FACULTAD 50 AÑOS DE INGENIERÍA UNaM

2024 JIDeTEV ISSN 2591-4219

XIV Jornadas de Investigación, Desarrollo Tecnológico, Extensión, Vinculación y Muestra de la Producción

Simulador de Circuitos Eléctricos Trifásicos de cargas Balanceadas y Desbalanceadas en MATLAB.

Martínez Felipe [a], Meier Rafael [a,b], Portillo Franco [a,b], Statkiewicz Maximiliano [a,c]

[a] *Universidad Nacional de Misiones, Facultad de Ingeniería, Oberá, Misiones, Argentina.*
[b] *GID-IE, FI-UNaM, Oberá, Misiones, Argentina.*
[c] *LABSE, FI-UNaM, Juan Manuel de Rosas 325, Oberá, Misiones, Argentina*

e-mails: felipemartines2001@gmail.com, rafaelmeier2001@gmail.com
francoportillo191002@gmail.com, maximistat@gmail.com

Resumen

Este artículo presenta el desarrollo y la implementación de un simulador de circuitos eléctricos trifásicos utilizando el software MATLAB,

El objetivo principal de este es proporcionar una herramienta educativa que permita a los estudiantes y profesionales del área de ingeniería eléctrica realizar cálculos y análisis de circuitos trifásicos en configuraciones balanceadas y desbalanceadas. El alcance del simulador incluye la evaluación de diferentes métodos de conexión, como la configuración en estrella (Y) y triángulo (Δ).

Se desarrolló un simulador en MATLAB que utiliza funciones y scripts personalizados para modelar y analizar circuitos trifásicos. La metodología incluyó la programación de algoritmos para el cálculo de los parámetros eléctricos de cargas balanceadas y desbalanceadas, y la implementación de interfaces gráficas de usuario para facilitar la interacción con el simulador.

Los resultados muestran que el simulador proporciona valores precisos de corriente, voltaje y potencia, y permite a los usuarios observar el comportamiento de los circuitos bajo diferentes condiciones de carga. Además, el simulador facilita la comprensión de los conceptos de potencia trifásica y el impacto de las conexiones en el rendimiento del sistema.

Palabras Clave *– Circuitos Trifásicos Desbalanceados y Balanceados, Conexión Estrella / Triángulo , Corriente Trifásica , Potencia Trifásica , Simulador.*

1 Introducción

El estudio y análisis de circuitos eléctricos trifásicos es fundamental en la ingeniería eléctrica debido a su aplicación extendida en sistemas de generación, transmisión y distribución de energía eléctrica. En particular, los métodos de conexión en estrella (Y) y triángulo (Δ) son esenciales para entender el comportamiento de las cargas y las interacciones entre las fases en estos sistemas.

Investigaciones recientes han explorado métodos avanzados de simulación y modelado para facilitar el estudio de estos circuitos. Herramientas como MATLAB han demostrado ser extremadamente útiles para la simulación de circuitos eléctricos debido a sus capacidades para cálculos algebraicos en sistemas complejos y su interfaz gráfica que permite una visualización intuitiva de los resultados. Los simuladores en MATLAB permiten a los estudiantes y profesionales realizar cálculos detallados y experimentar con diferentes configuraciones de carga y conexión, mejorando la comprensión teórica y práctica de los sistemas trifásicos.

Sin embargo, a pesar de la disponibilidad de diversas herramientas y recursos, los simuladores específicos para circuitos trifásicos que integran tanto el análisis de cargas balanceadas como desbalanceadas en una única plataforma educativa son relativamente escasos. Este trabajo aborda esta necesidad mediante la creación de un simulador en MATLAB que ofrece una solución integral para el análisis de circuitos trifásicos en un entorno académico.

El presente informe está estructurado de la siguiente manera: en la sección 2 se detallará el desarrollo metodológico del simulador, incluyendo los algoritmos y enfoques utilizados en su programación. La sección 3 presentará los resultados obtenidos del uso del simulador, con ejemplos prácticos que ilustran su funcionalidad. Finalmente, la sección 4 discutirá las conclusiones del trabajo, abordando la eficacia del simulador y las recomendaciones para futuros desarrollos.

