UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA ELÉCTRICA

SYMBULATOR

UN SIMULADOR DE CIRCUITOS LINEALES PARA CALCULADORAS

Trabajo de Graduación para optar por el título de
Licenciado en Ingeniería Electromecánica

Roberto Pérez-Franco
6-700-1101

2001

Página de Calificación del Tribunal Examinador

Profesor Medardo Logreira: *100/100*

Profesora Eliane Boulet de Cabrera: *100/100*

Profesora Marcela Paredes: 100/100

Observaciones Generales del Tribunal Calificador: *Es un excelente trabajo que provee un aporte innovador en el campo de la simulación numérica y simbólica de los circuitos eléctricos, y que ha recibido reconocimiento a nivel mundial. Este trabajo es un orgullo para la Facultad de Ingeniería Eléctrica y para la Universidad Tecnológica de Panamá. Los miembros del jurado recomiendan que se promocione este trabajo entre la comunidad Tecnológica.*

Advertencia

El autor del *Symbulator* es fundador del Movimiento PAX[1]. Es decisión del autor que este programa solamente puede ser usado para aplicaciones pacíficas, y nunca para aplicaciones bélicas. Es deber del usuario respetar esta decisión.

---

[1] El Movimiento PAX es un movimiento pacifista conformado por científicos, ingenieros y técnicos de diferentes nacionalidades, comprometidos ante ellos mismos y ante las generaciones venideras con la paz en el mundo, el progreso pacífico de la raza humana y el desarrollo sostenible de todos los pueblos. Rompe las reglas. Únete a PAX. http://pax.w3.to

Agradecimientos

A los profesores Medardo Logreira, Eliane Boulet de Cabrera y Marcela Paredes de la Universidad Tecnológica de Panamá, por apoyarme incondicionalmente en el desarrollo de este proyecto.

Deseo agradecer a todos los que me aman: Dios, por darme todo; mi esposa Mónica, por mantenerme vivo; mi madre Eka, por darme la vida; mi padre Tito, por darme dos alas amplias y fuertes, y un cielo enorme para probarlas; y mi hermana Ekita, por darme poesía.

Agradezco a mi padre la revisión gramatical y a José Vega la revisión técnica de este texto, a Miguel Latorraca la confección de las imágenes de los circuitos presentados, y a mi esposa su ayuda en la confección del índice y la lista de figuras. Estoy en deuda igualmente, con Doug Burkett y Tim Hutcheson por la buena voluntad, el esfuerzo y las muchísimas horas de su valioso tiempo que han dedicado a la traducción cuidadosa de esta tesis al inglés, bajo el título de "*The Symbulator Book*". Mi eterna gratitud a todos los amigos y usuarios que me estimularon con sus sugerencias y comentarios, para desarrollar, mejorar y probar el *Symbulator* durante estos años: Tim Hutcheson (EUA); Lars Frederiksen (Dinamarca); Joe Riel (EUA); Arne Harstad (Noruega); Erwin Baert (Bélgica); Charles Ware (EUA); Doug Burkett (EUA); Reinhard Willinski (Alemania); Kamil Malinski (Austria); Jake Adams (EUA); Daniele Martini (Italia); Rozgonyi Szabolcs (Hungría); Michael Rans (Reino Unido); Alex Astashyn (Rusia); Al Charpentier (EUA); Nevin McChesney (EUA); Ivan Oro Yu (Panamá); y Pepe Iborra (España).

Tabla de Contenido

Capítulo 4 - Simulación experta en corriente directa

Capítulo 5 - Equivalentes de Thévenin y Norton

Capítulo 6 - Simulación numérica en corriente alterna

Capítulo 7 - Bipuertos

Capítulo 8 - Simulación simbólica en corriente alterna

Lista de Figuras

Resumen

Tiene el lector en sus manos la documentación completa del *Symbulator*, el cual es un simulador simbólico de circuitos lineales para calculadoras TI. Presenta detalladamente los conceptos teóricos de la simulación en el *Symbulator*, acompañados de casi un centenar de problemas, resueltos paso a paso.

El Capítulo 1 presenta los conceptos generales que el usuario debe conocer antes de realizar cualquier simulación: los requisitos que debe cumplir la calculadora, la filosofía de la notación empleada para nombrar los nodos y describir el circuito, y los usos de comandos, herramientas y funciones.

La lectura del Capítulo 2 es imprescindible, pues éste presenta paso a paso y con todo detalle, todas las técnicas que el usuario necesita conocer para realizar simulaciones numéricas en corriente directa. Esta es la simulación básica. Se instruye al lector en el empleo de los elementos de circuito sencillos: cortocircuitos, resistencias, fuentes de corriente y fuentes de voltaje. Estas fuentes se presentan tanto en su forma independiente y también en su forma dependiente de corriente y voltaje.

El Capítulo 3 discute los aspectos teóricos y prácticos necesarios para realizar una simulación simbólica, para encontrar ya sea respuestas numéricas o respuestas simbólicas. En este tipo de simulación, el *Symbulator* realmente se destaca.

La lectura del Capítulo 4 es opcional. En él se presentan en profundidad las técnicas de simulación en Modo (Experto). Este tipo de simulación requiere de un conocimiento claro del funcionamiento interno del *Symbulator*, y permite obtener respuestas numéricas en simulaciones simbólicas, con menos esfuerzo y en menos tiempo.

El Capítulo 5 explica el uso de la herramienta thevenin, la cual permite encontrar los circuitos equivalentes de Thévenin y Norton de una red. Además de los problemas numéricos, algunos problemas simbólicos se desarrollan para ilustrar la obtención de equivelantes numéricos de redes que incluyen elementos de valor desconocido.

En el Capítulo 6 se trata la simulación en corriente alterna. Esta simulación se lleva a cabo utilizando fasores. En este capítulo se introducen nuevos elementos: la capacitancia, la inductancia, la inductancia mutua y el transformador ideal. Igualmente, se ilustran las dos maneras en que se pueden manejar los problemas con valores RMS.

El Capítulo 7 aborda, de forma completa, el tema de los bipuertos en el *Symbulator*. Describe cómo realizar simulaciones utilizando seis diferentes tipos de bipuertos. Igualmente, explica cómo obtener los parámetros, en estos seis tipos de bipuertos, para una red de dos terminales cualquiera. Numerosos ejemplos complementan la teoría.

El Capítulo 8 muestra cómo se pueden realizar simulaciones simbólicas en corriente alterna. Esta simulación, más avanzada que las anteriores, requiere de nuevos e interesantes conceptos, que son presentados claramente y sustentados con múltiples ejemplos. Ellos incluyen los equivalentes Thévenin en corriente alterna y la frecuencia de resonancia.

El Capítulo 9 está dedicado a disecar unos cuantos casos de particular interés, los cuales por su novedad o complejidad requieren de una explicación más minuciosa que el resto de los problemas. El uso de gráficas, soluciones simbólicas que requieren más de una simulación y el uso del amplificador operacional ideal, son algunos de los nuevos conceptos que se aprenderán en sus páginas.

En el Capítulo 10 aprenderá el lector a simular en el dominio de la frecuencia compleja. Múltiples ejemplos de problemas con polos y ceros, amplificadores operacionales ideales, equivalentes de Thévenin y gráficas de respuesta a la frecuencia, acompañan a los conceptos que se presentan.

El Capítulo 11 es uno de los capítulos más valiosos de este trabajo. En él se explica en detalle cómo obtener respuestas en el dominio del tiempo en el *Symbulator*. Éste es otro de los temas en los cuales el potencial del simulador se hace manifiesto. Múltiples ejemplos ilustran el uso de condiciones iniciales y la resolución de circuitos con interruptores en varias simulaciones.

En el Capítulo 12 se muestra el uso de dos herramientas que incluye el *Symbulator* para graficar respuestas: una para confeccionar gráficas de Bode y la otra para graficar funciones del tiempo. Ilustran su uso dos problemas de profesores de la Universidad Tecnológica de Panamá.

El primer apéndice trata sobre la historia del simulador; el segundo, sobre el origen autóctono de las técnicas que aparecen en esta tesis; el tercero, sobre cuándo declinar el uso del simulador; y el cuarto, presenta una lista completa de los problemas presentados, indicando su origen.

En las últimas páginas, presentamos nuestras conclusiones y damos algunas recomendaciones finales.

Introducción

El propósito general de esta tesis es instruir al lector en el hermoso y recién creado arte de la simulación simbólica de circuitos lineales. En forma particular, pretende transmitir el conocimiento y las técnicas que permitirán aprovechar al máximo el *Symbulator*, como una poderosa herramienta para el aprendizaje de teoría de circuitos.

Se asume que el lector posee ya o está obteniendo de otra fuente los conceptos básicos de teoría de circuitos. Se asume también que el lector posee una calculadora TI-89 ó TI-92Plus y su respectivo *Manual de Usuario*. No es necesario que el lector posea experiencia en el uso de esta máquina. Esta tesis debe leerse con la calculadora a la mano. Se asume que la calculadora cumple los requisitos del *Symbulator*. Léanse los capítulos cuidadosamente, en estricto orden, empezando por el primero y ejecutando todos los ejemplos.

Las dudas pueden ser presentadas al autor a través del sitio `http://www.todoexpertos.com`, buscando "Symbulator". Si se encuentra algún error, por favor repórtelo al correo electrónico: `symbulator@perez-franco.com`

El texto completo de esta tesis y la última versión del programa, pueden encontrarse en la Internet[2] en `http://perez-franco.com/symbulator`

---

[2] *Actualmente también esta dirección* `http://sq.calc.org`

Capítulo 1

Conceptos generales del *Symbulator*

1.1 ¿Qué es Symbulator?

Symbulator es un simulador de circuitos lineales, con capacidad de respuestas simbólicas y numéricas, el cual permite el análisis en corriente directa, corriente alterna, dominio de la frecuencia y análisis transitorio. Acepta una gran variedad de elementos lineales. Ocupa menos de 60KB, está programado en Basic y usa como plataforma cierto tipo de calculadoras Texas Instruments.

1.2 Requisitos

La calculadora Texas Instruments en la cual se ejecuta el *Symbulator* debe ser modelo TI-89 ó TI-92 Plus, tener instalado un *AMS* (Sistema de Matemáticas Avanzadas, es decir un Sistema Operativo) versión 2.05 y tener el programa *DiffEq* versión 2.05 correctamente archivado. También el *Symbulator* debe estar correctamente archivado. En adelante, nos referiremos a esta calculadora como “la calculadora TI”.

1.2.1 Fuentes de información para la instalación

Si su calculadora no cumple alguno de estos requisitos, debe actualizarla antes de proseguir. Como esta tesis trata del *Symbulator*, no de la operación de calculadoras TI, no entraremos en detalles sobre cómo realizar esta actualización. Listamos a continuación las fuentes de información que pueden ser útiles:

Para actualizar el Sistema Operativo de una calculadora TI, consúltese la documentación que ofrece el fabricante en su sitio en la Internet: http://education.ti.com.

