

AMPLIACIÓN DE LAS CAPACIDADES DE CÁLCULO DEL ANALIZADOR SIMBÓLICO AnSIRE-MAI USANDO LA MATRIZ DE ADMITANCIA INDEFINIDA

EXPANSION OF THE CALCULATION CAPABILITIES OF THE AnSIRE-MAI SYMBOLIC ANALYZER USING THE INDEFINITE ADMITTANCE MATRIX

Francisco J. Viloría M., Francisco J. Araujo R., Jodrick X. Colina T.

Universidad de Los Andes. Facultad de Ingeniería. Mérida- Venezuela. 5101

fviloria@ula.ve

Recibido: 22-02-2025

Aceptado: 21-08-2025

RESUMEN

El analizador simbólico de redes usando la matriz de admitancia indefinida (AnSiRE-MAI) es un programa de análisis de redes eléctricas. Proporciona resultados numéricos y simbólicos y su motor de cálculo se basa en la matriz de admitancia indefinida. En el presente trabajo, se muestran las ampliaciones en las capacidades del AnSiRE-MAI. Se han agregado nuevos elementos, como fuentes variables en el tiempo e interruptores que permiten realizar análisis de transitorios, mejorando las prestaciones del software. También se desarrolló una interfaz de usuario con un instalador para facilitar su uso en computadoras con sistema operativo Windows 7 o superior. El software fue programado en los lenguajes de programación C, C++ y QML

Palabras clave: Analizador simbólico de redes, matriz de admitancia indefinida, interfaz de usuario.

ABSTRACT

The Symbolic Network Analyzer using the Indefinite Admittance Matrix (AnSiRE-MAI) is a program for analyzing electrical circuits. It provides both numerical and symbolic results, and its computational engine is based on the indefinite admittance matrix. This work presents the enhancements made to the capabilities of AnSiRE-MAI. New elements have been added, such as time-varying sources and switches, enabling transient analysis and improving the software's performance. Additionally, a user interface with an installer was developed to facilitate its use on computers running Windows 7 or later. The software was programmed using the C, C++, and QML programming languages.

Keywords: Analyzed symbolic circuits, indefinite admittance matrix, user interface.

Viloria Manzanilla Francisco: Maestría en Automatización e Instrumentación Universidad de Los Andes (ULA). Mérida, Venezuela. Ingeniero Electricista, (ULA). email: fviloria@ula.ve.

Araujo Ruza, Francisco Javier: Ingeniero Electricista, Universidad de Los Andes (ULA), Mérida, Venezuela. email: araujoru@ula.ve.

Colina Trump, Jodrick Xavier: Ingeniero Electricista, Universidad de Los Andes (ULA), Mérida, Venezuela email: jodrickcolina@gmail.com

Introducción

El analizador simbólico de redes (AnSiRE) es un programa de análisis de redes. Inicialmente, fue desarrollado por Omar Ruiz¹ y mejorado por Juan Escalante.²

El motor de cálculo de este software se basaba en el método de análisis nodal modificado (ANM).³ Posteriormente, se realizó una actualización en el software, reemplazando el motor de cálculo por uno más eficiente. Este nuevo motor utiliza la matriz de admitancia indefinida, lo que dio origen al analizador simbólico de redes usando la matriz de admitancia indefinida (AnSiRE-MAI). Esta actualización fue desarrollada por Jesús González.⁴

El analizador simbólico de redes AnSiRE-MAI es capaz de calcular tensiones en nodos y corrientes en ramas, determinar funciones de transferencia y trabajar con el método de superposición con un costo computacional bastante reducido si se compara con el software origen. Este software tiene integrado los siguientes elementos: fuentes independientes en DC, resistencias, capacitores, inductores, fuentes controladas, transformadores ideales, amplificadores operacionales y acoplamiento magnético.

Dentro de las principales ampliaciones de cálculo realizadas al AnSiRE-MAI, se encuentran las siguientes:⁵

⊗ Se implementaron señales variables en él, incluyendo formas de onda senoidal, cuadrada, triangular y diente de sierra. A su vez, se incorporaron dos nuevos elementos: el interruptor de una vía y el interruptor de dos vías para realizar el análisis transitorio.

⊗ Se realizaron mejoras y optimizaciones al software para así mantener la velocidad que lo caracteriza como también brindar al usuario una experiencia más satisfactoria en cuando al planteamiento de las redes a resolver, la ejecución y los resultados.

⊗ Una de las particularidades que también es destacable fue el desarrollo de una interfaz de usuario con su instalador que permite

definir las redes, realizar la ejecución y mostrar los resultados haciendo al software AnSiRE-MAI fácil y eficaz de usar.

Cada uno de los algoritmos, funciones y herramientas incorporadas al AnSiRE-MAI, permiten el uso del software en cualquier unidad curricular impartida en la carrera de Ingeniería Eléctrica y carreras afines a la Universidad de Los Andes o de otra institución universitaria que hagan estudios de redes eléctricas.

MARCO TEÓRICO ANSIRE-MAI

El AnSiRE-MAI es un software en desarrollo, actualmente se encuentra en fase beta, reflejando los resultados en el dominio de la frecuencia. Esto se debe a que se tiene menor costo computacional al resolver las redes en el dominio de la frecuencia que directamente en el dominio del tiempo; el proceso de llevar al dominio de la frecuencia la red, es realizado por el motor de cálculo del AnSiRE-MAI. Con base a lo expuesto, al resultado obtenido habría que aplicarle la transformada inversa de Laplace si se desea el mismo en el dominio del tiempo o interpretar el resultado directamente en el dominio de la frecuencia.