Desarrollo

1.1 Metodología y Software utilizado

El desarrollo del trabajo partió por proponer una alternativa resolutiva de circuitos trifásicos en los distintos sistemas de conexionamiento y carga. Para ello y en base a lo estudiado, se investigó a fondo una forma de interconectar los conocimientos adquiridos con la realización de un programa informático a modo de automatizar y optimizar la solución a la problemática propuesta. En consecuencia, a lo pensado se optó por hacer uso de un Software denominado “MATLAB” el cuál se basa en el desarrollo de algoritmos e interfaz basadas en el cálculo numérico.

El Software en cuestión permitió realizar una toma de datos por parte del usuario, posteriormente procesarlos y dar una respuesta. Esta respuesta estaría dada en función a la problemática propuesta por el usuario, es decir, retornaría los cálculos entregados en forma gráfica y analítica en función si los datos brindados eran para un sistema estrella o triangulo con carga balanceada o desbalanceada.

En consecuencia denominamos “Simulador” al programa completo ya que este mismo se realizó en 3 bloques fundamentales, Estos 3 bloques fundamentales eran las 3 configuraciones pautadas, realizadas cada una con su código pertinente en MATLAB acompañada de su interfaz gráfica

En base a esto, el usuario debería elegir previamente el tipo de configuración que desea simular para proceder con la inserción de los datos y que se realicen los cálculos apropiados.

1.2 Funcionamiento del programa y funciones utilizadas

El Simulador funciona partiendo de la elección previa del tipo de configuración a simular, en base a este proceso se ejecuta su respectivo programa y se abre la interfaz gráfica correspondiente, la cual permite visualizar un gráfico genérico de como está dispuesta la configuración elegida.

Luego del proceso anterior, el usuario se encuentra con "casillas" ubicadas en el gráfico genérico los cuales son usados para realizar la introducción de los datos, los datos que deben ser brindados son los valores de las impedancias, tanto parte real como imaginaria y las tensiones. Estos datos claramente correspondientes para dada la configuración en particular.

La explicación que se realizó previamente es con lo que el usuario interactúa, el funcionamiento del programa internamente funciona meramente por el cálculo numérico con los datos brindados por su parte.

El funcionamiento interno del programa funciona de la siguiente forma:

Se leen los datos brindados por el usuario y se les asignan correspondientes variables, las cuales se las trabaja matemáticamente. En caso de las impedancias se levantan los datos y se definen los complejos, se hallan sus módulos y ángulos. En el caso de las tensiones se levantan sus datos y se las desfasa en sus respectivos ángulos.

Dependiendo la configuración elegida el programa varía en sus cálculos, estos cálculos se dan mediante las fórmulas generales propuestas para cada tipo de configuración, las cuales se pueden encontrar como de costumbre en una bibliografía específica del tema.

Con los cálculos realizados se proceden a realizar los vectores tensión/corriente dependiendo el caso de análisis.

A continuación, algunas de las "funciones y palabras reservadas" utilizadas:

1. App."(variable)": Usada para referirnos al valor almacenado en el casillero donde se insertan los datos.

2. Quiver: Usada para la realización de las gráficas vector tensión y corriente.
3. Text: Usada para señalización de texto o valores en las gráficas.

