc
# force square figure and square axes looks better for polar, IMO
width, height = matplotlib.rcParams['figure.figsize']
size = min(width, height)+4
# make a square figure
fig = plt.figure(figsize=(size, size))
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8], polar=True)
plt.yticks([0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2])
ax.plot(0,1,color='#ee8d18', lw=0.4)
ax.set_rmax(2.0)
plt.grid(True)
ax.set_title(titulo3, fontsize=15)

if a[3]!="":
    if a[4]!="":
        if a[5]!="":
            va=float(a[3])
            fa=float(a[6])
            vb=float(a[4])
            fb=float(a[7])
            vc=float(a[5])
            fc=float(a[8])
            va1=float(b[3])
            fa1=float(b[6])
            vb1=float(b[4])
            fb1=float(b[7])
            vc1=float(b[5])
            fc1=float(b[8])
            arr1 = plt.arrow((fa/180.*np.pi)-ref, 0, 0.0, va, alpha
= 0.5, width = 0.04,
                                length_includes_head=True , head_width=0.1 ,
edgecolor = 'black',facecolor = 'blue',lw = 1, zorder = 5, label ="Fase a, pre-

```

```

falla") #desde el inico, 0, 0.5 (angulo, inicio del modulo )es el inicio del
vector, y
        # el prox 0, 1 es lo que se le suma al inicio. entonces el
vector quedaría derecho desde 0 a 0 grados, y va desde 0.5 a (0.5+1) 1.5 de
largo
        # zorder es el nivel de la linea o en este caso flecha que se
esté graficando. significa que si hay 5 lineas superpuestas, la de mas arriba
sera la de zorder 5
        # arrow at 45 degree
        arr2 = plt.arrow((fb/180.*np.pi)-ref, 0, 0.0, vb, alpha
= 0.5, width = 0.04,
                        length_includes_head=True , head_width=0.1 ,
edgecolor = 'black', facecolor = 'red', lw = 1, zorder = 5, label ="Fase b, pre-
falla")

        arr3 = plt.arrow((fc/180.*np.pi)-ref, 0, 0.0, vc, alpha
= 0.5, width = 0.04,
                        length_includes_head=True , head_width=0.1 ,
edgecolor = 'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c,
pre-falla")

        arr4 = plt.arrow((fa1/180.*np.pi)-ref, 0, 0, va1, alpha
= 1, width = 0.04,
                        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black',facecolor = 'blue',lw = 1, zorder = 5, label ="Fase a, post-
falla") #desde el inico, 0, 0.5 (angulo, inicio del modulo )es el inicio del
vector, y
        # el prox 0, 1 es lo que se le suma al inicio. entonces el
vector quedaría derecho desde 0 a 0 grados, y va desde 0.5 a (0.5+1) 1.5 de
largo
        # zorder es el nivel de la linea o en este caso flecha que se
esté graficando. significa que si hay 5 lineas superpuestas, la de mas arriba
sera la de zorder 5
        # arrow at 45 degree
        arr5 = plt.arrow((fb1/180.*np.pi)-ref, 0, 0, vb1, alpha
= 1, width = 0.04,
                        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black', facecolor = 'red', lw = 1, zorder = 5, label ="Fase b,
post-falla")

        arr6 = plt.arrow((fc1/180.*np.pi)-ref, 0, 0, vc1, alpha
= 1, width = 0.04,
                        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c,
post-falla")

    else:
        va=float(a[3])

```

```

        fa=float(a[6])
        va1=float(b[3])
        fa1=float(b[6])
        arr1 = plt.arrow((fa/180.*np.pi)-ref, 0, 0, va, alpha = 0.5,
width = 0.04,
                        length_includes_head=True , head_width=0.1 , edgecolor =
'black',facecolor = 'blue',lw = 1, zorder = 5, label="Fase a")
        arr4 = plt.arrow((fa1/180.*np.pi)-ref, 0, 0, va1, alpha = 1,
width = 0.04,
                        length_includes_head=True , head_width=0.05 , edgecolor
= 'black',facecolor = 'blue',lw = 1, zorder = 5, label="Fase a, r0")
elif a[4]!="":
    vb=float(a[4])
    fb=float(a[7])
    vb1=float(b[4])
    fb1=float(b[7])
    arr2 = plt.arrow((fb/180.*np.pi)-ref, 0, 0, vb, alpha = 0.5, width =
0.04,
                    length_includes_head=True , head_width=0.1 , edgecolor =
'black', facecolor = 'red', lw = 1, zorder = 5, label="Fase b")