METODOLOGÍA

La ampliación de las capacidades de cálculo del AnSiRE-MAI se fundamenta en la necesidad de incorporar distintas fuentes de alimentación e interruptores, así como también la incorporación de una interfaz de usuario. Cada uno de los algoritmos diseñados y modificados, fueron desarrollados de manera que pudieran trabajar de forma fluida con los demás algoritmos existentes en el software. Para lograr esto, se dividió cada elemento, función y algoritmo en varias unidades de estudio.

Matriz de Admitancia Indefinida

El AnSiRE-MAI, usa como motor de cálculo los conceptos asociados a la matriz de admitancia indefinida. EL análisis de esta matriz puede ser descrito mediante la

consideración de una red de n terminales (con un número arbitrario de elementos de red activos y pasivos conectados de cualquier forma) como se muestra en la Figura 1, existiendo un terminal exterior que es definido como nodo de referencia.

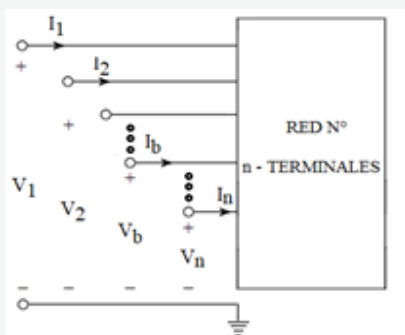


Figura 1. Representación de una red de n terminales.

Sean V_1, V_2, \dots, V_n la transformada de Laplace de las tensiones medidas en los terminales 1, 2, ... n respecto al nodo de referencia externo; I_1, I_2, \dots, I_n las transformadas de Laplace de las corrientes que entran a los terminales 1, 2, ..., n.

Como la red es lineal, las corrientes y tensiones están relacionadas por medio de la ecuación 1

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} + \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{bmatrix} \quad \text{ec. 1}$$

Que puede ser generalizada como:

$$[I] = [Y_{ind}] \cdot [V] + [J] \quad \text{ec. 2}$$

Y_{ind} es llamada Matriz de Admitancia Indefinida (MAI) porque el punto de referencia para las tensiones de los nodos es un punto arbitrario, pero no especificado y se encuentra fuera de la red. J son las corrientes que fluyen por los terminales cuando todos los nodos con conectados al nodo de referencia arbitrario. Si las corrientes J son nulas, entonces la ecuación 2 se convierte en

$$I = [Y_{ind}] \cdot [V] \quad \text{ec. 3}$$

Como en el caso de redes de varios puertos, los elementos de Y_{ind} son admitancias de cortocircuito porque ellos son obtenidos mediante

$$Y_{ij} = \left. \frac{I_i}{V_j} \right|_{V_x=0, x \neq j} \quad \text{ec. 4}$$

lo cual indica que Y_{ij} es la admitancia local o de punto matriz, conocida también como admitancia driving-point entre el terminal i y el nodo externo de referencia.

Cuando todos los demás terminales están conectados a tierra (o al punto de referencia), Y_{ij} representa la admitancia de transferencia desde el terminal j al terminal i, bajo la condición de que todos los terminales excepto el j se encuentren conectados al punto de referencia.

Propiedades de la matriz de admitancia indefinida

La matriz de admitancia indefinida de acuerdo a la forma en que está definida tiene asociadas algunas propiedades interesantes:

⊙ El orden de la matriz es igual al número de nodos existentes en la red.

⊙ La matriz admitancia indefinida es singular, lo que implica que su determinante es igual a cero.

⊙ La suma de los elementos de cualquier columna o fila es igual a cero.

⊙ La matriz admitancia de corto circuito o simplemente la matriz admitancia de una red de (n-1) puertos con un terminal común puede obtenerse de la matriz de admitancia indefinida, considerando a la red compuesta de n terminales, simplemente eliminándose la fila y columna correspondiente al terminal común.

⊙ Si dos terminales de la red se unen para formar uno solo, entonces las entradas de la fila y la columna correspondiente al nuevo terminal es obtenida sumando las entradas de las filas y columnas correspondientes a los dos terminales que están siendo unidos.

⊙ Si se asume uno de los nodos como tierra, es decir con voltaje igual a cero, entonces todos los elementos de la columna

correspondiente a este nodo serán cero, por lo cual se puede prescindir de ella y proceder entonces a eliminar la columna de la matriz. \odot Todos los cofactores de primer orden de la matriz son iguales.

Solución de una red de n-terminales

La red mostrada en la Figura 2, puede resolverse con un mínimo de trabajo, esta red es alimentada por una fuente de corriente conectada en los terminales G-H y los voltajes pueden ser medidos en los terminales J-K y R-S. Si en la red hay más de una fuente, como la red es lineal, puede aplicarse el teorema de superposición.³

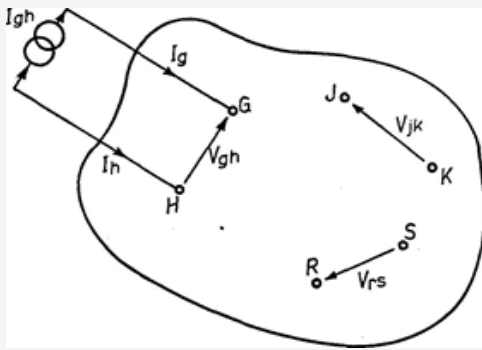


Figura 2. Red de n-terminales, mostrando solamente 3 pares de terminales.³

Las ecuaciones que describen esta red pueden expresarse como:

$$\begin{bmatrix} 0 \\ \vdots \\ I_g \\ I_h \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{1n} \\ \vdots & \vdots & \vdots \\ Y_{g1} & Y_{g2} & \dots & Y_{gn} \\ Y_{h1} & Y_{h2} & \dots & Y_{hn} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ \vdots \\ \vdots \\ V_n \end{bmatrix} \quad \text{ec. 5}$$