Los programas *Symbulator* y *DiffEq* pueden obtenerse gratuitamente en el sitio oficial del Symbulator, cuya dirección se presenta en el punto *1.3*. Para instalar en la calculadora estos programas se debe utilizar el programa de computadora ***Graph-Link***, producido por TI, el cual se puede obtener gratuitamente en la dirección que hemos presentado en el párrafo anterior.

1.2.2 Fuentes de información para archivar los programas

Los programas se ejecutan más rápidamente en la calculadora TI cuando están correctamente archivados en la memoria *Flash ROM*. Tanto el *Symbulator* como el *DiffEq* poseen un programa para archivarse automáticamente. Instrucciones detalladas sobre cómo ejecutarlo se incluyen en la documentación que los acompaña cuando se les baja del sitio oficial

1.3 Sitio en la Internet

El *Symbulator* posee un sitio oficial en la Internet en http://sq.calc.org. Este sitio está en inglés. Aparte de la documentación completa y muchos problemas resueltos paso a paso, también se puede encontrar ahí la versión más reciente del programa, de forma gratuita.

1.4 Concepto del algoritmo

Para analizar un circuito eléctrico, el *Symbulator* utiliza un método que emula las técnicas manuales de análisis de circuito que aprenden los estudiantes en el salón de clases. De hecho, el simulador se comporta como un excelente estudiante que prestó atención y aprendió su lección. Como el programa sigue un camino reproducible por el estudiante, es también una herramienta de aprendizaje muy poderosa.

El *Symbulator* conoce la ley de Ohm-Cavendish, y la ley de las corrientes de Kirchhoff. Con estas dos leyes, el programa analiza los circuitos utilizando solamente el **análisis de nodos**. Aunque no conoce la ley de los voltajes de Kirchhoff, siempre la cumple, porque esta ley se cumple automáticamente al cumplir las dos anteriores. El programa nunca utiliza análisis de mallas. Gracias al poderoso programa de matemática avanzada de la calculdadora TI, el análisis de nodos es suficiente para todos los casos, aún los más complejos.

El *Symbulator* también conoce los teoremas de Thévenin-Helmholtz y Norton-Mayer. Y sabe realizar gráficas de Bode. Para poder realizar la transformada y la transformada inversa de LaPlace, el *Symbulator* utiliza el programa *DiffEq*, que es el programa de transformada de LaPlace más rápido jamás escrito para una calculadora, fruto de la mente genial del programador danés Lars Frederiksen, a quien el autor tiene el orgullo de contar entre sus amigos.

El *Symbulator* genera las ecuaciones de cada nodo y cada fuente de voltaje, así como ecuaciones especiales para elementos especiales, y al mismo tiempo va identificando las incógnitas primarias del circuito. Al final, encuentra el valor (ya sea simbólico o numérico) de todas estas incógnitas primarias resolviendo las ecuaciones, y luego obtiene a partir de las primarias los valores de las incógnitas secundarias y terciarias. Así, al final ofrece toda la información sobre el circuito que un estudiante pudiese desear.

1.5 Puertas

El corazón del *Symbulator* está compuesto por varios archivos llamados *step#,* los cuales se encargan de realizar la simulación. El usuario, sin embargo, nunca toca directamente estos archivos, sino que utiliza diferentes programas que actúan como *puertas* de acceso que lo conducen a esos otros programas indirectamente. Existen cuatro puertas de acceso, una para cada tipo de análisis: 1) para corriente directa se usa ***dc***, 2) para corriente alterna se usa ***ac***, 3) para dominio de la frecuencia se usa ***fd***, y 4) para análisis transitorio se usa ***tr***.

1.6 Herramientas

Existen también ocho herramientas, que son auxiliares ya sea para la simulación de circuitos o bien para el entendimiento de los resultados de las simulaciones: 1) para reducir resistencias en paralelo se usa ***par***, 2) para mostrar respuestas complejas en forma polar se usa ***absang***, 3) para graficar una función del tiempo se usa ***plot***, 4) para dibujar gráficos de Bode se usa ***bode***, 5) para encontrar el equivalente Thévenin de un circuito se usa ***thevenin***, 6) para encontrar los parámetros equivalentes de bipuerto de un circuito se usa ***port***, 7) para encontrar la ganancia de un amplificador se usa ***gain***, y 8) para llevar una respuesta del dominio de la frecuencia al dominio del tiempo se usa ***fd_to_tr***. En su momento, veremos las capacidades y la notación propia de cada una de estas puertas y herramientas.

Para una referencia rápida, el *Symbulator* incluye un file de ayuda breve. Basta escribir `scs\help()` en la calculadora y el menú de ayuda aparecerá. Esta ayuda está en inglés.

1.7 Comandos

A veces, para resolver un problema usando el *Symbulator*, el usuario debe usar comandos propios de la calculadora, tales como los comandos para resolver ecuaciones, o funciones de programas auxiliares como las de transformada de Laplace del programa *DiffEq*.

1.8 Sobre la notación

1.8.1 Filosofía

La filosofía de la notación empleada por el *Symbulator* para describir un circuito se basa en el principio de que el usuario debe proporcionar al simulador sólo el mínimo de información necesaria para describir completamente cada elemento del circuito.

1.8.2 Descripción textual

Dado que es un programa diseñado para ser tan veloz como lo requiere su uso en el salón de clases, la descripción del circuito se hace con texto. No se utiliza una interfaz gráfica, pues recargaría de trabajo al procesador de la calculadora, el cual es un Motorola 68000 de 12MHz, y haría lento el proceso de descripción del circuito.

La descripción del circuito es una cadena textual de elementos, compuestos por una fila para cada elemento del circuito, casi siempre de cinco columnas. El primer término de cada fila es el nombre del elemento, el cual debe empezar con una letra que indentifique al elemento que aquella fila representa. Luego se colocan los nombres de los nodos a los cuales está conectado el elemento, pudiendo ser estos nombres tanto números como letras. Finalmente se coloca el valor del elemento, en su unidad de medida correspondiente, el cual puede ser un valor numérico, simbólico, o incluso una expresión algebraica. En el caso de elementos que almacenan energía (o sea inductores y capacitores) con condiciones iniciales, éstas se colocan al final de la fila como último elemento en la descripción.

Los elementos de cada fila se separan entre sí mediante comas “ **,** ”. El límite entre una fila y otra, es decir entre la descripción de un elemento y la de otro, se indica por medio de punto-coma “ **;** ”. Ejemplo, para un circuito de tres elementos y cinco términos por fila:

```
"a1,b1,c1,d1,e1;a2,b2,c2,d2,e2;a3,b3,c3,d3,e3"
```

Internamente, el *Symbulator* convertirá esta descripción a una matriz, para trabajar rápidamente con ella.

1.8.3 Nodos

1.8.3.1 Orden de los nodos

El orden en que se introducen los nombres de los nodos en la descripción de un elemento no es aleatorio. Este orden indica la dirección en la cual se está definiendo el flujo de la corriente y la caída de voltaje en ese elemento. Esto es especialmente importante en el caso de fuentes, de elementos que controlan fuentes dependientes, de elementos almacenadores de energía con condiciones iniciales y de inductancias con inductancia mutua.

1.8.3.2 Nombres de los nodos

Antes de correr cualquier simulación, el usuario debe identificar a cada uno de los nodos y de los elementos del circuito con un nombre único.

En el caso de los nodos, el nombre puede ser un número, una letra, o una combinación de ambos. Cada nodo debe recibir un solo nombre. Esto quiere decir que, cada vez que se mencione este nodo en la descripción del circuito, se le debe nombrar de la misma manera.

Es normal que aquellos usuarios que no están acostumbrados a usar un simulador de circuitos, olviden al momento de ver las respuestas el nombre que le han dado a algún nodo o elemento. Por eso, mientras se adquiere experiencia, es una práctica recomendable anotar los nombres en algún papel, de preferencia sobre el dibujo mismo del circuito, al lado de cada nodo y elemento.

1.8.3.3 Nodo de referencia

Siempre debe existir un nodo que sirva de referencia. Para esta función, su voltaje siempre es de 0 voltios. Este nodo se conoce como *tierra* (o *ground*) y debe identificarse siempre con el nombre *0* - es decir, cero.

1.8.4 Variables reservadas

El sistema operativo de la calculadora mantiene algunas variables ocupadas contínuamente para su uso interno propio. Estas variables se conocen como variables reservadas, pues la máquina las reserva para sí misma. Una lista completa de estas variables reservadas puede encontrarse al final del Manual del Usuario de la calculadora. Vale mencionar aquí aquellas que un usuario del *Symbulator* podría intentar usar en un descuido: *nc*, *ok*, *rc*, *sx*, *tc*, *yc*, *zc*, y aquellas entre *c1* y *c99*.

El *Symbulator*, a su vez, utiliza muchas variables internas para propósitos de realizar la simulación. Estas variables son muchas, y se generan de acuerdo a los nombres de los elementos del circuito, lo que hace imposible listarlas de antemano. Sin embargo, tienen algo en común: todas empiezan con letras griegas. Esto hace que resulte una tarea fácil evitar su uso accidental; basta con recordar no usar variables que empiecen con letras griegas.

Capítulo 2
Simulación numérica en corriente directa

2.1 Antes de ejecutar cualquier simulación

Dado que el *Symbulator* entregará al usuario las respuestas de la simulación en forma de variables almacenadas en la *carpeta actual*, es una práctica saludable limpiar carpeta de variables innecesarias antes de simular. Así se evitarán las confusiones que podrían producirse al reposar en la carpeta actual tanto las respuestas de la última simulación como las respuestas viejas de simulaciones anteriores. Se recomienda, entonces, borrar todas las variables innecesarias de la carpeta actual antes de cada simulación. Aunque no es obligatorio el hacerlo, esta práctica puede facilitar al principiante la lectura de las respuestas generadas en la última simulación.

2.1.1 Carpeta actual

La *carpeta actual* es la carpeta en la cual está operando la calculadora en un momento dado. Usualmente, la *carpeta actual* es la carpeta principal de la calculadora, llamada MAIN. El usuario puede usar cualquier otra carpeta si lo desea. Pero a menos que existan razones específicas para lo contrario, se recomienda usar siempre la carpeta MAIN.

Una manera de saber cuál es la *carpeta actual* en un momento dado, es mirar la esquina inferior izquierda de la pantalla. Ahí, en letras pequeñas, se muestra el nombre de la *carpeta actual*.

2.1.2 Variables innecesarias

Las *variables innecesarias* son todas aquellas variables cuya presencia en la *carpeta actual* no es necesaria para la ejecución de la simulación. Un típico ejemplo de este tipo de variables son aquellas respuestas generadas por el *Symbulator* en una simulación anterior.

2.1.3 Procedimiento

Un usuario familiarizado con el uso de la calculadora TI sabrá borrar las variables innecesarias. Una explicación completa se puede encontrar en el Manual de Usuario de la calculadora. De todas formas, he aquí una explicación detallada. Para borrar las variables innecesarias, complete los siguientes pasos:

a) Identifique cuál es la *carpeta actual*. Para identificarla, basta con mirar la esquina inferior izquierda de la pantalla de la calculadora. Obviamente, la calculadora debe estar encendida. Si no ve ningún nombre en esa ubicación, presione la tecla " y vuelva a mirar. Como lo sabe bien todo usuario de la calculadora TI, al presionar esa tecla hacemos aparecer la *pantalla hogar* (Home Screen).