    arr5 = plt.arrow((fb1/180.*np.pi)-ref, 0, 0, vb1, alpha = 1, width =
0.04,
                    length_includes_head=True , head_width=0.05 , edgecolor =
'black', facecolor = 'red', lw = 1, zorder = 5, label="Fase b, r0")

else:
    vc=float(a[5])
    fc=float(a[8])
    vc1=float(b[5])
    fc1=float(b[8])
    arr3 = plt.arrow((fc/180.*np.pi)-ref, 0, 0, vc, alpha = 0.5, width =
0.04,
                    length_includes_head=True , head_width=0.1 , edgecolor =
'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c")

    arr6 = plt.arrow((fc1/180.*np.pi)-ref, 0, 0, vc1, alpha = 1, width =
0.04,
                    length_includes_head=True , head_width=0.05 , edgecolor =
'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c, r0")

i=est+10
while i<len(r1): #va de la primera linea del resultado de la estacion para el t2
(estado estacionario) saltando 20 filas
    aux=r1[i].split(";")
    d=1-(0.05*(i/100))

```

```

        if aux[3]!="":
            fa=float(aux[6])
            va=float(aux[3])
            c = plt.scatter((fa/180.*np.pi)-ref, va,s=2,color='blue',
alpha=d)
        if aux[4]!="":
            fb=float(aux[7])
            vb=float(aux[4])
            c = plt.scatter((fb/180.*np.pi)-ref, vb,s=2,color='red',
alpha=d)
        if aux[5]!="":
            fc=float(aux[8])
            vc=float(aux[5])
            c = plt.scatter((fc/180.*np.pi)-ref, vc,s=2,color='green',
alpha=d)
        i=i+20

angle = np.deg2rad(30)
ax.legend(loc='upper right', bbox_to_anchor=(0.67 + np.cos(angle)/2, 0.84 +
np.sin(angle)/2))

plt.savefig(titulo4)
plt.clf()
plt.close()

##### BIFASICA CON PUESTA A TIERRA #####
h = open("C:/SimulacionesATP/graficos/bt.csv", "r+")
rl=h.readlines()
b=rl[est+10].split(";") #solo grafico en estado estacionario
titulo5="Resultados estación "+str(est)+" en estado estacionario"
titulo6=" Resultados estacion "+str(est)+" estado estacionario bt"
        # radar green, solid grid lines
plt.rc('grid', color='grey', linewidth=0.5, linestyle='-')   #'#316931'
plt.rc('xtick', labels=15) #tamaño de los numeros de los angulos
plt.rc('ytick', labels=8) #tamaño de los numeros de los circulos de las
divisiones 0.25, 0.5, 0.75 etc
        # force square figure and square axes looks better for polar, IMO
width, height = matplotlib.rcParams['figure.figsize']
size = min(width, height)+4
        # make a square figure
fig = plt.figure(figsize=(size, size))
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8], polar=True)
plt.yticks([0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2])
ax.plot(0,1,color='#ee8d18', lw=0.4)
ax.set_rmax(2.0)
plt.grid(True)
ax.set_title(titulo5, fontsize=15)
if a[3]!="":

```

```

if a[4]!="":
    if a[5]!="":
        va=float(a[3])
        fa=float(a[6])
        vb=float(a[4])
        fb=float(a[7])
        vc=float(a[5])
        fc=float(a[8])
        va1=float(b[3])
        fa1=float(b[6])
        vb1=float(b[4])
        fb1=float(b[7])
        vc1=float(b[5])
        fc1=float(b[8])
        arr1 = plt.arrow((fa/180.*np.pi)-ref, 0, 0.0, va, alpha
= 0.5, width = 0.04,
                        length_includes_head=True , head_width=0.1 ,
edgecolor = 'black',facecolor = 'blue',lw = 1, zorder = 5, label ="Fase a, pre-
falla") #desde el inico, 0, 0.5 (angulo, inicio del modulo )es el inicio del
vector, y
                # el prox 0, 1 es lo que se le suma al inicio. entonces el
vector quedaría derecho desde 0 a 0 grados, y va desde 0.5 a (0.5+1) 1.5 de
largo
                # zorder es el nivel de la linea o en este caso flecha que se
esté graficando. significa que si hay 5 lineas superpuestas, la de mas arriba
sera la de zorder 5
                # arrow at 45 degree
                arr2 = plt.arrow((fb/180.*np.pi)-ref, 0, 0.0, vb, alpha
= 0.5, width = 0.04,
                        length_includes_head=True , head_width=0.1 ,
edgecolor = 'black', facecolor = 'red', lw = 1, zorder = 5, label ="Fase b, pre-
falla")