1.3 Componentes de la Interfaz Gráfica

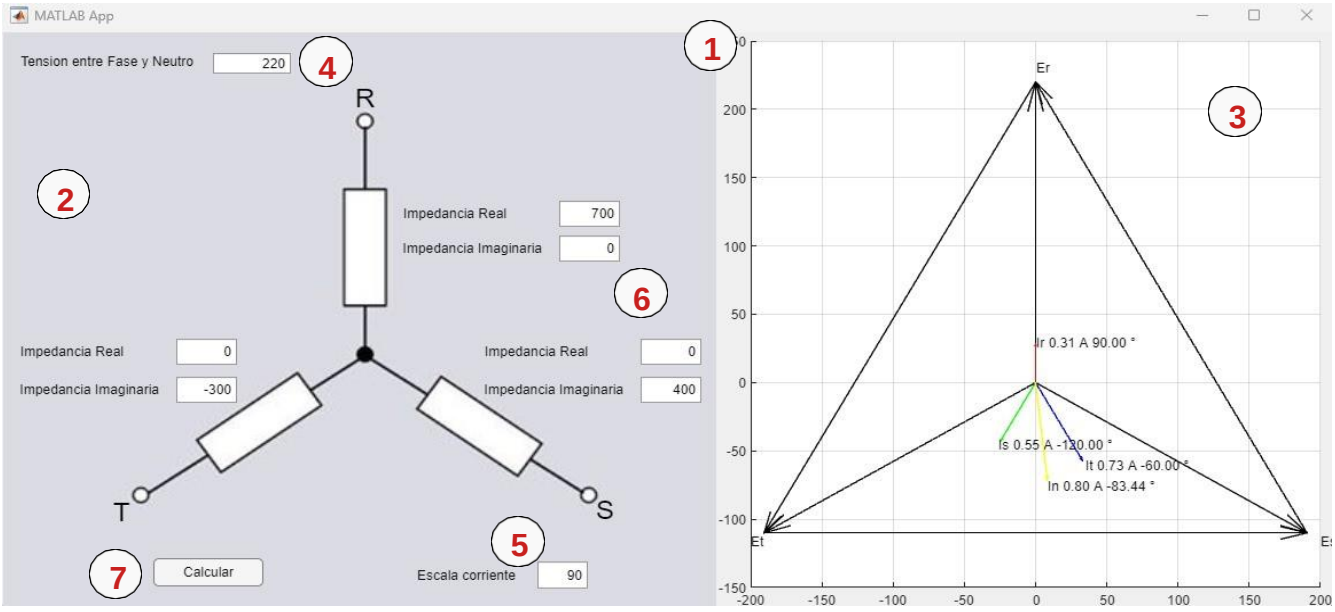
Las interfaces gráficas y los componentes de cada configuración son los siguientes:

<u>Estrella</u> <u>sin</u> <u>Neutro</u>:

Leyendas:

1. Interfaz Gráfica.
2. Gráfico genérico configuración Estrella.
3. Gráfico vectores tensión y neutro desplazado.
4. Gráfico vectores corriente.
5. Introducción de tensión de fase respecto al neutro.
6. Introducción de valores de impedancias.
7. Botón para dar inicio a la simulación.

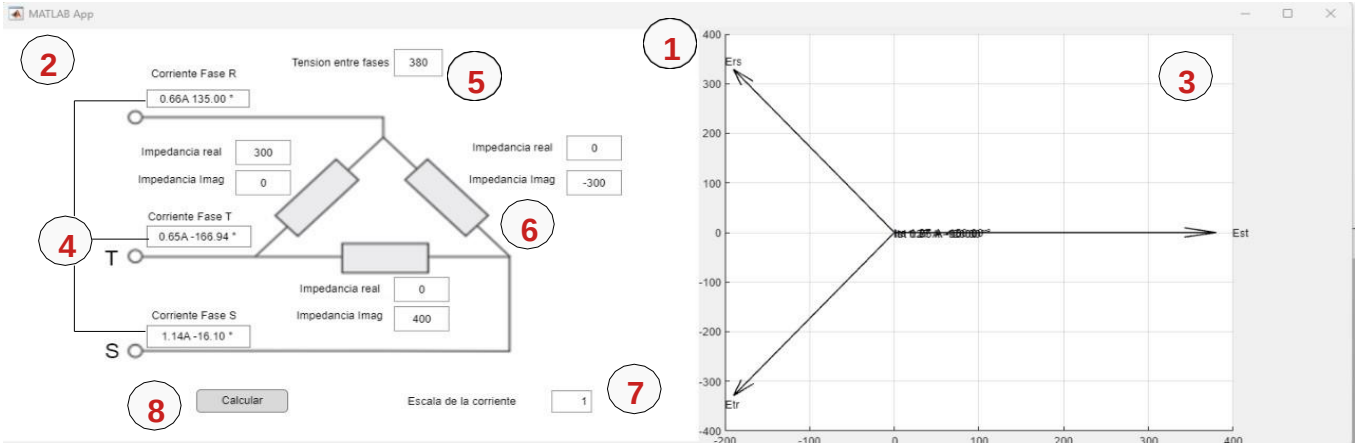
Estrella con neutro:



Leyendas:

1. Interfaz Gráfica.
2. Gráfico genérico configuración Estrella.
3. Gráfico vectores corriente.
4. Introducción de valor de tensión de fase respecto al neutro.
5. Proporción de los vectores corriente respecto a la escala del gráfico.
6. Introducción de valores de impedancias.
7. Botón para dar inicio a la simulación.