Donde $I_g = -I_h = I_{gh}$ y los voltajes están referidos a un nodo de referencia indefinido. Si todos los terminales, excepto G y H, se dejan en circuito abierto, la admitancia de transferencia entre los pares de terminales G, H y J, K, que es la relación entre la corriente I_{gh} y el voltaje V_{JK} , viene dada por:

$$\frac{I_{gh}}{V_{jk}} = \text{sgn}(k-j) \cdot \text{sgn}(h-g) \cdot \frac{Y_{uv}}{Y_{gj,hk}} \quad \text{ec. 6}$$

Donde Y_{uv} es el cofactor de la admitancia indefinida definida en la ec. 5 y $Y_{gj,hk}$ es el cofactor formado por la eliminación de la filas g y h y las columnas j y k, la función $\text{sgn}()$ denota el “signo de”, siendo +1 si $x > 0$.

Se pueden derivar de la ec 6 otras expresiones.^{4,5} La ganancia de voltaje entre los terminales JK y RS cuando se aplica una corriente de excitación en los terminales G y H tiene la forma:

$$G_{jk,rs}^{gh} = \left. \frac{V_{jk}}{V_{rs}} \right|_{I_{gh}} = \text{sgn}(s-r) \cdot \text{sgn}(k-j) \cdot \frac{Y_{gj,hk}}{Y_{gr,hs}} \quad \text{ec. 7}$$

La ganancia de voltaje del driving-point se define cuando uno de los voltajes se mide en el par de terminales donde se aplica la corriente de conducción:

$$G_{jk,gh}^{gh} = \left. \frac{V_{jk}}{V_{gh}} \right|_{I_{gh}} = \text{sgn}(h-g) \cdot \text{sgn}(k-j) \cdot \frac{Y_{gj,hk}}{Y_{gg,hh}} \quad \text{ec. 8}$$

Utilizando estas expresiones, es posible calcular las tensiones o corrientes en cualquier parte de la red, evitando la inversión de matrices, como ocurre en los métodos clásicos empleados para la resolución de redes.

Diseño y modificación de algoritmos del AnSIRE-MAI

El programa original de AnSIRE-MAI hace uso de la mayoría de los elementos básicos (resistencias, capacitores, inductores, fuentes independientes de tensión y corriente, fuentes controladas, amplificadores operacionales, transformadores ideales y acoplamiento magnético).

Para simular una red se hace uso de las reglas para la descripción de redes usados en SPICE (Simulation Program with Integrated Circuit Emphasis), un simulado de propósito general, desarrollado originalmente en la Universidad de California, Berkeley, en 19736.

Este programa ha derivado en una infinidad de herramientas que actualmente se usan a nivel educativo y comercial. Se ha tratado de respetar las definiciones originales para

mantener la compatibilidad, para más información se puede referir al programa AnSIRE-MAI original desarrollado por González Eslava.⁴

Algoritmo de interruptores

La versión original de AnSIRE-MAI, no cuenta con un algoritmo que permita emular el comportamiento de interruptores, la ampliación de las capacidades de cálculo en esta nueva versión, incluye el diseño de dos tipos de interruptores: el interruptor de una vía y el interruptor de dos vías. Aunque la lógica aplicada a ambos es similar, su declaración difiere debido a las características particulares de cada uno.

Interruptor de una vía: Este interruptor tiene la función de abrir o cerrar una rama de una red. Su comportamiento se emula como una resistencia configurable por el usuario, con un valor de resistencia r_{ON} cuando el interruptor está cerrado y r_{OFF} cuando está abierto. Por defecto, los valores asignados son $r_{ON} = 0 \Omega$ y $r_{OFF} = 1 G\Omega$.

La declaración de este elemento se realiza de la siguiente manera:

Nombre_del_elemento Nodo_1 Nodo_2 ON/OFF Tiempo (r_{ON}) (r_{OFF})

Se puede usar un número ilimitado de interruptores, salvo en los casos en que el

tiempo se defina de forma simbólica. En esta situación particular, solo es posible declarar un único interruptor. Esta característica aporta una notable flexibilidad y versatilidad al algoritmo.

Además, cada variable asociada al interruptor puede definirse de manera simbólica, lo que ofrece un alto grado de personalización y adaptabilidad a las necesidades del usuario. Para una mejor comprensión de la simbología utilizada en el interruptor, se recomienda consultar la Figura 2.

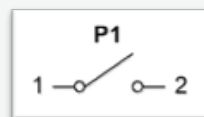


Figura 2. Simbología de interruptor de una vía.

En la Tabla 1 se reflejan los valores obligatorios y por defecto, así como la definición de que cada parámetro que se han de colocar al declarar el elemento.

Interruptor de dos vías: El interruptor de dos vías permite redirigir el flujo de corriente entre dos ramas de la red. tanto el interruptor de una vía como el interruptor de dos vías comparten la misma lógica y utilizan los mismos algoritmos. La principal diferencia radica en su formato de declaración, que se detalla a continuación.