                arr3 = plt.arrow((fc/180.*np.pi)-ref, 0, 0.0, vc, alpha
= 0.5, width = 0.04,
                        length_includes_head=True , head_width=0.1 ,
edgecolor = 'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c,
pre-falla")

                arr4 = plt.arrow((fa1/180.*np.pi)-ref, 0, 0, va1, alpha
= 1, width = 0.04,
                        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black',facecolor = 'blue',lw = 1, zorder = 5, label ="Fase a, post-
falla") #desde el inico, 0, 0.5 (angulo, inicio del modulo )es el inicio del
vector, y
                # el prox 0, 1 es lo que se le suma al inicio. entonces el
vector quedaría derecho desde 0 a 0 grados, y va desde 0.5 a (0.5+1) 1.5 de
largo

```

```

        # zorder es el nivel de la linea o en este caso flecha que se
        # esté graficando. significa que si hay 5 lineas superpuestas, la de mas arriba
        # sera la de zorder 5
        # arrow at 45 degree
        arr5 = plt.arrow((fb1/180.*np.pi)-ref, 0, 0, vb1, alpha
= 1, width = 0.04,
                        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black', facecolor = 'red', lw = 1, zorder = 5, label ="Fase b,
post-falla")

        arr6 = plt.arrow((fc1/180.*np.pi)-ref, 0, 0, vc1, alpha
= 1, width = 0.04,
                        length_includes_head=True , head_width=0.05 ,
edgecolor = 'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c,
post-falla")

    else:
        va=float(a[3])
        fa=float(a[6])
        va1=float(b[3])
        fa1=float(b[6])
        arr1 = plt.arrow((fa/180.*np.pi)-ref, 0, 0, va, alpha = 0.5,
width = 0.04,
                        length_includes_head=True , head_width=0.1 , edgecolor =
'black',facecolor = 'blue',lw = 1, zorder = 5, label="Fase a")
        arr4 = plt.arrow((fa1/180.*np.pi)-ref, 0, 0, va1, alpha = 1,
width = 0.04,
                        length_includes_head=True , head_width=0.05 , edgecolor
= 'black',facecolor = 'blue',lw = 1, zorder = 5, label="Fase a, r0")

    elif a[4]!="":
        vb=float(a[4])
        fb=float(a[7])
        vb1=float(b[4])
        fb1=float(b[7])
        arr2 = plt.arrow((fb/180.*np.pi)-ref, 0, 0, vb, alpha = 0.5, width =
0.04,
                        length_includes_head=True , head_width=0.1 , edgecolor =
'black', facecolor = 'red', lw = 1, zorder = 5, label="Fase b")

        arr5 = plt.arrow((fb1/180.*np.pi)-ref, 0, 0, vb1, alpha = 1, width =
0.04,
                        length_includes_head=True , head_width=0.05 , edgecolor =
'black', facecolor = 'red', lw = 1, zorder = 5, label="Fase b, r0")

    else:
        vc=float(a[5])

```

```

    fc=float(a[8])
    vc1=float(b[5])
    fc1=float(b[8])
    arr3 = plt.arrow((fc/180.*np.pi)-ref, 0, 0, vc, alpha = 0.5, width =
0.04,
                    length_includes_head=True , head_width=0.1 , edgecolor =
'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c")

    arr6 = plt.arrow((fc1/180.*np.pi)-ref, 0, 0, vc1, alpha = 1, width =
0.04,
                    length_includes_head=True , head_width=0.05 , edgecolor =
'black', facecolor = 'green', lw = 1, zorder = 5, label="Fase c, r0")