Triángulo:



Leyendas:

1. Interfaz Gráfica.
2. Gráfico genérico configuración Triángulo.
3. Gráfico vectores corriente.
4. Valores de corriente de fases con su respectivo ángulo, arrojado por el programa.
5. Introducción de valor de tensión entre fases.
6. Introducción de valores de impedancias.
7. Proporción de los vectores corriente respecto a la escala del gráfico.
8. Botón para dar inicio a la simulación.

2 Conclusión

En este trabajo hemos desarrollado un simulador en el software “MATLAB” con el objetivo de facilitar la enseñanza y comprensión del funcionamiento de un sistema trifásico. El mismo permite a los usuarios obtener un resultado grafico al instante, donde se puede visualizar el comportamiento de un sistema de forma rápida y sencilla.

El simulador demostró ser muy preciso y confiable en los diferentes escenarios para los que fue programado, validando los resultados contra teorías y modelos establecidos, demostrando que es una herramienta útil para el análisis y la enseñanza de sistemas eléctricos trifásicos.

Este posee ciertas limitaciones y se pueden realizar mejoras a futuro, como ampliar su capacidad para simular condiciones más complejas, o una integración que nos permita analizar datos en tiempo real para visualizar el comportamiento de un sistema de forma continua.

3 Apéndice

3.1 Código fuente del programa en MATLAB

Código fuente de configuración en estrella sin neutro:

```
format short;

    cla(app.axes, 'reset')
    cla(app.axes1, 'reset')

    tension_f = app.tension.Value;
    cero=complex(0,0);
    %Desfaso las tensiones 120 y
    Er=complex(tension_f*cosd(90),tension_f*sind(90));
    Es=complex(tension_f*cosd(-30),tension_f*sind(-30));
    Et=complex(tension_f*cosd(210),tension_f*sind(210));
    Ers=Er-Es;
    Etr=Et-Er;
    Est=Es-Et;

    %Impedancia Fase R
    ZRr = app.ZRr.Value;
    ZRi = app.ZRi.Value;
    ZR = complex(ZRr,ZRi);
    modulo_ZR= sqrt(real(ZR)^2+imag(ZR)^2);
    angulo_ZR= atand((real(ZR)/imag(ZR)));

    %Impedancia Fase S
    ZSr = app.ZSr.Value;
    ZSi = app.ZSi.Value;
    ZS = complex(ZSr,ZSi);
    modulo_ZS= sqrt(real(ZS)^2+imag(ZS)^2);
    angulo_ZS= atand((real(ZS)/imag(ZS)));
```

```
%Impedancia Fase T
ZTr = app.ZTr.Value;
ZTi = app.ZTi.Value;
ZT = complex(ZTr,ZTi);
modulo_ZT= sqrt(real(ZT)^2+imag(ZT)^2);
angulo_ZT= atand((real(ZT)/imag(ZT)));

%Cálculo Vector desplazado del neutro
Eoo=((((Er*((1)/(ZR))+Es*((1)/(ZS))+Et*((1)/(ZT))))/(((1)/(ZR))+((1)/(ZS))+((1)/(ZT)))));
modulo_Eoo= sqrt(real(Eoo)^2+imag(Eoo)^2);
angulo_Eoo= atand((real(Eoo)/imag(Eoo)));

%Tension R sobre la carga
Ero=Er-Eoo;
modulo_Ero= sqrt(real(Ero)^2+imag(Ero)^2);
angulo_Ero= atand((real(Ero)/imag(Ero)));

%Tension S sobre la carga
Eso=Es-Eoo;
modulo_Eso= sqrt(real(Eso)^2+imag(Eso)^2);
angulo_Eso= atand((real(Eso)/imag(Eso)));

%Tension T sobre la carga
Eto=Et-Eoo;
modulo_Eto= sqrt(real(Eto)^2+imag(Eto)^2);
angulo_Eto= atand((real(Eto)/imag(Eto)));

%Grafico el quilombo de tensiones