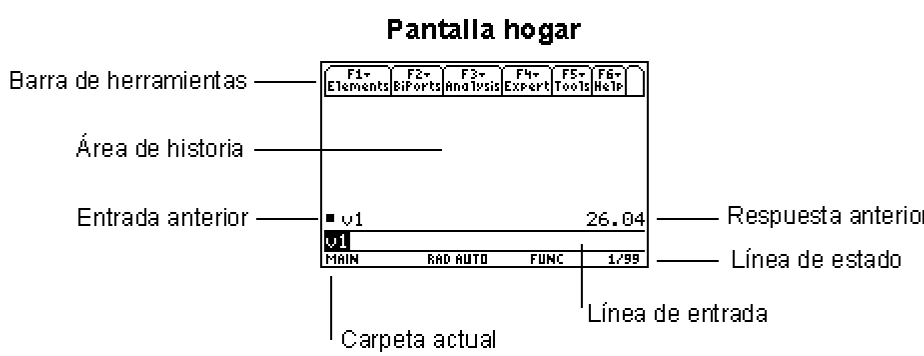

Figura 1. Pantalla hogar, en la TI-89

La pantalla hogar está compuesta de varias partes: la parte superior se conoce como la *barra de herramientas* (Toolbar). La parte blanca del medio se conoce como el *área de* *historia* (History Area), y es ahí donde aparecen las entradas anteriores y las respuestas anteriores. El renglón siguiente se conoce como la *línea de entrada* (Entry Line), y es ahí donde se escriben las órdenes para la calculadora. Y la última franja, que es una delgada cinta llena de letras y números, se conoce como la *línea de estado* (Status Line), en donde aparece el estado de la TI. El primer dato, pegado a la izquierda de la pantalla, es la *carpeta actual*. Usualmente dirá MAIN.

b) **Entre al *Var-Link*.** *Var-Link* es el nombre que recibe el entorno en el cual se pueden inspeccionar y manipular las variables que la calculadora tiene almacenados en las diferentes carpetas en la memoria. Veamos cómo podemos entrar a ese entorno. Presione la tecla **2** y luego la tecla **|**. Tenga la precaución de no confundir la tecla **|** con la tecla **·**.

c) **Entre a la *carpeta actual*.** En la pantalla aparecerán las diferentes carpetas que tiene la calculadora. En el teclado encontrará cuatro flechas azules, que son las flechas para desplazar el cursor por la pantalla. Presione las flechas azules de arriba y abajo hasta destacar con el fondo negro el nombre de la *carpeta actual*. Si el nombre de la carpeta tiene una flecha al lado, significa que contiene variables. Si por el contrario el nombre de la carpeta no tiene nada al lado, significa que la carpeta está limpia. Si está limpia, no hay que hacer nada más. Si contiene variables, debemos proseguir. Habiendo destacado el nombre de la carpeta, presione la tecla azul de la derecha, para abrir la carpeta.

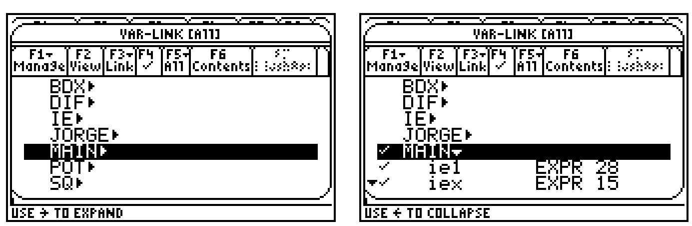

Figura 2. Entorno Var-Link, en la TI-89, antes (izq.) y después (der.) de abrir MAIN

d) Borre las variables innecesarias. Moviéndose con las flechas de arriba y abajo, destaque el nombre de alguna variable que considere innecesaria. Presione la tecla

† . Un pequeño gancho aparecerá al lado del nombre de la variable. Marque así con ganchos todas las variables en la carpeta que considere innecesarias. Una vez haya hecho esto, presione la tecla 0 . Recibirá un mensaje (en inglés) pidiéndole que confirme si desea borrar las variables seleccionadas. Para borrarlas, presione la tecla ¸ .

En el caso de que se desee borrar la carpeta entera, puede seleccionarse la carpeta misma presionando † y luego presionando la tecla 0 . Si la *carpeta actual* es la carpeta MAIN, se borrará todo su contenido, sin borrarse la carpeta misma. Esta carpeta no se puede borrar, por ser la principal de la calculadora. Recibirá un mensaje de error anunciándole que esta carpeta no se puede borrar. Pero habrá logrado su cometido: limpiar de variables innecesarias la *carpeta actual*. Este es el procedimiento más recomendable, pues es rápido y eficaz. Debe tenerse la precaución de no almacenarse en MAIN variables importantes que se deseen preservar.

2.2 Primer paso universal

El *Symbulator* ofrece cuatro diferentes análisis para simular un circuito. Cada análisis hace uso de una *puerta* distinta y opera con una lógica diferente. Sin embargo, todos ellos tienen algo en común: requieren que el estudiante sepa describir correctamente los elementos y nodos que componen el circuito. Así, el primer paso para todos los análisis es:

Nómbrense todos los nodos y elementos del circuito.

En el **Capítulo 1** explicamos la manera de nombrar a los nodos. La manera de nombrar a los elementos será explicada en diferentes capítulos, a medida que se vayan presentando estos elementos. Todos estos nombres son únicos, y una vez que son asignados, el usuario debe ser consistente en su uso.

2.3 Puerta para corriente directa: dc

Como dijimos, *puerta* es el nombre que recibe un programa que sirve de acceso a otro. Para realizar una simulación en corriente directa, se usa la *puerta* llamada **sq\dc**.

Las letras **dc** son el nombre de la puerta. Las letras **sq** le indican a la calculadora en qué carpeta se encuentra el programa **dc**. Así, al escribir **sq\dc** se le está indicando a la calculadora: “Ejecuta el programa **dc** que está en la carpeta **sq**”.

2.4 Dato de entrada: el circuito

Al utilizar ésta o cualquiera otra de las cuatro puertas, es necesario suministrarle al programa el circuito que se quiere simular, a manera de dato de entrada. El circuito debe colocarse justo a continuación del nombre de la puerta, entre paréntesis redondos, así: **sq\dc(**circuito**)**.

Hay dos maneras de hacerlo. La primera manera es colocando entre los paréntesis el texto que describe al circuito, es decir:

```
sq\dc("descripción del circuito")
```

La segunda manera es colocando entre los paréntesis el nombre de alguna variable en la cual se haya almacenado el texto que describe al circuito, es decir:

```
"descripción del circuito" -> variable

sq\dc(variable)
```

Ambas funcionarán igualmente bien. La segunda tiene la ventaja de que el circuito es almacenado y se conserva en la memoria bajo el nombre de una variable, lo que permite reutilizarlo más tarde si se le desea modificar o simular nuevamente.

2.5 Respuestas

2.5.1 Filosofía

La filosofía del *Symbulator* es darle al usuario como datos de salida todas las respuestas que pudiese desear. Estas respuestas no se muestran automáticamente, sino que se almacenan en variables con nombres intuitivos. La idea detrás de esto es no abrumar al usuario con la enorme cantidad de respuestas que se generan en cada simulación. El usuario solicitará al programa únicamente aquellas respuestas que le interesen.

En el caso de la simulación de corriente directa, las respuestas son los voltajes, las corrientes y las potencias. Veamos:

2.5.2 Voltajes en los nodos

El *Symbulator* nos entrega los **voltajes** en todos los nodos con respecto al nodo de referencia, en voltios (V). Estos voltajes se almacenan en variables cuyo nombre es ***v*** más el nombre del nodo. Por ejemplo, si el nodo se llama ***q***, el nombre de la variable en la cual se almacena el voltaje del nodo es ***vq***.

2.5.3 Caídas de voltaje en los elementos

El *Symbulator* nos entrega las **caídas de voltajes** a través de todos los elementos, definidas del primer nodo en la descripción con respecto al segundo nodo, en voltios (V). Estas caídas de voltaje se almacenan en variables cuyo nombre es ***v*** más el nombre del elemento. Por ejemplo, si el elemento se llama ***r5***, el nombre de la variable en la cual se almacena la caída de voltaje a través del elemento es ***vr5***.

2.5.4 Corrientes

El *Symbulator* nos entrega las **corrientes** a través de todos los elementos, definidas a través del elemento, desde el primer nodo en la descripción, hacia el segundo nodo, en amperios (A). Estas corrientes se almacenan en variables cuyo nombre es ***i*** más el nombre del elemento. Por ejemplo, si el elemento se llama ***r5***, el nombre de la variable en la cual se almacena la corriente a través del elemento es ***ir5***.

2.5.5 Potencias

El *Symbulator* nos entrega las **potencias** reales <u>consumidas</u> por todos los elementos, en vatios o watts (W). Estas potencias se almacenan en variables cuyo nombre es ***p*** más el nombre del elemento. Por ejemplo, si el elemento se llama ***r5***, el nombre de la variable en la cual se almacena la potencia real consumida por el elemento es ***pr5***. Nótese que, como la potencia se define cual si fuese consumida, en los casos de un elemento que entrega potencia en vez de consumirla - como es el caso de la mayoría de las fuentes - la potencia en la respuesta será negativa.

2.6 Cortocircuito

Idealmente, un cortocircuito es un conductor perfecto, que transmite toda la corriente que desee atravesarlo sin producir caída de voltaje ni consumir potencia.

2.6.1 Notación

Desde el punto de vista de la simulación, el cortocircuito es el elemento de circuito más sencillo que existe. Sin embargo, es el elemento menos usado. Empezamos con él por ser el más fácil de describir para el usuario.

La filosofía de la notación es darle al simulador la mínima información necesaria para describir el elemento completamente. Cuando el elemento que queremos describir es un cortocircuito, la mínima información necesaria para describirlo por completo es la siguiente:

1. *¿Qué clase de elemento es?* Es un cortocircuito. Al *Symbulator* podemos señalarle qué clase de elemento estamos describiendo mediante la *primera letra* del nombre que le demos al elemento. La letra que identifica a los cortocircuitos es la **s**. Es decir, si la primera letra del nombre del elemento es una **s**, entonces el *Symbulator* comprenderá que el elemento que estamos describiendo es un cortocircuito.

2. *¿Cómo se llama el elemento?* Dado que un circuito puede tener varios elementos de la misma clase, debemos darle un nombre particular a cada elemento, para poder distinguirlo. Así pues, si un circuito tiene dos cortocircuitos, debemos darle a cada uno un nombre diferente. Eso sí: ambos nombres deben empezar con **s**, porque ambos representan a cortocircuitos. Por ejemplo, podemos nombrar a uno de ellos **s1** y a otro **s2**. Otra alternativa sería nombrar a uno **sa** y a otro **sb**. Es válido cualquier nombre que empiece con **s** y que esté compuesto por números y/o letras. Una vez que el usuario ha definido

el nombre de un elemento, el *Symbulator* usará este nombre para todas las respuestas relacionadas a este elemento.