i=est+10
while i<len(r1): #va de la primera linea del resultado de la estacion para el t2
(estado estacionario) saltando 20 filas
    aux=r1[i].split(";")
    d=1-(0.05*(i/100))
    if aux[3]!="":
        fa=float(aux[6])
        va=float(aux[3])
        c = plt.scatter((fa/180.*np.pi)-ref, va,s=2,color='blue',
alpha=d)
    if aux[4]!="":
        fb=float(aux[7])
        vb=float(aux[4])
        c = plt.scatter((fb/180.*np.pi)-ref, vb,s=2,color='red',
alpha=d)
    if aux[5]!="":
        fc=float(aux[8])
        vc=float(aux[5])
        c = plt.scatter((fc/180.*np.pi)-ref, vc,s=2,color='green',
alpha=d)
    i=i+20
angle = np.deg2rad(30)
ax.legend(loc='upper right', bbox_to_anchor=(0.67 + np.cos(angle)/2, 0.84 +
np.sin(angle)/2))
plt.savefig(titulo6)
plt.clf()
plt.close()
print("El grafico fue almacenado en la carpeta")

```

6.1.5 Algoritmo de detección de eventos de tensión

6.1.5.1 Cálculo de componentes simétricas

“calculo_componentes_simetricas.py”


```

tipo="bt" #monofasico m, bifasico a tierra bt o bifasico b
h = open("C:/SimulacionesATP/algoritmos/bt.csv", "r+")
vpost=h.readlines()
h.close()
i=1
for i in estaciones:
    k=i+10 #trabajo con los valores en estado estacionario
    while k<len(vpost):
        b=vpost[k].split(";")
        #almaceno los valores de tension y sus fases en radianes y los convierto
en complejos: vac, vbc y vcc
        va=float(b[3])
        fa=((float(b[6]))*math.pi)/180
        vac=cmath.rect(va,fa)
        vb=float(b[4])
        fb=((float(b[7]))*math.pi)/180
        vbc=cmath.rect(vb,fb)
        vc=float(b[5])
        fc=((float(b[8]))*math.pi)/180
        vcc=cmath.rect(vc,fc)
        #calculo las componentes simetricas
        homo=(vac+vbc+vcc)/3
        inv=(vac+a*vbc+a*a*vcc)/3
        dir=(vac+vbc*a*a+vcc*a)/3
        #llevo las componenes simetricas a coordenadas polares para almacenar en
el .csv
        mod_homo=abs(homo)
        fase_homo=cmath.phase(homo)*180/math.pi
        mod_inv=abs(inv)
        fase_inv=cmath.phase(inv)*180/math.pi
        mod_dir=abs(dir)
        fase_dir=cmath.phase(dir)*180/math.pi
        lres=str(nodo)+";"+tipo+";"+str(i)+";"+b[2]+";"+b[1]+";"+b[3]+";"+b[4]+";
"+b[5]+";"+b[6]+";"+b[7]+";"+b[8]+";"
        lres=lres+str(mod_homo)+";"+str(mod_inv)+";"+str(mod_dir)+";"+str(fase_h
omo)+";"+str(fase_inv)+";"+str(fase_dir)+";"+"\\n"
        resultados.append(lres)
        k=k+10
tipo="b" #monofasico m, bifasico a tierra bt o bifasico b
h = open("C:/SimulacionesATP/algoritmos/b.csv", "r+")
vpost=h.readlines()
h.close()
i=1
for i in estaciones:
    k=i+10 #trabajo con los valores en estado estacionario
    while k<len(vpost):
        b=vpost[k].split(";")

```

```

        #almaceno los valores de tension y sus fases en radianes y los convierto
en complejos: vac, vbc y vcc
        va=float(b[3])
        fa=((float(b[6]))*math.pi)/180
        vac=cmath.rect(va,fa)
        vb=float(b[4])
        fb=((float(b[7]))*math.pi)/180
        vbc=cmath.rect(vb,fb)
        vc=float(b[5])
        fc=((float(b[8]))*math.pi)/180
        vcc=cmath.rect(vc,fc)
        #calculo las componentes simetricas
        homo=(vac+vbc+vcc)/3
        inv=(vac+a*vbc+a*a*vcc)/3
        dir=(vac+vbc*a*a+vcc*a)/3
        #llevo las componenes simetricas a coordenadas polares para almacenar en
el .csv
        mod_homo=abs(homo)
        fase_homo=cmath.phase(homo)*180/math.pi
        mod_inv=abs(inv)
        fase_inv=cmath.phase(inv)*180/math.pi
        mod_dir=abs(dir)
        fase_dir=cmath.phase(dir)*180/math.pi
        lres=str(nodo)+";"+tipo+";"+str(i)+";"+b[2]+";"+b[1]+";"+b[3]+";"+b[4]+
;"+b[5]+";"+b[6]+";"+b[7]+";"+b[8]+";"
        lres=lres+str(mod_homo)+";"+str(mod_inv)+";"+str(mod_dir)+";"+str(fase_h
omo)+";"+str(fase_inv)+";"+str(fase_dir)+";"+"\n"
        resultados.append(lres)
        k=k+10
ubicacion="C:/SimulacionesATP/algoritmos/componentes/resultados_nodo_"+str(nodo)
+".csv"
r=open(ubicacion,"w+")
for i in range(len(resultados)):
    r.write(resultados[i])
r.close()
print("Han finalizado los cálculos")

```

6.1.5.2 Cálculo de E1