Nombre del elemento Nodo 1 Nodo común

Tabla 1. Valores por defecto del interruptor de una vía

Parámetro	Valor por defecto	Definición/observaciones
Nombre del elemento	obligatorio	Todo interruptor de una vía debe comenzar con la letra clave "P", seguido de caracteres adicionales que permitan identificar y declarar el elemento de forma única.
Nodo 1	obligatorio	Nodo donde se ubicará el terminal positivo del interruptor de una vía.
Nodo 2	obligatorio	Nodo donde se ubicará el terminal negativo del interruptor de una vía.
ON / OFF	obligatorio	Caracteres claves que indican el estado inicial del interruptor. ON si el interruptor está cerrado. OFF si el interruptor está abierto.
Tiempo en segundos	obligatorio	Tiempo de cambio de estado del interruptor, expresado en segundos o de forma simbólica
r_{ON}	opcional	Si no se declara este valor, la resistencia que emula al interruptor en estado cerrado (ON) será 0Ω por defecto. Si el usuario especifica un valor, la resistencia en ON tomará dicho valor, también puede usarse un valor simbólico.
r_{OFF}	opcional	La declaración de este valor es opcional, si no se coloca ningún valor, el valor que tendrá la resistencia que emula al interruptor cuando está abierto (OFF) será de $1 G\Omega$, caso contrario, si el usuario coloca un valor la resistencia en OFF será dicha valor (puede ser también simbólico).

Fuente: Viloria, Araujo, Colina

Tabla 2. Valores por defecto del interruptor de dos vías.

Parámetro	Valor por defecto	Definición/observaciones
<i>Nombre del elemento</i>	obligatorio	Todo interruptor de dos vías debe comenzar con la letra clave "S", seguida de caracteres que permitan declararlo e identificarlo de manera única.
<i>Nodo 1</i>	obligatorio	Nodo donde se ubicará el elemento. El nodo 1 y el nodo común simularán el camino en estado ON hasta que el interruptor cambie de estado tras el tiempo establecido, momento en el cual pasará a OFF.
<i>Nodo común</i>	obligatorio	Nodo común que tendrá el elemento al cambiar de estado.
<i>Nodo 2</i>	obligatorio	Nodo donde se ubicará el elemento. El nodo 2 y el nodo común simularán el camino en estado OFF hasta que el interruptor cambie de estado tras el tiempo establecido, momento en el cual pasará a ON.
<i>Tiempo en segundos</i>	obligatorio	Tiempo de cambio de estado del interruptor en segundos (puede ser también simbólico).
<i>rON</i>	opcional	Si no se declara este valor, la resistencia que emula al interruptor en estado cerrado (ON) será 0 Ω por defecto. Si el usuario especifica un valor, la resistencia en ON tomará dicho valor, que también puede ser simbólico.
<i>rOFF</i>	opcional	La declaración de este valor es opcional, si no se coloca ningún valor, el valor que tendrá la resistencia que emula al interruptor cuando está abierto (OFF) será de 1 GΩ, caso contrario, si el usuario coloca un valor la resistencia en OFF será dicha valor (puede ser también simbólico).

Fuente: Viloria, Araujo, Colina

Nodo 2 ON/OFF Tiempo (rON) (rOFF)
Al igual que el interruptor de una vía, el de dos vías puede declararse con todos sus parámetros de forma simbólica. Además, se pueden incluir múltiples interruptores de ambos tipos simultáneamente. Sin embargo, si el tiempo se declara simbólicamente, solo se permite usar un interruptor, ya sea de una o dos vías. Para comprender mejor la simbología del interruptor de dos vías, consulte la figura 3.

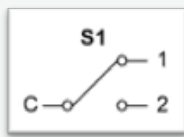


Figura 3. Simbología del interruptor de dos vías.

En la Tabla 2 se reflejan los valores obligatorios y por defecto, así como la definición de que cada parámetro que se han de colocar al declarar el elemento.

Fuentes variables en el tiempo

En el software se han incorporado fuentes variables en el tiempo para aumentar su versatilidad. Estas incluyen señales senoidales, cuadradas, triangulares y de diente de sierra.

Para integrarlas, fue necesario modificar las rutinas existentes y añadir nuevas funciones. Estas actualizaciones se implementaron con éxito, garantizando el correcto funcionamiento de las fuentes variables en el tiempo.

A continuación, se detalla cómo se implementó cada tipo de fuente en el software.

Fuente de señal senoidal: La función de la señal senoidal se implementó a partir de la ec. 9 reflejada a continuación:

$$F(s) = \frac{A \cdot s \cdot \sin(\beta) + A \cdot w \cdot \cos(\beta)}{s^2 + w^2} \quad \text{ec. 9}$$

Esta ecuación es obtenida a partir de aplicar la transformada de Laplace a la función seno con amplitud A, frecuencia w y fase β en el dominio del tiempo, siendo tal ecuación:

$$f(t) = A \cdot \sin(w \cdot t \cdot \beta) \quad \text{ec. 10}$$

Fuente de señal cuadrada: Para implementar la fuente de señal cuadrada, así como las señales triangular y diente de sierra que se describirán a continuación, fue necesario encontrar una solución alternativa. Esto se debe a que las expresiones de estas señales discontinuas no pueden ser

procesadas directamente por el analizador de expresiones del software.

Primero, se analizó un período de la señal y se transformó al dominio de la frecuencia, obteniendo la siguiente ecuación:

$$F(s) = \alpha \frac{A}{s} \quad \text{ec. 11}$$

Donde A representa la amplitud y s la variable en el dominio de la frecuencia. α es un factor variable que depende de la ubicación del punto de evaluación dentro de la ecuación fundamental valiendo esta (1,-1) indicando así el segmento positivo y negativo de la función onda cuadrada.