3. *¿Dónde se encuentra conectado el elemento?* Un cortocircuito tiene dos extremos, cada uno de ellos conectado a un nodo distinto. Es necesario darle al simulador los nombres de los dos nodos a los cuales está conectado el cortocircuito. En el *Symbulator*, el orden en que se presentan los nodos siempre indica la dirección en que se define la corriente del elemento. En el caso de un cortocircuito, la corriente se define como fluyendo a través del elemento desde el primer nodo hacia el segundo nodo de la descripción.

Hemos mencionado anteriormente que cada elemento del circuito se describe como una fila de términos en la descripción del circuito. La información que hemos identificado sobre estas líneas como la mínima necesaria debe ser arreglada en la fila para introducirla en la descripción del circuito. Así, los cortocircuitos se definen como se muestra a continuación:

s*nombre, primer nodo, segundo nodo*

Esta fila condensa, en forma sencilla, la mínima información necesaria para describir el cortocircuito. Nótese que sólo hay tres elementos en la fila: un nombre que empieza con **s**, el primer nodo y el segundo nodo. Hay un cuarto elemento de información en esa fila, que no es evidente a primera vista: el orden en que se presentan los nodos contiene información sobre la dirección en que el usuario desea que el simulador defina el flujo de la corriente a través del elemento. Implícita en la fila, está la orden: "*Define la corriente del cortocircuito* ***s****nombre fluyendo a través del elemento desde el primer nodo hacia el segundo nodo.*"

2.6.2 Simetría de la descripción

Una vez que inicie la simulación, el texto de la descripción del circuito será convertido por el Symbulator en una matriz. Al igual que cualquier otra computadora para aplicaciones matemáticas, las calculadoras TI no permiten que una matriz sea asimétrica. Es decir, al introducir una matriz, todas las filas deben tener el mismo tamaño.

Si una fila tiene *n* términos, entonces todas la filas deben tener *n* términos. De lo contrario, se producirá un error.

Hemos mencionado que para describir un cortocircuito se requiere una fila de tres términos. Otros elementos de circuito, tales como las fuentes, requieren una fila de cuatro términos. Y otros, tales como los elementos almacenadores de energía, requieren una fila de cinco términos.

Como todas las filas deben tener necesariamente el mismo número de términos, se hace necesario identificar cuál fila es la que tiene más términos. Seguidamente, hay que rellenar con ceros (0) las filas que tengan menos términos que esa fila mayor.

Por ejemplo, en una descripción de circuito cuya fila mayor tiene cuatro términos, a una fila que describa un cortocircuito habrá que agregarle un cero al final para convertirla en una fila de cuatro términos:

s*nombre, primer nodo, segundo nodo, 0*

Otro ejemplo. En una descripción de circuito cuya fila mayor tiene cinco términos, a la fila que describa a un cortocircuito se le deben agregar dos ceros:

s*nombre, primer nodo, segundo nodo, 0, 0*

En el caso de las filas que describen otros tipos de elementos de circuito, en ocasiones también se presenta la misma necesidad de agregar ceros al final de las descripciones para mantener la simetría de la matriz interna.

2.6.3 Respuestas relacionadas

Hemos dicho que el voltaje de cada nodo se entrega como respuesta al final de cada simulación, almacenado en una variable. Igualmente, cada elemento del circuito generará respuestas relacionadas a él, que serán almacenadas en variables con nombres intuitivos.

En el caso de un cortocircuito, se entrega una sola respuesta relacionada: la **corriente** a través de este elemento. Como la caída de voltaje a través del cortocircuito es nula, igualmente es nula la potencia consumida por él. Por lo tanto, estas variables no se entregan como respuesta.

2.7 Resistencia

Idealmente, una resistencia es un conductor imperfecto, que se opone en mayor o menor grado al flujo de la corriente que desee atravesarlo, produciendo una caída de voltaje y consumiendo potencia.

2.7.1 Notación

En el caso de una resistencia, la mínima información necesaria para describirla por completo es la siguiente:

1. *¿Qué clase de elemento es?* Es una resistencia. La letra que identifica a las resistencias es la **r**.

2. *¿Cómo se llama la resistencia?* Para diferenciar una resistencia dada de cualquier otra resistencia, se le debe dar un nombre cualquiera. Basta con que este nombre empiece con **r**.

3. *¿Dónde se encuentra conectada la resistencia?* Una resistenca tiene dos extremos, cada uno de ellos conectado a un nodo distinto. Es necesario darle al simulador los nombres de los dos nodos a los cuales está conectada la resistencia, en el orden adecuado que indique la dirección en la cual deseamos definir la corriente y la caída de voltaje de la resistencia. La corriente se definirá como fluyendo a través del elemento desde el primer nodo hacia el segundo nodo de la descripción. La caída de voltaje se definirá como del primer nodo con respecto al segundo nodo de la descripción.

4. *¿Cuánto vale la resistencia?* Una resistencia tiene un valor. Este valor debe entregarse al *Symbulator* en ohmios. En el caso del análisis de corriente directa, el valor de una resistencia puede ser un número real o una variable.

Así, las resistencias se definen como se muestra a continuación:

```
rnombre, nodo 1, nodo 2, valor en ohmios
```

Nótese que la fila que describe a la resistencia está compuesta de cuatro términos. En este caso, también se utiliza un cero al final en el caso de que alguna otra fila en la matriz del circuito tenga cinco términos. Ejemplo:

```
rnombre, nodo 1, nodo 2, valor, 0
```

Debe tenerse precaución de no introducir el valor de la resistencia en una unidad distinta a los ohmios (Ω). Por ejemplo, si el valor de la resistencia se introduce en kiloohmios (kΩ), las respuestas estarán equivocadas.

En el caso de otros análisis distintos a la corriente directa, los valores pueden ser fasores, funciones de la frecuencia o funciones del tiempo.

2.7.2 Conductancia

Una conductancia y una resistencia son el mismo elemento analizado desde dos perspectivas distintas. La resistencia se analiza desde el punto de vista de qué tanto se opone al flujo de la corriente, y se mide en ohmios. La conductancia se analiza desde el punto de vista de qué tanto permite el flujo de la corriente, y se mide en siemens o en mhos.

Si tenemos un valor de resistencia y queremos convertirlo en un valor de conductancia, basta con invertirlo. Para invertir un valor, se le puede elevar a –1, o se puede dividir 1 entre el valor. Por ejemplo, una resistencia de 5 ohmios, equivale a una conductancia de $5^{-1} = 0.2$, o también $1/5 = 0.2$ siemens.

Para efectos de la simulación, las conductancias deben ser simuladas como resistencias. Cuando se introduce el valor, por ende, debe introducirse en ohms, no en siemens. La calculadora TI es tan avanzada que permite que la conversión se haga en la misma fila de la descripción, así:

```
rnombre, nodo 1, nodo 2, 1/(valor en siemens)
```

O así:

`rnombre, nodo 1, nodo 2, (valor en siemens)`$^{-1}$

2.7.3 Respuestas relacionadas

En el caso de una resistencia, se entregan tres respuestas relacionadas: la **corriente** a través de la resistencia, la **caída de voltaje** en la resistencia y la **potencia real consumida** por la resistencia. La corriente se almacena en una variable llamada `i`*nombre*, donde *nombre* es el de la resistencia. La caída de voltaje se almacena en una variable llamada `v`*nombre*, donde *nombre* es el de la resistencia. La potencia real consumida, por su parte, se almacena en una variable llamada `p`*nombre*, donde *nombre* es el de la resistencia.

2.8 Fuente de corriente

Idealmente, una fuente de corriente es un elemento que forza un flujo definido de corriente a través de él, y entrega o consume potencia.

2.8.1 Notación

En el caso de la fuente de corriente, la mínima información necesaria para describirla por completo es la siguiente:

1. *¿Qué clase de elemento es?* Es una fuente de corriente. La letra que identifica a las fuentes de corriente es la **j**.

2. *¿Cómo se llama la fuente?* Para diferenciar una fuente de corriente dada de cualquier otra fuente de corriente, se le debe dar un nombre cualquiera. Basta con que este nombre empiece con **j**.

3. *¿Dónde se encuentra conectada la fuente?* Una fuente de corriente tiene dos extremos, cada uno de ellos conectado a un nodo distinto. Es necesario darle al simulador los nombres de los dos nodos a los cuales está conectada la fuente, en el orden adecuado que indique la dirección en la cual deseamos definir la corriente de la fuente: fluyendo a través del elemento desde el primer nodo hacia el segundo nodo de la descripción. En un dibujo de un

circuito, la dirección en que debemos definir la fuente nos la indica la dirección en que apunta la flecha de la fuente. Este orden en los nodos definirá también, indirectamente, la polaridad de la variable de caída de voltaje en la fuente: del primer nodo con respecto al segundo nodo de la descripción.

4. *¿Cuánto vale la fuente?* Una fuente de corriente tiene un valor. Este valor debe entregarse al *Symbulator* en amperios. En el caso del análisis de corriente directa, el valor de una fuente de corriente independiente puede ser un número real o una variable. Si la fuente es dependiente, su valor debe ser una expresión algebraica. Esta expresión estará compuesta por un número y una variable. Esta variable será una de las futuras respuestas de la simulación. Los ejemplos ilustrarán este punto más claramente.

Así, las fuentes de corriente se definen como se muestra a continuación:

***j**nombre, nodo 1, nodo 2, valor en amperios*

El valor de la fuente de corriente es captado por el simulador considerándolo como fluyendo - a través de la fuente - desde el nodo 1 hacia el nodo 2. Por ejemplo, si al definir una fuente de corriente, el valor que se introduce es *10*, el simulador considerará que a través de la fuente fluye una corriente de 10 amperios desde el nodo 1 hacia el nodo 2.

Nótese que la fila que describe a la fuente de corriente está compuesta de cuatro términos. En este caso, también se utiliza un cero al final en el caso de que alguna otra fila en la descripción del circuito tenga cinco términos. Ejemplo:

***j**nombre, nodo 1, nodo 2, valor, 0*

Debe tenerse la precaución de no introducir el valor de la fuente en una unidad distinta a los amperios (A). Por ejemplo, si la corriente se introduce en miliamperios (mA), las respuestas estarán equivocadas.

En el caso de otros análisis distintos a la corriente directa, los valores pueden ser fasores, funciones de la frecuencia o funciones del tiempo.

El que las fuentes dependientes se definan usando la misma notación que las independientes es una de las grandes ventajas del *Symbulator* con respecto a otros simuladores, los cuales requieren de nombres y técnicas especiales que usan nodos o ramas de control para definir este tipo de fuentes.