```

"calculo_e1.py"
import cmath
import math
#####
l = open("C:/SimulacionesATP/algoritmos/resultados_fallas.csv","r+")
ll=l.readlines()
l.close()
resultados=list()
lineauno="estacion;vfa;vfb;vfc;fa;fb;fc;vab;vbc;vca;fvab;fvbc;fvca;E1\n"

```

```

resultados.append(lineauno)
estaciones=[1,2,5,6,7,8,9,10] #trabajo solo con los nodos trifasicos
for i in estaciones:
    b=ll[i].split(";")
    #almaceno los valores de tension y sus fases en radianes y los convierto en
    complejos: vac, vbc y vcc
    va=float(b[3])
    fa=((float(b[6]))*math.pi)/180
    vac=cmath.rect(va,fa)
    vb=float(b[4])
    fb=((float(b[7]))*math.pi)/180
    vbc=cmath.rect(vb,fb)
    vc=float(b[5])
    fc=((float(b[8]))*math.pi)/180
    vcc=cmath.rect(vc,fc)
    #calculo las tensiones de linea
    vab_c=vac-vbc
    vbc_c=vbc-vcc
    vca_c=vcc-vac
    #llevo las tensiones de linea a coordenadas polares para almacenar en el
    .csv
    vab=abs(vab_c)
    fvab=cmath.phase(vab_c)*180/math.pi
    vbc=abs(vbc_c)
    fvbc=cmath.phase(vbc_c)*180/math.pi
    vca=abs(vca_c)
    fvca=cmath.phase(vca_c)*180/math.pi
    ##### CALCULO DE E1 #####
    e1=(va+vb+vc)/3
    lres=str(i)+";"+b[3]+";"+b[4]+";"+b[5]+";"+b[6]+";"+b[7]+";"+b[8]+";"
    lres=lres+str(vab)+";"+str(vbc)+";"+str(vca)+";"+str(fvab)+";"+str(fvbc)+";"
+str(fvca)+";"+str(e1)+";"+ "\n"
    resultados.append(lres)
ubicacion="C:/SimulacionesATP/algoritmos/resultados_vlinea.csv"
r=open(ubicacion,"w+")
for i in range(len(resultados)):
    r.write(resultados[i])
r.close()
print("Han finalizado los cálculos")

```

6.1.5.3 Determinación de eventos de tensión

```

"eventosdetension.py"
import cmath
import math
#####
nodo=864 #MODIFICAR SEGUN NODO
l = open("C:/SimulacionesATP/algoritmos/resultados_vlinea.csv","r+")

```