Al tratarse de una función periódica, el resultado final se le multiplica por la siguiente expresión:

$$F(S) = \frac{1}{1 - e^{-sT}} \quad \text{ec. 12}$$

Donde T es el periodo de la función. Todo este proceso se codificó obteniendo los resultados esperados.

Fuente de señal triangular: Tal como se describió en el apartado anterior, se abordó este elemento desde otra perspectiva por las limitantes explicadas obteniendo la siguiente ecuación:

$$F(S) = \alpha \frac{A}{s} + \frac{4 * \beta * A}{T * s^2} \quad \text{ec. 13}$$

Y finalmente al resultado final se le multiplica por la ecuación ec 12.

Fuente de señal diente de sierra: La lógica implementada para esta fuente es la misma que los casos anteriores, obteniendo la siguiente ecuación:

$$F(S) = \frac{A}{T} * \frac{1}{s^2} \quad \text{ec. 14}$$

Y finalmente al resultado final se le multiplica por la ecuación ec. 12.

Declaración de las fuentes desarrolladas: La declaración de cualquiera de las fuentes implementadas en AnSIRE-MAI, tendrá la siguiente estructura:

Nombre_elemento nodo 1 nodo 2 Tipo_fuente Offset Amplitud Frecuencia Fase

Los parámetros declarados, deben ser estructurados de acuerdo a los datos presentados en la Tabla 3.

Nuevo formato de declaración y corrección de algoritmos que presentaban fallas

A lo largo del desarrollo del software se hicieron múltiples correcciones y modificaciones a las rutinas existentes, esto con la finalidad de obtener los resultados que se esperan, así como optimizar y acoplar las mejoras al software. A continuación, se nombran y se define brevemente algunas de correcciones realizadas a las rutinas existentes:

Modificación a la declaración de fuentes independientes DC:

Esta modificación se realizó con la finalidad de adosar las nuevas fuentes utilizando parte de los algoritmos existentes, por lo que tomó un rol importante en el estudio e implementación de las nuevas fuentes.

Corrección a fuentes DC:

Dado que el motor de cálculo del software opera en el dominio de la frecuencia, todos los elementos declarados se transforman automáticamente a dicho dominio. Sin embargo, esto no ocurría correctamente con las fuentes DC, ya que el motor las interpretaba como señales de impulso en lugar de señales constantes. Este inconveniente fue corregido exitosamente, asegurando su correcto funcionamiento.

Corrección a la cantidad máxima de caracteres en parámetro "valor":

La variable utilizada para almacenar valores como la amplitud de la señal, la resistencia, entre otros, podía ser simbólica o numérica, pero estaba limitada a 15 caracteres. En el caso de valores numéricos, parte de esos 15 caracteres se reservaban para los decimales. Por ello, si se declaraba un valor numérico mayor a 9.999.999 o una variable con más de 15 caracteres, se producía un desbordamiento. Esto hacía que los caracteres sobrantes ocuparan la memoria reservada para otro parámetro, ya que la asignación de memoria era

Tabla 3. Valores por defecto del interruptor de las fuentes variables en el tiempo.

Parámetro	Valor por defecto	Definición/observaciones
<i>Nombre_elemento</i>	obligatorio	Debe iniciar con su letra clave "V" o "I" (dependiendo de si es una fuente de tensión independiente o fuente de corriente de independiente) seguido de caracteres para declarar e identificar al elemento.
<i>Nodo 1</i>	obligatorio	Nodo donde se ubicará el terminal positivo de la fuente.
<i>Nodo 2</i>	obligatorio	Nodo donde se ubicará el terminal negativo de la fuente.
<i>Tipo_fuente</i>	obligatorio	Caracteres claves que indican el tipo de fuente. SIN para la señal seno, SQR para la señal cuadrada, TRI para la señal triangular y SAW para la señal diente de sierra.
<i>Offset</i>	obligatorio	Valor offset agregada a la señal.
<i>Amplitud</i>	obligatorio	Amplitud de la señal.
<i>Frecuencia</i>	obligatorio	Frecuencia de la señal.
<i>Fase</i>	opcional	Fase de la señal, valor por defecto 0

Fuente: Viloria, Araujo, Colina

consecutiva, generando un error semántico. Para solucionar este problema, el límite se amplió a 100 caracteres.

Formato de mensaje de errores: Se añadieron nuevos mensajes de error para informar al usuario cuando declare un elemento incorrectamente u ocurra un fallo durante la ejecución. Esto evita cierres inesperados del software y permite al usuario conocer la razón por la cual la simulación no pudo completarse.

Comandos

Se ha respetado los comandos originales de SPICE6 y se han incluido nuevas características como la aplicando el teorema de superposición (.SP) y la definición del nodo de referencia (.GND)

Esquema general del programa

El esquema general del programa se representa en la figura 4, tiene como archivo de entrada la descripción de la red en un archivo texto con extensión .CIR (denominado netlist como lo establece SPICE original), seguidamente se procesa y se crean la matriz de admitancia indefinida con los elementos pasivos.

Posteriormente se modifica dicha matriz para agregar los elementos activos y ajustar las dimensiones de la matriz según los modelos usados. Posteriormente se procesan los comandos usados según los requerimientos del usuario para obtener las funciones de transferencias, las tensiones o corrientes de forma simbólica.

Tabla 4. Comandos reconocidos por AnSIRE-MAI.