2.8.2 Respuestas relacionadas

En el caso de una fuente de corriente, se entregan tres respuestas relacionadas: la corriente a través de la fuente, la caída de voltaje en la fuente y la potencia real consumida por ella. La corriente se almacena en una variable llamada `i`*nombre*, donde *nombre* es el de la fuente. La caída de voltaje se almacena en una variable llamada `v`*nombre*, donde *nombre* es el de la fuente. La potencia real consumida, por su parte, se almacena en una variable llamada p*nombre*, donde *nombre* es el de la fuente.

En primera instancia, puede parecer algo inútil que la corriente de una fuente de corriente se entregue como respuesta. Podía decirse que esta corriente debe ser conocida desde antes de la simulación, y el entregarla como respuesta no aporta nada nuevo al usuario. Esto es cierto en el caso de los simuladores numéricos. Sin embargo, es diferente en el caso del *Symbulator*, el cual ofrece capacidades 100% simbólicas, pues la corriente de una fuente de corriente puede ser una variable desconocida en el momento en que se ejecuta la simulación. Por ello, la corriente podría ser desconocida, lo que la convertiría en una respuesta muy importante para el usuario. Esa es la razón por la cual el *Symbulator*, en todas las simulaciones, la entrega almacenada en una variable.

Dependiendo de la configuración de un circuito, una fuente podría consumir potencia. Sin embargo, lo más frecuente es que una fuente entregue potencia. Hemos dicho arriba que la potencia que se da como respuesta es la potencia consumida, o sea el negativo de la potencia entregada. Es importante tener esto en mente cuando se interpreten estas respuestas.

Un ejemplo. Si la respuesta en la variable `p`*nombre* tiene un valor de –10, esto significa que la fuente consumió –10 watts. Esto es igual a decir que la fuente entregó 10 watts. Otro ejemplo. Si, por el contrario, p*nombre* tiene un valor de 15, esto significa que la fuente consumió 15 watts, lo que equivale a decir que entregó –15 watts.

Problema N° 001

Se resolverá un primer problema. Como éste es el primero, se explicará en detalle todo el procedimiento. En los siguientes problemas, las explicaciones se harán menos necesarias, y se irán omitiendo paulatinamente.

Planteamiento. Dado el siguiente circuito, obtenga los voltajes de todos los nodos.

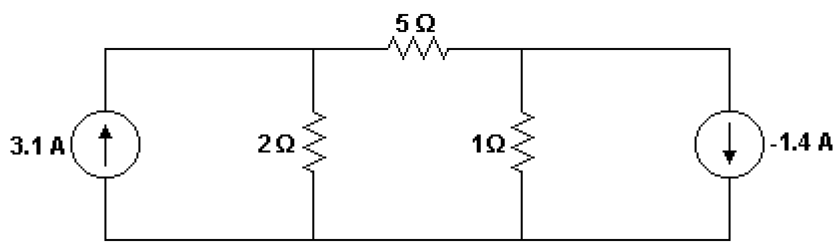

Figura 3. Circuito para el Problema N° 001.

Solución:

El **primer paso** es universal, y consiste en nombrar todos los nodos y elementos del circuito. En el circuito mostrado arriba hay tres nodos y cinco elementos. *Universal* significa que se aplica a todas las simulaciones, independientemente del análisis empleado.

De los tres nodos, uno de ellos debe forzosamente declararse como nodo de referencia. La práctica usual para la elección del nodo de referencia es escoger aquel nodo que tenga más fuentes conectadas a él. En verdad, cualquier nodo puede servir como nodo de referencia. Tomaremos para este problema el nodo inferior como nodo de referencia. Por lo tanto, le damos el nombre 0. A los otros dos nodos, les daremos los siguientes nombres: 1 al de la izquierda y 2 al de la derecha.

A los elementos les daremos los siguientes nombres: a la resistencia de 2 ohmios le llamaremos r1; a la resistencia de 5 ohmios le llamaremos r2; a la resistencia de 1 ohmio le llamaremos r3; a la fuente de corriente de 3.1 amperios le llamaremos j1; y a la fuente de corriente de –1.4 amperios le llamaremos j2. Los nodos de cada fuente de corriente deben introducirse en la descripción en el orden tal que indiquen la dirección de flujo de corriente que indica la flecha de esa fuente en el dibujo. Los nodos de las resistencias pueden introducirse en el orden que el usuario desee.

Veamos una figura del mismo circuito mostrado arriba, dibujado en una forma un poco más sencilla, y con los nombres de nodos y elementos colocados en el dibujo.

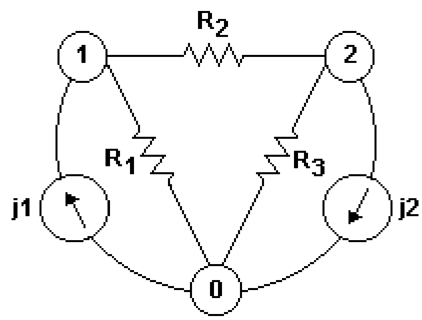

Figura 4. Circuito simplificado para el Problema N° 001.

Así, culmina el primer paso.

El **segundo paso** es escribir el texto que describe al circuito y ordenar la simulación. Este paso podría dividirse en dos pasos: primero escribir el texto, almacenándolo en una variable, y luego ordenar la simulación con esta variable. Sin embargo, hemos preferido realizar estos dos pasos como uno solo. Usualmente, resulta más rápido fundir varios pasos en uno.

La simulación se ordena escribiendo en la *línea de entrada* de la calculadora la orden **sq\dc** seguida de la descripción del circuito entre paréntesis. La calculadora interpreta esta orden como: *"Ejecuta el programa **dc** que está en la carpeta **sq**, utilizando como dato de entrada este **circuito** que te estoy dando entre paréntesis"*.

Para asegurarnos de que la calculadora está lista para que escribamos en la *línea de entrada*, basta con presionar la tecla " . Ahora escribamos en la *línea de entrada* la orden y la descripción:

```
sq\dc("j1,0,1,3.1;   r1,1,0,2;   r2,1,2,5;   r3,2,0,1;
j2,2,0,-1.4")
```

Para escribir los números, basta con presionar las teclas correspondientes. Para escribir el signo \ hay que presionar la tecla **2** y luego la tecla **©**. Para escribir las letras, hay que presionar la tecla **j** y luego la tecla que ostente sobre ella la letra deseada, en color púrpura.

Es importante notar que el signo que aparece delante del valor de la fuente j2, o sea delante del 1.4, es un signo de *negativo*, el cual se escribe con la tecla · . No se debe confundir con el signo de *resta*, el cual se escribe con la tecla **|** .

Es aún más importante asegurarnos de entender qué quiere decir ese signo negativo. Hemos presentado los nodos en la descripción de la fuente j2 de la forma en que nos lo indica la flecha de la figura, es decir desde el nodo 2 hacia el nodo 0. Que el valor de la fuente sea negativo, significa que la corriente que fluye de 2 hacia 0 es de –1.4 A. Sería equivalente decir que la fuente va desde el nodo 0 hacia el nodo 2 y que su valor es de +1.4 A.

Nótese que, como ningún elemento es descrito por una fila de 5 términos, no hay necesidad de rellenar con ceros el final de ninguna fila.

Cuando hayamos escrito la orden y la descripción en la *línea de entrada*, presionamos la tecla ¸ . Esto dará inicio a la simulación. En la pantalla aparecerán ciertas frases (en inglés) que nos indican el progreso de la simulación. Al cabo de algunos segundos, en el *área de historia* de la calculadora aparecerá la palabra **Done**, indicando que la simulación ha terminado. En la mía tomó 13 segundos. Dependiendo del modelo de la calculadora, la simulación puede tomar más o menos tiempo.

Almacenadas en la *carpeta actual*, en variables con nombres apropiados, están todas las respuestas. Para ver la lista completa de todas las respuestas generadas, el usuario puede entrar en el entorno *Var-Link* a la *carpeta actual*. Si entramos, siguiendo los primeros tres pasos del procedimiento que usamos para borrar las variables innecesarias, veremos que las siguientes variables están dentro de la *carpeta actual*:

- Las corrientes a través de todos los elementos, en amperios: `ij1, ij2, ir1, ir2, ir3.` Estas corrientes están definidas como fluyendo desde el primer nodo hacia el segundo nodo en la descripción de cada elemento.

- Las caídas de voltaje en todos los elementos, en voltios: `vj1, vj2, vr1, vr2, vr3.` Estas caídas de voltaje están definidas del primer nodo con respecto al segundo nodo en la descripción de cada elemento.

- Las potencias reales consumidas por todos los elementos, en watts: `pj1, pj2, pr1, pr2, pr3.`

- Los voltajes de todos los nodos, con respecto al nodo de referencia, en voltios: `v0, v1 y v2`.

Nótese que, si el usuario no había limpiado la carpeta actual de variables viejas, estas variables también estarán ahí cuando entremos al entorno *Var-Link*.

Volvamos a la *línea de entrada*, presionando la tecla " . Presionemos la tecla **M** para limpiar la *línea de entrada*.

El problema nos pregunta los voltajes en todos los nodos. Para ver cada uno de estos valores, escribimos su nombre en la *línea de entrada* y presionamos la tecla ¸ . El valor almacenado aparecerá en *el área de historia*.

Si escribimos `v0` en la *línea de entrada* y presionamos la tecla ¸ , aparecerá **0** en *el área de historia*, indicando el valor en voltios del voltaje del nodo 0. Esto es lo esperado, pues 0 voltios es y debe ser siempre el voltaje del nodo de referencia.

Si escribimos `v1` en la *línea de entrada* y presionamos la tecla ¸ , aparecerá **`5.`** en *el área de historia*, indicando el valor en voltios del voltaje del nodo 1.

Si escribimos `v2` en la *línea de entrada* y presionamos la tecla ¸ , aparecerá **`2.`** en *el área de historia*, indicando el valor en voltios del voltaje del nodo 2.

Con esto hemos respondido a la incógnita que nos planteó el problema. Veamos otro problema semejante.

Problema N° 002

Planteamiento. Dado el siguiente circuito, obtenga los voltajes de todos los nodos.

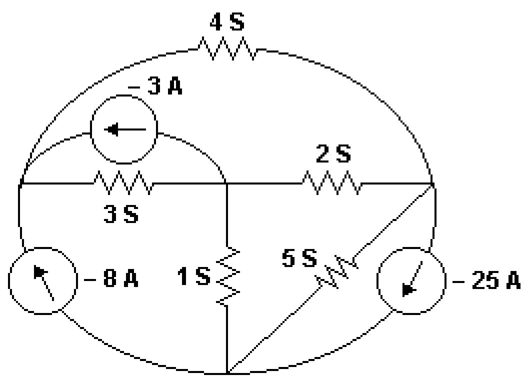

Figura 5. Circuito para el Problema N° 002.

Solución:

El único concepto nuevo en este circuito con respecto al anterior es que presenta conductancias en vez de resistencias. Como ya explicamos arriba, para simularlas basta con invertir su valor e introducirlas como resistencias.