```

le1=l.readlines()
l.close()
ubicacion="C:/SimulacionesATP/algoritmos/componentes/resultados_nodo_"+str(nodo)
+".csv"
k = open(ubicacion,"r+")
vpost=k.readlines()
k.close()
resultados=list()
lineauno="nodo;tipo;estacion;tiempo;resistencia;va;vb;vc;vab3;vbc3;vca3;vmin;vma
x;homopolar;directa;inversa;h ideal;d ideal;i ideal;e1;h;evento\n"
resultados.append(lineauno)
for i in range(1,len(vpost),1):
    b=vpost[i].split(";")
    est=int(b[2])
    if est>4:
        a=le1[est-2].split(";")
    else:
        a=le1[est].split(";")
    e1_pre=float(a[13])
    #almaceno los valores de tension y sus fases en radianes y los convierto en
complejos rectangulares: var, vbr y vcr
    va=float(b[5])
    fa=((float(b[8]))*math.pi)/180
    var=cmath.rect(va,fa)
    vb=float(b[6])
    fb=((float(b[9]))*math.pi)/180
    vbr=cmath.rect(vb,fb)
    vc=float(b[7])
    fc=((float(b[10]))*math.pi)/180
    vcr=cmath.rect(vc,fc)
    #calculo las tensiones de linea
    vab_c=var-vbr
    vbc_c=vbr-vcr
    vca_c=vcr-var
    #llevo las tensiones de linea a coordenadas polares para almacenar en el
.csv
    vab=abs(vab_c)
    fvab=cmath.phase(vab_c)*180/math.pi
    vbc=abs(vbc_c)
    fvbc=cmath.phase(vbc_c)*180/math.pi
    vca=abs(vca_c)
    fvca=cmath.phase(vca_c)*180/math.pi
    #calculo Vmin y Vmax
    vab3=vab/(3**0.5)
    vbc3=vbc/(3**0.5)
    vca3=vca/(3**0.5)
    tensiones=[va,vb,vc,vab3,vbc3,vca3]
    vmin=min(tensiones)

```

```

vmax=max(tensiones)
tensionesfase=[va,vb,vc]
lim_inf=min(tensionesfase)
lim_sup=max(tensionesfase)
#calculo h
h=vmin/e1_pre
if (lim_inf > 0.9) and (lim_sup < 1.1):
    evento="N"
    hideal=0
    dideal=0
    iideal=0
elif vmax>=1.1:
    h1=e1_pre*(1-h) #homopolar
    h2=e1_pre #directa
    h3=0 #inversa
    errh=(float(b[11])-h1)**2+(float(b[12])-h2)**2+(float(b[13])-h3)**2
    if h>0.75:
        i1=2*(e1_pre*(1-h)) #homopolar
        i2=e1_pre #directa
        i3=0 #inversa
        erri=(float(b[11])-i1)**2+(float(b[12])-i2)**2+(float(b[13])-i3)**2
        hi1=i1
        di2=i2
        ini3=i3
    else:
        ii1=e1_pre/2 #homopolar
        ii2=e1_pre*(1+((2*h)/3)) #directa
        ii3=e1_pre*(1-((2*h)/3)) #inversa
        erri=(float(b[11])-ii1)**2+(float(b[12])-ii2)**2+(float(b[13])-
ii3)**2
        hi1=ii1
        di2=ii2
        ini3=ii3
    error=[errh,erri]
    errmin=min(error)
    if errmin==errh:
        evento="h"
        hideal=h1
        dideal=h2
        iideal=h3
    elif errmin==erri:
        evento="i"
        hideal=hi1
        dideal=di2
        iideal=ini3
elif vmin<=0.9:
    #calculo las componentes simetricas ideales con subindices 1:homopolar
    2:directa 3:inversa

```

```

a1=0
a2=e1_pre*h
a3=0
b1=(e1_pre*(1-h))/3
b2=(e1_pre*(2+h))/3
b3=(e1_pre*(1-h))/3
c1=0
c2=(e1_pre*(1+h))/2
c3=(e1_pre*(1-h))/2
d1=0
d2=(e1_pre*(1+h))/2
d3=(e1_pre*(1-h))/2
e1=(e1_pre*(1-h))/3
e2=(e1_pre*(1+(2*h)))/3
e3=(e1_pre*(1-h))/3
f1=0
f2=(e1_pre*(1+(2*h)))/3
f3=(e1_pre*(1-h))/3
g1=0
g2=(e1_pre*(1+(2*h)))/3
g3=(e1_pre*(1-h))/3
#compruebo si Vmin es una tension de fase
if (vmin==va) or (vmin==vb) or (vmin==vc):
    erra=(float(b[11])-a1)**2+(float(b[12])-a2)**2+(float(b[13])-a3)**2
    errb=(float(b[11])-b1)**2+(float(b[12])-b2)**2+(float(b[13])-b3)**2
    errd=(float(b[11])-d1)**2+(float(b[12])-d2)**2+(float(b[13])-d3)**2
    errf=(float(b[11])-f1)**2+(float(b[12])-f2)**2+(float(b[13])-f3)**2
    error=[erra,errb,errd,errf]
    errmin=min(error)
    if errmin==erra:
        evento="a"
        hideal=a1
        dideal=a2
        iideal=a3
    elif errmin==errb:
        evento="b"
        hideal=b1
        dideal=b2
        iideal=b3
    elif errmin==errd:
        evento="d"
        hideal=d1
        dideal=d2
        iideal=d3
    else:
        evento="f"
        hideal=f1
        dideal=f2

```