Comando	Definición/observaciones	ejemplo
<i>.AC</i>	Calcula la respuesta de frecuencia de la última <i>.TF</i> definida	<i>.AC DEC (n_puntos) (f_inicial) (f_final)</i>
<i>.GND</i>	Define el nodo de referencia	<i>.GND nodo</i>
<i>.PLOT</i>	Calcula e imprime hasta 4 variables de V o I	<i>.PLOT par1 par2 par3 par4</i>
<i>.SP</i>	Realiza cálculos aplicando el teorema de superposición	<i>.SP</i>
<i>.TF</i>	Calcula la función de transferencia	<i>.TF (Salida) (Fuente_entrada)</i>
<i>.TRAN</i>	Calcula análisis transitorio de la última expresión definida	<i>.TRAN (Paso) (Tiempo_final)</i>

Fuente: Viloria, Araujo, Colina

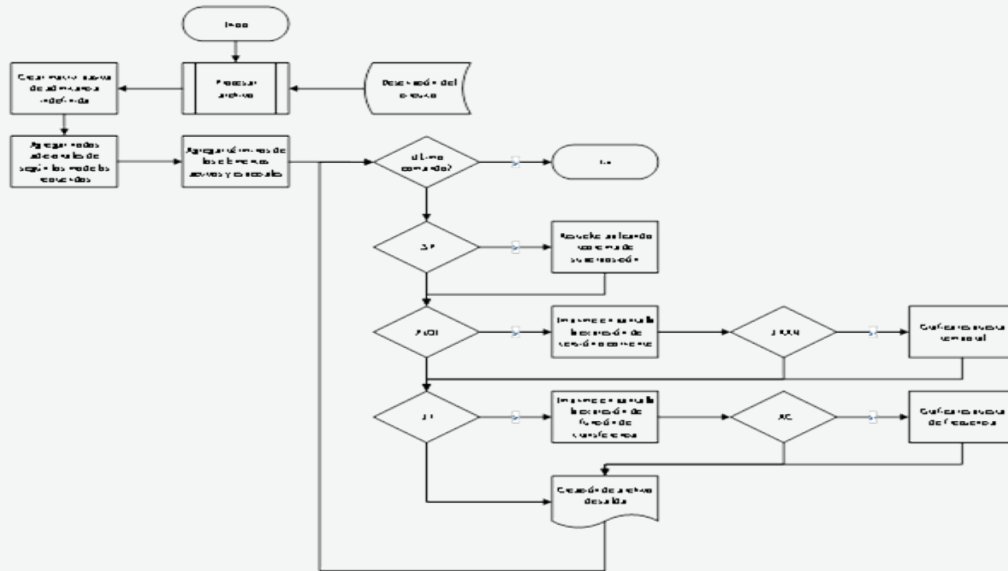


Figura 4. Algoritmo general del motor de cálculo.

INTERFAZ DE USUARIO

La interfaz de usuario del software AnSiRE-MAI fue desarrollada con QT, utilizando C, C++ y QML. Esta mejora simplifica significativamente el análisis de redes, reemplazando el proceso anterior que requería crear archivos de transferencia de datos, ingresar datos manualmente y pasar rutas como argumentos en la línea de comandos.

Con la nueva interfaz, el usuario puede ingresar datos directamente, ejecutar simulaciones, visualizar resultados, modificar archivos existentes y guardar resultados de forma sencilla. Además, incluye herramientas y menús que automatizan tareas y mejoran la experiencia del usuario.

Para tener una mejor idea de cómo se ve esta interfaz de usuario, se puede consultar la figura 5.

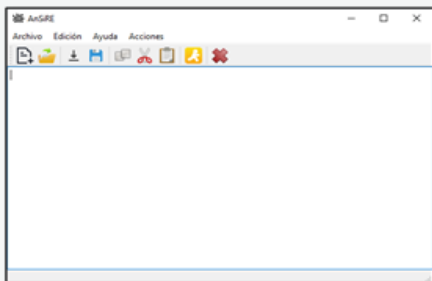


Figura 5. Interfaz de usuario.

En la figura 5, se puede observar que se tienen 4 menús: archivo, edición, ayuda y acciones, a su vez, se observa una barra de herramientas; cada uno de esos menús se desglosa en un conjunto de herramientas bastante intuitivas que a continuación se describirán brevemente:

Menú archivo

La interfaz de usuario cuenta con cinco pestañas principales: "Abrir", "Nuevo", "Guardar", "Guardar como" y "Cerrar", cada una diseñada para mejorar la experiencia del usuario:

Abrir: Permite abrir archivos existentes para visualizarlos, modificarlos o ejecutarlos, facilitando el acceso a proyectos previos.

Nuevo: Crea una nueva ventana de simulación, ideal para limpiar el área de trabajo o iniciar simulaciones con un nombre diferente.

Guardar: Guarda los cambios en la ubicación actual o, si no existe una, solicita un nombre y ubicación, asegurando que el trabajo quede almacenado correctamente.

Guardar como: Permite guardar el documento en una ubicación diferente o crear copias en otras ubicaciones, ofreciendo mayor flexibilidad en la gestión de archivos.

Cerrar: Finaliza el proceso y cierra el programa de manera rápida y sencilla.

Además, cada pestaña tiene una tecla de acceso rápido para facilitar la navegación y el uso eficiente de la interfaz.

Menú edición

La interfaz de usuario incluye tres pestañas adicionales: "Copiar", "Cortar" y "Pegar", diseñadas para facilitar la manipulación de texto:

Copiar: Duplica el texto seleccionado para reutilizarlo en otra parte del documento o en otra aplicación.

Cortar: Elimina el texto seleccionado y lo transfiere al portapapeles para moverlo a otra ubicación.