El **primer paso** es universal, y consiste en nombrar todos los nodos y elementos del circuito. En el circuito mostrado arriba hay cuatro nodos, uno de los cuales se debe tomar como referencia. Nosotros escogimos el inferior, porque tiene el mayor número de fuentes conectadas a él. Así mismo, el circuito tiene ocho elementos: tres fuentes de corriente independientes y cinco conductancias. Simularemos las conductancias como resistencias. No vamos a detallar aquí los nombres que les hemos dado a los elementos y a los nodos, pues podrán verse en la descripción del circuito, que se muestra en el siguiente paso. Recuérdese que estos nombres son según el gusto del usuario.

El **segundo paso** es escribir en la *línea de entrada* de la calculadora la orden de la simulación, especificando que queremos usar la puerta **dc,** seguida de la descripción del circuito que queremos usar como dato de entrada, entre paréntesis:

```
sq\dc("j1,0,1,-8;  j2,2,1,-3;  j3,3,0,-25;  r1,1,2,1/3;
r2,1,3,1/4; r3,2,3,1/2; r4,2,0,1; r5,3,0,1/5")
```

Recuérdese que el signo de *negativo* que aparece ante los valores de las fuentes se escribe con la tecla · ; no se debe confundir con el signo de *resta*, el cual se escribe con la tecla **|** .

Tras escribir la orden y la descripción en la *línea de entrada*, presionamos la tecla ¸ para dar inicio a la simulación. Vemos en la pantalla las frases que nos indican el progreso de la simulación, y al cabo de algunos segundos, en *el área de historia* de la calculadora aparecerá la palabra **Done**, indicando que la simulación ha terminado. Mi calculadora tomó 20 segundos en completar esta simulación.

Almacenadas en la *carpeta actual*, en variables con nombres apropiados, están todas las respuestas. Veamos cuáles son los voltajes de los nodos. No hace falta preguntar el voltaje del nodo 0, pues es el de referencia.

Escribimos `v1` en la *línea de entrada* y presionamos ¸ . En *el área de historia* aparece **1**. Este es el voltaje del nodo 1, en voltios.

Escribimos `v2` en la *línea de entrada* y presionamos ¸ . En *el área de historia* aparece **2**, como voltaje del nodo 2, en voltios.

Escribimos `v3` en la *línea de entrada* y presionamos ¸ . En *el área de historia* aparece **3**, como voltaje del nodo 3, en voltios.

Estas son las respuestas correctas. Antes de ver más problemas, aprendamos algunos conceptos nuevos.

2.9 Modos de operación

La calculadora TI es, a mi parecer, la mejor calculadora que ha sido construída hasta el día de hoy. Un aspecto realmente sobresaliente es la manera inteligente en que maneja los números exactos y aproximados.

Si presionamos la tecla **3** , podremos ver los modos de operación de la calculadora. Se encuentran distribuidos en tres grupos, los cuales pueden ser vistos presionando las teclas ***f*** , **„** y **…**. Presione **„** para ver el segundo grupo. En la mitad inferior de la pantalla verá que dice Exact/Approx......AUTO. Este es el modo que controla la manera en que la calculadora maneja los números exactos y aproximados. Presione **N** para volver a la *pantalla hogar*. Presione **M** para limpiar la *línea de entrada*.

Vamos a ver de qué se trata este modo de operación y cómo impacta las simulaciones del *Symbulator*.

El Symbulator ajusta automáticamente el modo en AUTO. Cuando la calculadora TI está en ese modo, un `2` significa para ella un 2 exacto, mientras que un `2.` (o sea un 2 seguido de un punto) significa para ella un número aproximado que se acerca muchísimo a un 2. De hecho la calculadora considerará cualquier número que tenga un punto como un número aproximado. Veamos esto en dos ejemplos:

Escriba en la *línea de entrada* `2/3` y presione ¸ . (El signo de división “ / ” se obtiene con la tecla **[÷]**). Verá que en el *área de historia* aparecerá como respuesta **`2/3`** idéntico a la entrada. Esto significa que la calculadora ha asimilado la fracción que introdujimos como compuesta por enteros exactos. Por lo tanto, la respeta como tal y no trata de aproximarla a un valor con decimales.

Ahora escriba en la *línea de entrada* `2./3` y presione ¸ . Verá que en el *área de historia* aparecerá **`.666666667`**. Esta respuesta, que le pondría a cualquier lector del

Apocalipsis los pelos de punta, significa que la calculadora ha asimilado lo que introdujimos como una fracción compuesta por un número entero y un número aproximado. Por lo tanto, como al menos uno de los números involucrados en la respuesta es un número aproximado, la calculadora automáticamente aproxima la respuesta a un valor con decimales.

Si el lector es muy observador habrá notado que los voltajes que obtuvimos en el Problema 001 venían acompañados por un punto al final, mientras que los voltajes que obtuvimos en el problema 002 no tenían ese punto junto a ellos. ¿Por qué? La explicación es sencilla. En el primer problema había valores de circuito que eran aproximados (por ejemplo, 3.1 ó -1.4), y por lo tanto las respuestas fueron también aproximadas. En el segundo problema, por el contrario, todos los valores eran enteros exactos, y las respuestas fueron exactas también.

Esto nos enseña algo: siempre que los valores de un circuito se introduzcan como enteros, las respuestas serán enteras. Y siempre que se introduzcan como aproximados, las respuestas serán aproximadas.

Una manera de convertir un número exacto, por ejemplo 31/10, a aproximado, es usar el comando *approx* de la calculadora TI. Así, si escribimos `approx(31/10)`, la calculadora nos dará el valor aproximado de este número exacto, o sea 3.1.

Podemos también convertir un número aproximado, tal como 3.1, a exacto usando el comando *exact* de la calculadora TI. Así, si escribimos `exact(3.1)`, la calculadora nos dará el valor exacto de este número aproximado, o sea 31/10.

2.10 Fuentes dependientes

En términos sencillos, una fuente dependiente es aquella cuyo valor depende de otra variable del circuito. Una fuente dependiente de corriente es aquella cuyo valor es función de una corriente de control, que está en otra parte del circuito. Por su parte, una fuente dependiente de voltaje es aquella cuyo valor es función de un voltaje de control, que está en otra parte del circuito.

Hemos dicho ya que una de las grandes ventajas del *Symbulator* es que permite la simulación natural de las fuentes dependientes, sin necesidad de artificios o nomenclaturas adicionales. Esto se debe a que el método empleado por este simulador es el de generar ecuaciones, a diferencia de los simuladores convencionales, los cuales usan el método conocido como *análisis nodal modificado* o *transformada gyrator*, que requiere de una nomenclatura diferente para distinguir las fuentes dependientes de las independientes.

Dos precauciones se deben tener al simular circuitos con fuentes dependientes en el *Symbulator*. La primera es asegurarse de expresar la variable de control, ya sea un voltaje o una corriente, en términos de las respuestas que se obtendrán de la simulación. Y la segunda es asegurarse de limpiar bien la *carpeta actual* de todas las variables innecesarias para la simulación, especialmente de aquellas que se llamen como las respuestas que esperamos obtener y que estamos usando para controlar las fuentes dependientes.

2.10.1 Fuentes dependientes de voltaje

Para una fuente dependiente de voltaje, el voltaje de control puede ser el voltaje de un nodo con respecto al nodo de referencia. En este caso, se puede expresar como el voltaje del nodo, así: `v1`. Pues siempre el voltaje del nodo de referencia es 0 voltios.

El voltaje de control también puede ser la caída de voltaje en un elemento. La caída de voltaje en una de las respuestas que nos da el Symbulator, y por ello el voltaje de control se puede expresar así: `vr1`.

El voltaje de control también puede ser la caída de voltaje entre dos nodos cualesquiera. En este caso, el voltaje de control se puede expresar como un diferencial del voltaje de los dos nodos, así: `(v1-v2)`.

Veamos un problema como ejemplo.

Problema N° 003

Planteamiento. Para el circuito de dos nodos de la figura, encuentre el valor de las corrientes que se han marcado en el dibujo como i_A, i_B e i_C.

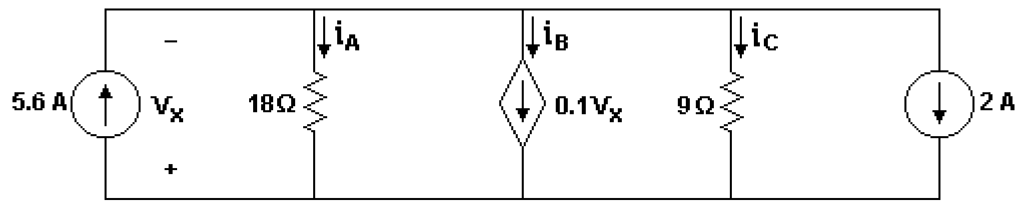

Figura 6. Circuito para el Problema N° 003.

Solución:

El **primer paso** es universal, y consiste en nombrar todos los nodos y elementos del circuito. El circuito tiene dos nodos. Tomaremos el nodo de abajo como referencia. En el circuito, tenemos una fuente dependiente de voltaje. Su voltaje de control es la caída de voltaje en la fuente de 5.6 amperios, y está marcado en el dibujo como v_x. La corriente de esta fuente de corriente fluye desde el nodo inferior hacia el nodo superior. Eso significa que, cuando la describamos, también su caída de voltaje quedará definida como del nodo inferior con respecto al nodo superior, polaridad que coincide con la caída de voltaje v_x del dibujo de circuito. Si nombramos a esta fuente `jx`, tendremos que el voltaje de control será `vjx`.

El **segundo paso** es escribir en la *línea de entrada* de la calculadora la orden de la simulación y la descripción del circuito.

El dibujo nos indica la dirección que debemos darle al flujo de corriente de las fuentes de corriente. Pero, ¿en qué dirección vamos a definir la corriente de las resistencias? Esto queda a nuestro criterio, y crea la necesidad de tomar nuevamente una decisión.

Las corrientes que nos pregunta el problema, i_A, i_B e i_C, vienen de arriba hacia abajo. Si definimos las corrientes de las resistencias como fluyendo desde arriba hacia abajo, habremos hecho coincidir las direcciones de las preguntas y las respuestas. Esto significa que no tendremos que cambiar el signo de las respuestas para contestar a las preguntas. Esta decisión es opcional. Puede usarse cualquier dirección para el flujo de la corriente de las resistencias. Las decisiones correctas nos ayudarán, sin embargo, a dar las corrientes como respuestas directamente.

Para facilitar aún más el responder a las preguntas, daremos a los elementos que están en las ramas de las corrientes i_A, i_B e i_C los siguientes nombres: r_A, j_B y r_{CC}, respectivamente. Esto no es obligatorio, pues el usuario le puede dar a los elementos el nombre que guste. Sin embargo, los nombres fáciles de relacionar suelen ser más amigables al momento de ver las respuestas. Nótese que no usamos la variable r_C, porque ésta, como dijimos en el punto *1.9.5*, es una variable reservada por la calculadora TI.