```

        iideal=f3
    else:
        erra=(float(b[11])-a1)**2+(float(b[12])-a2)**2+(float(b[13])-a3)**2
        errc=(float(b[11])-c1)**2+(float(b[12])-c2)**2+(float(b[13])-c3)**2
        erre=(float(b[11])-e1)**2+(float(b[12])-e2)**2+(float(b[13])-e3)**2
        errg=(float(b[11])-g1)**2+(float(b[12])-g2)**2+(float(b[13])-g3)**2
        error=[erra,errc,erre,errg]
        errmin=min(error)
        if errmin==erra:
            evento="a"
            hideal=a1
            dideal=a2
            iideal=a3
        elif errmin==errc:
            evento="c"
            hideal=c1
            dideal=c2
            iideal=c3
        elif errmin==erre:
            evento="e"
            hideal=e1
            dideal=e2
            iideal=e3
        else:
            evento="g"
            hideal=g1
            dideal=g2
            iideal=g3

    lres=b[0]+";"+b[1]+";"+b[2]+";"+b[3]+";"+b[4]+";"+b[5]+";"+b[6]+";"+b[7]+";"
+str(vab3)+";"+str(vbc3)+";"+str(vca3)+";"+str(vmin)+";"+str(vmax)+";"
    lres=lres+b[11]+";"+b[12]+";"+b[13]+";"+str(hideal)+";"+str(dideal)+";"+str(
iideal)+";"+str(e1_pre)+";"+str(h)+";"+evento+";"+ "\n"
    resultados.append(lres)
ubicacion2="C:/SimulacionesATP/algoritmos/resultados/eventos_nodo_"+str(nodo)+".
csv"
r=open(ubicacion2,"w+")
for i in range(len(resultados)):
    r.write(resultados[i])
r.close()
print("Han finalizado los cálculos")

```

6.1.5.4 Cálculo de valores RMS en MODELS

```

MODEL RMSMOD3
comment-----
Calculates the true rms value of a 3-phase signal based on integration.

```

```

SampleFreq frequency should be whole multiple of power frequency.
-----endcomment
DATA
xrms_ini {dflt:-1} -- initial rms value
scale {dflt:1} --divide la salida rms por este valor, usar valor de linea
SampleFreq {dflt:400} -- sample frequency in Hz
tsampleo {dflt:0.1} --el tiempo sobre muestra la tension en .lis
n {dflt:3} --numero de fases
nodo --ingresar el nodo sobre el cual se sensa la tension o corriente
muestra --ingresar el numero de muestra de la foto, 1, 2 o 3 por ej
tipo {dflt:1}-- 2 para corriente y 1 para tension
CONST freq {val:60}
INPUT
x[1..n] -- monitored signal
VAR
xrms[1..n]
x2[1..n] -- internal, x*x
ix2[1..n] -- internal, integral of x2
period -- 1/freq
k
written --variable de valor 0 o 1 para mostrar o no el valor
TIMESTEP min:recip(SampleFreq) --this restricts local timestep to avoid overflow
DELAY CELLS(ix2[1..n]): recip(freq*timestep) +2
INIT
period := recip(freq)
written:= 0
histdef(ix2[1..n]) := 0
integral(x2[1..n]) := 0
IF xrms_ini <0 THEN xrms[1..n]:=0 ELSE xrms[1..n]:=xrms_ini ENDIF
ENDINIT
EXEC
FOR k:=1 TO n DO
x2[k] := x[k]*x[k]
ix2[k] := integral(x2[k])
IF t>period THEN
xrms[k] := sqrt((ix2[k] - delay(ix2[k], period))*freq)/(scale/sqrt(3))
ENDIF
ENDFOR
if (T>=tsampleo) and (written=0) then
write('rms',' ',muestra,' ',nodo)
for k:=1 to n do
write(xrms[k], tsampleo)
endfor
written:=1
endif
ENDEXEC
ENDMODEL

```

6.1.5.5 Cálculo de fasores en *MODELS*