Pegar: Inserta el texto previamente copiado o cortado en la posición seleccionada.

Cada pestaña cuenta con teclas de acceso rápido, lo que agiliza estas acciones y mejora la eficiencia del usuario al trabajar con el software.

Menú ayuda

La interfaz de usuario incluye tres pestañas informativas: "Acerca de", "Acerca de QT" y "Acerca del programa":

Acerca de: Muestra información sobre el autor y el tutor académico del programa.

Acerca de QT: Proporciona detalles sobre el entorno de desarrollo QT, como su versión y características técnicas.

Acerca del programa: Ofrece orientación sobre el uso del AnSiRE-MAI, junto con consideraciones y detalles relevantes.

Cada pestaña cuenta con teclas de acceso rápido para facilitar la navegación y mejorar la experiencia del usuario.

Menú acciones

La interfaz de usuario incluye la pestaña "Ejecutar", que activa las funciones para ejecutar el AnSiRE-MAI. En el caso de una

nueva simulación, se solicita al usuario asignar un nombre y una ubicación para guardarla.

Tras guardar el archivo, se abre automáticamente la ventana de comandos (cmd) para mostrar la ejecución y los cálculos del software. Al finalizar, los resultados se cargan directamente en la interfaz. Esta pestaña también cuenta con una tecla de acceso rápido para mayor comodidad.

Barra de herramientas

Cuenta con herramientas en el que se puede realizar algunas de las funciones mencionadas en los menús, ubicando el cursor y haciendo clic izquierdo sobre alguna de las opciones

DETALLES DE IMPLEMENTACIÓN

Los algoritmos del AnSiRE-MAI se implementaron en el entorno QT 4.9.1 utilizando el lenguaje C y el compilador MinGW 5.13.0. La generación del archivo ejecutable (.exe) se llevó a cabo en Dev-C++ con el compilador TDM-GCC 4.9.2.

La interfaz de usuario fue desarrollada con QT 4.9.1 empleando herramientas Widget. Aunque la interfaz y el software principal son programas separados, se integran internamente mediante la función `system()`.

El instalador del programa se creó utilizando la librería QT Installer Framework 4.3.0, optimizando el proceso de instalación del software. Al ejecutarlo, se abre una ventana que guía al usuario a través de los pasos necesarios para completar la instalación correctamente.

Una vez instalado, el programa ocupará aproximadamente 883.69 MB en el disco duro y creará un acceso directo en el menú de inicio para facilitar su uso.

RESULTADOS

Para la comprobación del funcionamiento del software, se trabajaron algunas redes de prueba tanto simbólicos como numéricos y los resultados obtenidos se compararon con el simulador "Multisim".⁷

El software se instaló en diversos equipos con el sistema operativo de Windows de diferentes versiones (Windows 7 y Windows 10) tanto de 32 bits como de 64 bits, operando de manera correcta.

A continuación, se muestran dos ejemplos, uno con valores numéricos y otro con valores simbólicos.

Ejemplo N°1: Red con señal senoidal.

En la Figura 7(a), se muestra el esquema de la red a implementar, en la Figura 7(b) se aprecia la interfaz gráfica con la declaración de los elementos que componen la red y la respuesta que se desea obtener, en este caso el voltaje en la resistencia r1. La red cuenta con una señal seno con offset igual a 5 V, una amplitud de 5 V, frecuencia de 60 Hz y un desfase de 120° en serie con dos resistencias de 1 kΩ, se pide hallar la diferencia de potencial en la resistencia r1.

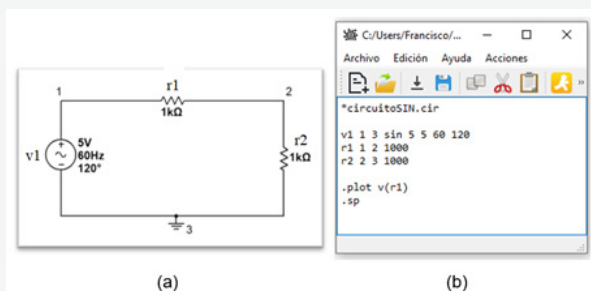


Figura 7. (a) Red con señal senoidal. (b) Netlist de la red con señal senoidal a simular.

En la Figura 8, se muestra el resultado que se obtiene en el dominio de la frecuencia al darle a la opción ejecutar en las opciones del menú.



Figura 8. Respuesta de la red con señal senoidal.

Ejemplo N°2: Red con amplificador operacional y señal cuadrada.

En la Figura 9(a), se muestra el esquema de la red a implementar, en la Figura 9(b) se aprecia la interfaz gráfica con la declaración de los elementos que componen la red y la respuesta que se desea obtener, en este caso la ganancia. La red está integrada por un amplificador operacional tipo inversor con amplitud A, frecuencia T (ambos simbólicos) offset o V y sus respectivas resistencias conectadas entre la fuente y el amplificador.

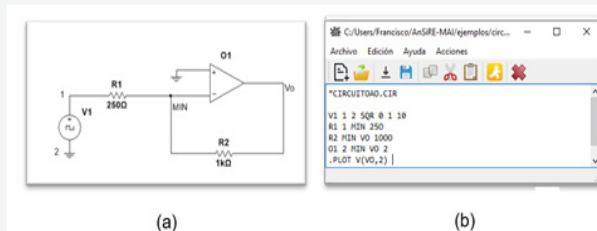


Figura 9. (a) Red con amplificador operacional y señal cuadrada. (b) Netlist con amplificador operacional y señal cuadrada a simular.