He aquí, pues, la orden y la descripción para la *línea de entrada*:

```
sq\dc("jx,0,1,5.6; ra,1,0,18; jb,1,0,.1*vjx; rcc,1,0,9; j2,1,0,2")
```

El valor de la fuente dependiente pudo haberse descrito también como `-.1*v1`. Presionamos la tecla ¸ para dar inicio a la simulación. Vemos en la pantalla frases sobre el progreso de la simulación, y luego la palabra **`Done`** en el *área de historia*, indicando el fin de la simulación. La mía tomó 12 segundos en completar esta simulación. Almacenadas en la *carpeta actual*, en variables con nombres apropiados, están todas las respuestas: voltajes, corrientes y potencias.

Escribimos `ira` en la *línea de entrada* y presionamos ¸. En *el área de historia* aparece **`3.`** como corriente a través del resistor r_A, en amperios, la cual corresponde a la corriente i_A que nos solicita el problema.

Con `ijb` en la *línea de entrada*, tras presionar ¸, obtenemos en *el área de historia* **`–5.4`** como corriente a través de la fuente de corriente j_B, en amperios, la cual corresponde a la corriente i_B que nos solicita el problema.

Con `ircc` en la *línea de entrada* y ¸ , obtenemos en *el área de historia* **`6.`** como corriente a través de la resistencia r_{CC}, en amperios, la cual corresponde a la corriente i_C que nos solicita el problema.

Este ejemplo ilustra el porqué es importante que el Symbulator nos dé como respuestas las corrientes de las fuentes de corriente. La corriente de la fuente j_B nos era desconocida, y el problema nos la solicitó como respuesta. Veamos otro ejemplo similar.

Problema N° 004

Planteamiento. Para el circuito de dos nodos de la figura, encuentre el valor de las corrientes que se han marcado en el dibujo como i_1, i_2, i_3 e i_4.

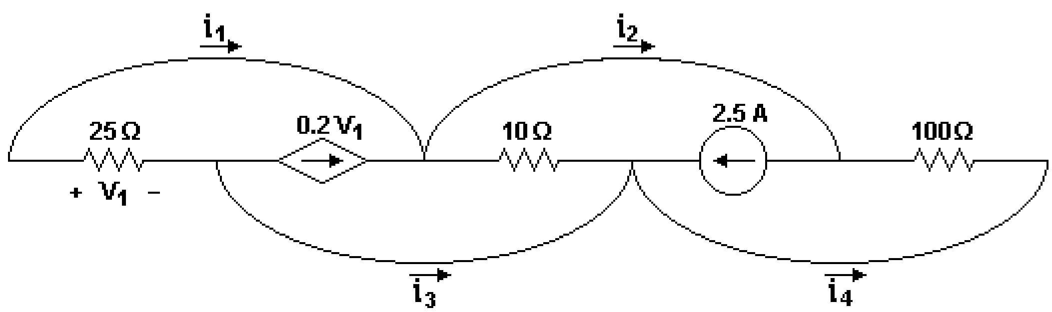

Figura 7. Circuito para el Problema N° 004.

Solución:

El **primer paso**, universal en su uso, es nombrar todos los nodos y elementos del circuito. El circuito tiene dos nodos. O al menos eso nos dice el encabezado del problema. Miremos más de cerca, con más cuidado. Claramente, vemos que existen seis puntos de unión en el circuito que podrían ser considerados nodos. Notamos también que los cuatro lazos externos no tienen ningún elemento resistivo o fuente en ellos, sino que son cortocircuitos. Tienen igual potencial en sus terminales. Esto quiere decir que los seis puntos de unión que identificamos están agrupados en dos grupos de tres, donde cada grupo tiene el mismo potencial. Cuando el problema nos dice "*el circuito de dos nodos*", se refiere a este hecho. Es cierto que, desde el punto de vista del potencial, pueden considerarse a estos seis puntos como sólo dos nodos. Ahora bien, el mismo encabezado nos pregunta cuánto valen las corrientes a través de estos cortocircuitos. La corriente puede *entrar* a un nodo, y también *salir* de él. Pero no puede *fluir a lo largo* de él. Por lo tanto, para efectos de la simulación, es necesario considerar esos tramos como cortocircuitos, y no como nodos. Esta conclusión es aplicable a cualquier simulador, ya sea *SPICE* o *Symbulator*. Por lo tanto, el circuito que vamos a simular tiene seis nodos y cuatro cortocircuitos, entre otros elementos.

Aprovecharemos este punto para explicar cómo definir una fuente dependiente de voltaje cuyo voltaje de control es el voltaje entre dos nodos. ¿Cuál de esos seis nodos será nuestro nodo de referencia? Cualquiera de ellos es una alternativa posible. Sólo una de las alternativas nos facilita las cosas: debemos tomar como nodo de referencia al nodo que está entre la resistencia de 2.5 ohmios y la fuente dependiente. ¿Por qué? Porque el tener este nodo como referencia nos permite definir el voltaje de control de la fuente dependiente como el voltaje de un solo nodo. Supongamos que nombramos los nodos de izquierda a derecha, así: 1 al primero, 0 al segundo (referencia), 2 al tercero, 3 al cuarto, 4 al quinto y 5 al sexto. Con estos nombres, el voltaje de control de la fuente dependiente sería *(v1-v0)*, y como *v0* = 0 volts, esto equivale a *v1*.

El **segundo paso** es escribir en la *línea de entrada* de la calculadora la orden de la simulación y la descripción del circuito.

¿Y cómo escogemos las mejores direcciones de flujo de corriente? El dibujo nos indica la dirección que debemos darle al flujo de las fuentes de corriente. En cuanto a las resistencias, da lo mismo cualquier dirección. Para los cortocircuitos, recordamos lo que aprendimos en el problema anterior: es conveniente hacer coincidir las direcciones de flujo de las preguntas y las respuestas. Así, pues, definimos las direcciones de flujo de corriente en los cortocircuitos según las direcciones de las incógnitas que nos solicita el problema.

Para responder a las preguntas más fácilmente, daremos a los cortocircuitos que están en las ramas de las corrientes i_1, i_2, i_3 e i_4 los nombres o s_1, s_2, s_3 y s_4.

He aquí, pues, la orden y la descripción para la *línea de entrada*:

```
sq\dc("r1,1,0,25; j1,0,2,.2*v1; r2,2,3,10; j2,4,3,2.5; r3,4,5,100; s1,1,2,0; s2,2,4,0; s3,0,3,0; s4,3,5,0")
```

Dos cosas son dignas de notar. La primera, que hizo falta colocar un cero al final de cada fila de cortocircuito, para mantener la simetría de la descripción. La segunda, que el hecho de que el voltaje de control de la fuente se llame *v1* igual que en el dibujo del circuito se debe exclusivamente a que nosotros, intencionalmente, decidimos colocar el nodo de referencia en el sitio apropiado, y llamar al primer nodo con el nombre 1. De otra forma, el voltaje de control hubiera sido de la forma *(vm-vn)*, donde m y n serían los nombres de los dos nodos de los extremos de la resistencia de 2.5 ohmios. La otra alternativa, muy fácil también, era haber definido el voltaje de control como `vr1`, o sea la caída de voltaje en `r1`, porque ya vimos que `r1` está definida entre el nodo `1` y el nodo `0`.

Damos inicio a la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`** en el *área de historia*. Mi calculadora tomó 32 segundos en simular este circuito. El tiempo aumentó porque aumentó la cantidad de nodos. Recuérdese que el Symbulator utiliza análisis de nodos. Esto significa que el número de ecuaciones e incógnitas aumenta igual que el número de nodos. Almacenadas en la *carpeta actual* están todas las respuestas, esperando por nosotros.

Obtengamos las corrientes, con el procedimiento ya conocido por nosotros para ver las respuestas. La corriente `is1` es de **`-2.`** amperios. La corriente `is2` es de **`3.`** amperios. La corriente `is3` es de **`-8.`** amperios. La corriente `is4` es de **`-.5`** amperios.

Veamos ahora otro tipo de fuentes dependientes.

2.10.2 Fuentes dependientes de corriente

Para las fuentes dependientes de corriente, la corriente de control es con mucha frecuencia la corriente a través de un elemento. En estos casos si, por ejemplo, el nombre del elemento fuese *r1*, la corriente puede ser expresada así: `ir1`.

Puede darse el caso de que la corriente de control sea la corriente a través de una rama en la cual no haya un elemento. En este caso, habrá que definir esa rama como un cortocircuito para poder pedir luego la corriente a través de él, como hicimos en el ejemplo anterior. Así si, por ejemplo, el cortocircuito se llama *s1*, la corriente se expresa como `is1`.

Veamos un problema como ejemplo.

Problema N° 005

Planteamiento. Dado el siguiente circuito, determine el valor del voltaje marcado como *v* y la potencia absorbida por la fuente independiente de corriente.

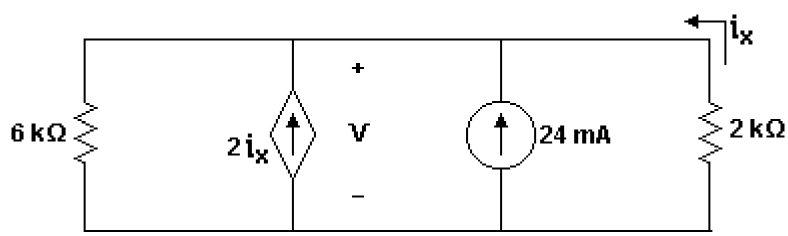

Figura 8. Circuito para el Problema N° 005.

Solución:

El **primer paso** es nombrar todos los nodos y elementos del circuito. Tomaremos como nodo de referencia al nodo inferior, para que así el voltaje *v* sea igual al voltaje del nodo superior. De haber tomado como referencia al nodo superior, el voltaje *v* sería igual al negativo del voltaje del nodo inferior. A la resistencia de 2kΩ le nombraremos r_X, y definiremos su flujo de corriente de abajo hacia arriba. La intención es hacer que la resistencia coincida tanto en dirección de flujo de corriente como en nombre con la corriente de control llamada en el dibujo i_X. Esto facilita la descripción, al menos visualmente. En el caso de la otra resistencia, la de 6kΩ, la dirección en que se defina su flujo no tiene importancia.

El **segundo paso** es escribir en la *línea de entrada* de la calculadora la orden de la simulación y la descripción del circuito.

```
sq\dc("r1,1,0,6E3; j1,0,1,2*irx; j2,0,1,24E-3; rx,0,1,2E3")
```

Nótese el hecho de que la corriente de control aparece en el dibujo del circuito como i_X, pero en la descripción del circuito se definió como ir_X. ¿Por qué no se definió esta corriente en la matriz como i_X ? La razón es muy simple: porque i_X no significa absolutamente nada para la calculadora, pues la calculadora no ha visto el dibujo del circuito. Hay que hablarle a la máquina en términos de variables que ella pueda comprender. Y la variable que ella comprende es ir_X .

En el valor `6E3`, la `E` quiere decir potencia de 10. Así, `2E3` quiere decir $2x10^3$. Esta es una nomenclatura usada actualmente en todas las calculadoras programables de alto desempeño, como las TI, las HP y las Casio.