hold (app.axes,"on");
grid (app.axes,"on");
quiver(app.axes,0,0,real(Er),imag(Er),0,'MaxHeadSize',0.3,'Color',[0,0,0]);
text(app.axes,real(Er)*1.05,imag(Er)*1.05,'Er');
quiver(app.axes,0,0,real(Es),imag(Es),0,'MaxHeadSize',0.3,'Color',[0,0,0]);
```

```
        text(app.axes,real(Es)*1.05,imag(Es)*1.05,'Es');
        quiver(app.axes,0,0,real(Et),imag(Et),0,'MaxHeadSize',0.3,'Color',[0,0,0]);
        text(app.axes,real(Et)*1.05,imag(Et)*1.05,'Et');

        %Ploteo tension RS
        quiver(app.axes,real(Es),imag(Es),real(Ers),imag(Ers),0,'MaxHeadSize',0.15,'Color',[0,0,0]);

        %Ploteo tension TR
        quiver(app.axes,real(Er),imag(Er),real(Etr),imag(Etr),0,'MaxHeadSize',0.15,'Color',[0,0,0]);

        %Ploteo tension ST
        quiver(app.axes,real(Et),imag(Et),real(Est),imag(Est),0,'MaxHeadSize',0.15,'Color',[0,0,0]);

        %Ploteo tension del neutro
        quiver(app.axes,0,0,real(Eoo),imag(Eoo),0,'MaxHeadSize',0.15,'Color',[1,0,1]);
text(app.axes,real(Eoo)*1.05,imag(Eoo)*1.05,sprintf('En-n %.2f V',modulo_Eoo),'Color',[1,0,1]);

        %Ploteo de las tensiones reales
  quiver(app.axes,real(Eoo),imag(Eoo),real(Ero),imag(Ero),0,'MaxHeadSize',0.15,'Color',[1,0,0]);
     text(app.axes,real(Er)*1.12,imag(Er)*1.12,sprintf('Er-n %.2f V',modulo_Ero),'Color',[1,0,0]);

  quiver(app.axes,real(Eoo),imag(Eoo),real(Eso),imag(Eso),0,'MaxHeadSize',0.15,'Color',[0,1,0]);
     text(app.axes,real(Es)*1.12,imag(Es)*1.12,sprintf('Es-n %.2f V',modulo_Eso),'Color',[0,1,0]);

  quiver(app.axes,real(Eoo),imag(Eoo),real(Eto),imag(Eto),0,'MaxHeadSize',0.15,'Color',[0,0,1]);
     text(app.axes,real(Et)*1.12,imag(Et)*1.12,sprintf('Er-n %.2f V',modulo_Eto),'Color',[0,0,1]);

        %----------Ploteo de las corrientes en el otro grafico----------
        %Calculo las Corrientes
        Ir=Ero/ZR;
        modulo_Ir= sqrt(real(Ir)^2+imag(Ir)^2);
        angulo_Ir= atan2d(imag(Ir),real(Ir));
        Is=Eso/ZS;
        modulo_Is= sqrt(real(Is)^2+imag(Is)^2);
```

```
angulo_Is= atan2d(imag(Is),real(Is));
It=Eto/ZT;
modulo_It= sqrt(real(It)^2+imag(It)^2);
angulo_It= atan2d(imag(It),real(It));

%.....Corrientes en otro grafico.....-
hold (app.axes1,"on");
grid (app.axes1,"on");
quiver(app.axes1,0,0,real(Ir),imag(Ir),0,'MaxHeadSize',0.15,'Color',[1,0,0])
text(app.axes1,real(Ir)*1.05,imag(Ir)*1.05,sprintf('Ir %.2f A %.2f ° ',modulo_Ir,angulo_Ir));

quiver(app.axes1,0,0,real(Is),imag(Is),0,'MaxHeadSize',0.15,'Color',[0,1,0])
text(app.axes1,real(Is)*1.05,imag(Is)*1.05,sprintf('Is %.2f A %.2f ° ',modulo_Is,angulo_Is));

quiver(app.axes1,0,0,real(It),imag(It),0,'MaxHeadSize',0.15,'Color',[0,0,1])
text(app.axes1,real(It)*1.05,imag(It)*1.05,sprintf('It %.2f A %.2f ° ',modulo_It,angulo_It));
```

1 Bibliografía

- "MATLAB: Introducción a la programación y resolución de problemas" por Stormy Attaway 2da edición; Editorial: Pearson

- "Electrotecnia" por Antonio J. Fernández 6ta edición; Editorial: Ediciones Paraninfo