Se ejecuta el software y el resultado que se obtiene se visualiza en la Figura 10.

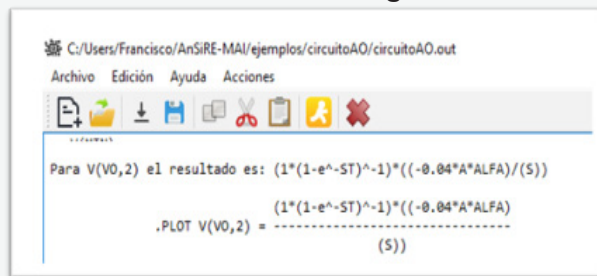


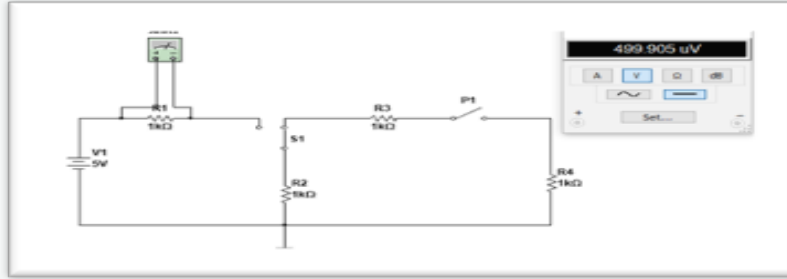
Figura 10. Resultados de la red con amplificador operaciones y señal onda cuadrada.

Ejemplo N°3: Red con interruptores de una y dos vías.

El ejemplo a describir en la interfaz se vería gráficamente como en la Figura 11 (a), en la cual se muestran los resultados que se obtienen al realizar el análisis en NI Multisim 14.0.

Se describe la red en la interfaz de AnSIRE-MAI, tal como se ve en la Figura 11. (b)

Se ejecuta el software y el resultado que se obtiene se visualiza en la Figura 12, el cual es el esperado.



(a)

```

C:/Users/Francisco/AnSiRE-MAI...
Archivo Edición Ayuda Acciones
*CIRCUITOPS.CIR
V1 1 7 DC 5
R1 1 2 1000
S1 2 3 4 5 100 10000000
R2 3 7 1000
R3 4 5 1000
P1 5 6 ON 10 100 10000000
R4 6 7 1000
.plot v(R1)
  
```

(b)

Figura 11 (a) Red con interruptores de una y dos vías en NI Multisim 14.0. (b) Netlist con interruptores de una y dos vías en AnSiRE-MAI.

```

C:/Users/Francisco/AnSiRE-MAI...
Archivo Edición Ayuda Acciones
Para V(R1) el resultado es:
0.00049990002499245
  
```

Figura 12. Resultados de la red con interruptores de una y dos vías.

Conclusiones

La ampliación de las capacidades de cálculo del AnSiRE-MAI permitió la incorporación de nuevos algoritmos, rutinas y mejoras, así como una actualización de su interfaz de usuario. Con las nuevas actualizaciones, el programa es capaz de realizar cálculos numéricos, simbólicos y mixtos de manera eficiente, manteniendo la velocidad de procesamiento y la precisión en los resultados que caracterizan al AnSiRE-MAI.

La implementación del algoritmo de interruptores fue exitosa, agregando funciones y condiciones que permiten declarar los parámetros de forma numérica, mixta o simbólica. Incluso se consideró la inclusión de la resistencia intrínseca si así lo desea el usuario.

La implementación de fuentes variables en el tiempo requirió un enfoque diferente debido a las limitaciones del analizador de expresiones del software. Sin embargo, se logró obtener un resultado satisfactorio en esta funcionalidad.

El nuevo entorno de escritura, lectura y ejecución del AnSiRE-MAI, se puede instalar en cualquier ordenador y comenzar a utilizarlo de inmediato, sin complicaciones.

Se implementaron los algoritmos, rutinas y funciones de manera estructurada y secuencial, lo que permite futuras actualizaciones y mejoras sin dificultades adicionales.

Referencias

- 1.- O.J., Ruiz Sotelo, Trabajo de Grado: Análisis Simbólico de Circuitos. Parte 1: Motor de Cálculo., Mérida: Universidad de Los Andes, 2017.
- 2.- J. J., Escalante Márquez. Trabajo de Grado: Ampliación de las capacidades de cálculo del analizador simbólico de redes AnSiRE. Mérida: Universidad de Los Andes, 2019.
- 3.- R. Spence, LINEAR ACTIVE NETWORKS, ABERDEEN: Jonh Wiley & Sons Ltd, 1970.
- 4.- J.A, González Eslava, Trabajo de Grado: Análisis Simbólico de Circuitos usando la Matriz de Admitancia Indefinida. Mérida: Universidad de Los Andes, 2022.
- 5.-J.X., Colina Tromp. Trabajo de Grado: Ampliación de las capacidades de cálculo del analizador simbólico usando la matriz de admitancia indefinida. Mérida: Universidad de Los Andes, 2022.
- 6.- L.W. Nagel LW, D.O. Pederson. Simulation Program with Integrated Circuit Emphasis. Berkeley: EECS Department, University of California;1973. Informe Nro: UCB/ERL M382. Disponible en: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/1973/ERL-382.pdf>. [Acceso: 02 febrero 2025].
- 7.-National Instruments. Multisim [software]. Versión 14. Austin, TX: National Instruments; 2025. Disponible en: <https://www.ni.com/es-cr/shop/product/multisim.html>. [Acceso: 02 junio 2025].