```

MODEL WRITEPROBEI
comment-----
Calculates the phasor of input V and I with FFT Radix2-8 algorithm.

```

The phasor is written to the LIS-file at the specified TIME.
 SampleFreq should be a whole multiple of the 8* power frequency.

-----endcomment

DATA

N {DFLT:3} --number of phases

TIME {DFLT:0} --sample time

SampleFreq {dfilt:480}--sample freq

nodo --ingresar id de estacion de medicion

muestra --ingresar el numero de foto de la muestra, 1, 2 o 3 por ej

CONST freq {val:60} --cambiar desde .atp a la frec del sistema

INPUT

V[1..N], I[1..N] --input signal to be transformed

VAR

P[1..N], Q[1..N],

k

Imagn[1..N], langl[1..N], Vmagn[1..N], Vangl[1..N], reV[1..N], imV[1..N],

rel[1..N], iml[1..N], OMEGA, alpha, x1, x3, x5, x7, xre, xim, delta_T[0..7],

written

HISTORY

V[1..N] {DFLT:0}

I[1..N] {DFLT:0}

timestep min: recip(SampleFreq)

DELAY CELLS (V[1..N]) : recip(FREQ*timestep)+2

DELAY CELLS (I[1..N]) : recip(FREQ*timestep)+2

INIT

OMEGA:= 2*PI*FREQ

written:=0

reV[1..N] :=0

imV[1..N] :=0

rel[1..N] :=0

iml[1..N] :=0

alpha:=1/sqrt(2)

for k:= 0 to 7 do

delta_T[k] := k*recip(FREQ*8)

endfor

ENDINIT

EXEC

if written=0 then

if (FREQ>0) and (t>=TIME-recip(FREQ)) then

for k:=1 to N do

x1 := delay(V[k],delta_T[0],1) - delay(V[k],delta_T[4],1)

x3 := delay(V[k],delta_T[2],1) - delay(V[k],delta_T[6],1)

x5 := delay(V[k],delta_T[1],1) - delay(V[k],delta_T[5],1)

x7 := delay(V[k],delta_T[3],1) - delay(V[k],delta_T[7],1)

xRe := x1 + (x5 - x7)*alpha

xIm := x3 + (x5 + x7)*alpha

ReV[k] := (xRe*cos(OMEGA*T)+xIm*sin(OMEGA*T))/4

ImV[k] := (xIm*cos(OMEGA*T)-xRe*sin(OMEGA*T))/4

x1 := delay(I[k],delta_T[0],1) - delay(I[k],delta_T[4],1)

x3 := delay(I[k],delta_T[2],1) - delay(I[k],delta_T[6],1)

x5 := delay(I[k],delta_T[1],1) - delay(I[k],delta_T[5],1)

x7 := delay(I[k],delta_T[3],1) - delay(I[k],delta_T[7],1)

xRe := x1 + (x5 - x7)*alpha

xIm := x3 + (x5 + x7)*alpha

Rel[k] := (xRe*cos(OMEGA*T)+xIm*sin(OMEGA*T))/4

```

    ImI[k] := (xIm*cos(OMEGA*T)-xRe*sin(OMEGA*T))/4
  endfor
endif
if t>=TIME then
  if FREQ>0 then
    for k:=1 to N do
      P[k]:=(reV[k]*rel[k]+imV[k]*imI[k])/2
      Q[k]:=(imV[k]*rel[k]-reV[k]*imI[k])/2
      Imagn[k]:=sqrt(rel[k]**2+imI[k]**2)
      Vmagn[k]:=sqrt(reV[k]**2+imV[k]**2)
      if (imI[k]=0) and (rel[k]=0) then
        langI[k]:=0
      else
        langI[k]:=atan2(imI[k],rel[k])*180/PI
      endif
      if (imV[k]=0) and (reV[k]=0) then
        VangI[k]:=0
      else
        VangI[k]:=atan2(imV[k],reV[k])*180/PI
      endif
    endfor
  else
    for k:=1 to N do
      P[k]:=I[k]*V[k]/2
      Q[k]:=0
      Imagn[k]:=I[k]
      langI[k]:=0
    endfor
  endif
  write('estacion de medicion',' ',nodo,' ',muestra)
  for k:=1 to N do
    write(Imagn[k],' ', langI[k],' ', P[k],' ',Q[k],' ', TIME)
  endfor
  written:=1
endif
endif
ENDEXEC
ENDMODEL

```