Digno de resaltar es que, aunque los valores de las resistencias en el dibujo están dados en kΩ, en la descripción sus valores deben introducirse en Ω. Igualmente en el caso de la fuente de corriente independiente, en el dibujo los valores están en mA, pero se deben describir en la matriz en A. La razón para esto es que la calculadora no tiene forma de adivinar si estamos hablando de Ω, de kΩ o de MΩ.

En ocasiones, es recomendable ver las ordenes que damos a la calculadora desde el punto de vista de ella. Así evitaremos situaciones absurdas, tales como definir la corriente de control como i_X y dar los valores de corriente en mA, las cuales nos llevarían a respuestas equivocadas.

Damos inicio a la simulación con ¸ . Tras las frases en la pantalla, aparece **Done** en el *área de historia*. Mi calculadora tomó 10 segundos en simular este circuito. Almacenadas en la *carpeta actual* están todas las respuestas. Este ejemplo nos brinda la oportunidad de apreciar claramente la agilidad del *Symbulator*, y su potencial para servir como herramienta de aprendizaje y verificación de respuestas. Resolviendo el circuito a mano, tomaría varios minutos obtener todos los voltajes, corrientes y potencias del circuito, y tendríamos el riesgo de cometer un error accidentalmente. Con el *Symbulator*, obtener todas estas respuestas toma apenas 10 segundos, y tenemos la tranquilidad que nos da la certeza de que todas las respuestas están correctas.

Obtengamos las respuestas con el procedimiento que ya nos es familiar. El voltaje *v* lo obtenemos con `v1` y es de **`14.4`** voltios. La potencia absorbida por la fuente independiente la obtenemos con `pj2` y es de **`-.3456`** watts. La pregunta que nos plantea el problema es la potencia absorbida, por lo que la respuesta obtenida se utiliza directamente. Si la pregunta hubiese sido la potencia entregada por la fuente, la respuesta hubiese requerido la inversión del signo de la respuesta, o sea `-pj2`, lo que nos da **`.3456`** watts (positivos). Veamos otro ejemplo similar, para afianzar los conceptos aprendidos hasta ahora.

Problema N° 006

Planteamiento. Determine i_X en el siguiente circuito.

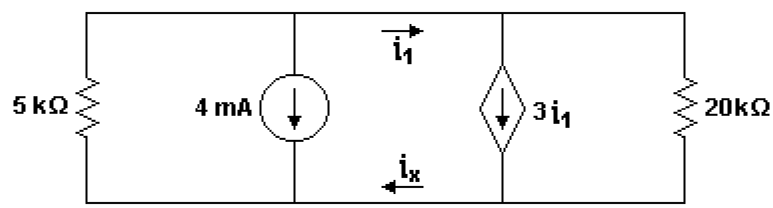

Figura 9. Circuito para el Problema N° 006.

Solución:

Como **primer paso**, nombre todos los nodos y elementos del circuito. Al igual que el Problema 004, este problema aparenta tener dos nodos, cuando en realidad requiere más nodos y algunos cortocircuitos para poder simularse. En el caso de este nuevo problema, la ubicación de la corriente de control i_1 forza a la existencia en la parte superior del circuito de dos nodos y un cortocircuito (al cual llamaremos convenientemente s_1). Por su parte, la peculiar ubicación de la corriente incógnita i_X obliga a la existencia en la parte inferior del circuito de dos nodos más y otro cortocircuito (al cual llamaremos convenientemente s_{XX}, pues s_X es una variable reservada).

Debido a la configuracion del circuito, da lo mismo nombrar a cualquier nodo como referencia. Igualmente irrelevante es la dirección en la cual se defina el flujo de corriente de las resistencias. La de las fuentes de corriente está definida por las flechas en el dibujo, como siempre. Y la de los cortocircuitos se selecciona para hacerla coincidir con la dirección de la corriente de control y la corriente incógnita.

El **segundo paso** es escribir en la *línea de entrada* de la calculadora la orden de la simulación y la descripción del circuito.

```
sq\dc("r1,1,0,5E3;  j1,1,0,4E-3;  s1,1,2,0;  sxx,3,0,0;
j2,2,3,3*is1; r2,2,3,20E3")
```

En este punto no debe ser sorpresa para el lector que la corriente de control aparezca en el dibujo como i_1 y en la descripción como is_1.

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece `Done` en el *área de historia*. Mi calculadora tomó 17,5 segundos en simular este circuito. Almacenadas en la *carpeta actual* están todas las respuestas.

Obtengamos la respuesta. La corriente i_X la obtenemos con `isxx` y es de **`.000571429`** amperios.

2.11 Fuente de voltaje

Idealmente, una fuente de voltaje o fuente de tensión es un elemento que forza una diferencia de voltaje definida entre sus dos terminales, y entrega o consume potencia.

2.11.1 Notación

En el caso de la fuente de voltaje, la mínima información necesaria para describirla por completo es la siguiente:

1. *¿Qué clase de elemento es?* Es una fuente de voltaje. La letra que identifica a las fuentes de voltaje es la **e**.

2. *¿Cómo se llama la fuente?* Para diferenciar una fuente de voltaje dada de cualquier otra fuente de voltaje, se le debe dar un nombre cualquiera, que empiece con **e**.

3. *¿Dónde se encuentra conectada la fuente?* Una fuente de voltaje tiene dos extremos, cada uno de ellos conectado a un nodo distinto. Es necesario darle al simulador los nombres de los dos nodos a los cuales está conectada la fuente, en el orden adecuado que indique la polaridad que deseamos darle a la fuente: el primer nodo con respecto al segundo nodo de la descripción. En un dibujo de un circuito, la polaridad que debemos darle a la fuente nos la indican los signos de positivo y negativo que se colocan junto a la fuente.

Al describir una fuente de voltaje, el orden en que se colocan los nodos tiene que ser aquel orden que defina claramente la polaridad que le estamos dando a la fuente. Es decir, la información principal que transmite el orden de los nodos en la descripción del elemento es la polaridad de la fuente. Ahora bien, una vez que hemos definido esta polaridad, quedan también establecidas, indirectamente, la dirección que tendrá en la respuesta el flujo de corriente a través de la fuente: desde el primer nodo hacia el segundo nodo de la descripción, y la polaridad que tendrá en la respuesta la caída de voltaje en la fuente: del primer nodo con respecto al segundo nodo de la descripción.

Este punto generalmente confunde a aquellos que no están familiarizados con la simulación de circuitos. Piensan que, si una fuente tiene el primer nodo positivo con respecto al segundo, la corriente en ella debe ir del segundo nodo hacia el primero. Y de hecho es así, en la mayoría de los casos.

Lo cierto es que establecer la polaridad que se usó para definir una variable no es lo mismo que establecer qué nodo es más positivo que otro. Tomemos como ejemplo la siguiente frase: “*El voltaje del nodo A es –5 voltios con respecto al nodo B*”. En esta frase, la polaridad que se ha defindo es del nodo A con respecto al nodo B. Sin embargo, el nodo B es positivo con respecto al nodo A.

Así mismo, establecer la dirección que se usó para definir una variable de flujo de corriente no es lo mismo que establecer en qué dirección está fluyendo la corriente. Tomemos otra frase como ejemplo: "*La corriente que fluye desde el nodo A hacia el nodo B es de –3 amperios*". En esta frase, la dirección del flujo de la corriente se está definiendo desde el nodo A hacia el nodo B, pero la corriente convencional está fluyendo realmente de B hacia A.

En pocas palabras: *En todos los elementos que hemos visto, la dirección de flujo de corriente que se usa para definir las respuestas, es desde el primer nodo hacia el segundo nodo, y la polaridad de caída de voltaje que se usa para definir las respuestas es del primer nodo con respecto al segundo. Además, en una fuente de voltaje, el valor de la fuente está dado asumiendo una polaridad del primer nodo con respecto al segundo.*

4. *¿Cuánto vale la fuente?* Una fuente de voltaje tiene un valor. Este valor debe entregarse al *Symbulator* en voltios. En el caso del análisis de corriente directa, el valor de una fuente de voltaje independiente puede ser un número real o una variable. Si la fuente es dependiente, su valor debe ser una expresión algebraica. Esta expresión estará compuesta por un número y una variable. Esta variable será una de las futuras respuestas de la simulación.

Así, las fuentes de voltaje se definen como se muestra a continuación:

e*nombre, nodo 1, nodo 2, valor en voltios*

Como hemos dicho, el valor de la fuente de voltaje es captado por el simulador considerándolo como el voltaje del nodo 1 con respecto al nodo 2. Por ejemplo, si al definir una fuente de voltaje, el valor que se introduce es *10*, el simulador considerará que la diferencia de voltaje del nodo 1 con respecto al nodo 2 es de 10 voltios.

Nótese que la fila que describe a la fuente de voltaje está compuesta de cuatro términos. En este caso, también se utiliza un cero al final en el caso de que alguna otra fila en la descripción del circuito tenga cinco términos. Ejemplo:

enombre, nodo 1, nodo 2, valor, 0

Debe tenerse la precaución de no introducir el valor de la fuente en una unidad distinta a los voltios (V). Por ejemplo, si el voltaje se introduce en milivoltios (mV), las respuestas estarán equivocadas.

En el caso de otros análisis distintos a la corriente directa, los valores pueden ser fasores, funciones de la frecuencia o funciones del tiempo.

2.11.2 Respuestas relacionadas

En el caso de una fuente de voltaje, se entregan tres respuestas relacionadas: la corriente a través de la fuente, la caída de voltaje en la fuente y la potencia real consumida por ella. La corriente se almacena en una variable llamada i*nombre*, donde *nombre* es el de la fuente. La caída de voltaje se almacena en una variable llamada v*nombre*, donde *nombre* es el de la fuente. La potencia real consumida, por su parte, se almacena en una variable llamada p*nombre*, donde *nombre* es el de la fuente.

La corriente, como hemos explicado detalladamente en párrafos anteriores, está definida como fluyendo desde el primer nodo de la descripción hacia el segundo nodo de la descripción.

La caída de voltaje está definida como del primer nodo de la descripción respecto al segundo nodo de la descripción.

La potencia que se da como respuesta es la potencia consumida, o sea el negativo de la potencia entregada. Es importante tener esto en mente cuando se interpreten estas respuestas.

Problema N° 007

Planteamiento. En el siguiente circuito, determine la potencia que absorbe cada uno de los elementos.

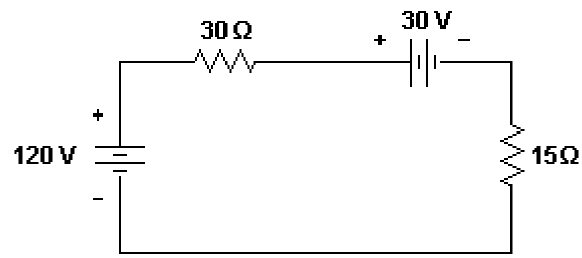

Figura 10. Circuito para el Problema N° 007.

Solución:

Como **primer paso**, nombre todos los nodos y elementos del circuito. Nombraremos al nodo inferior como referencia. Recuérdese que los nombres de las fuentes de voltaje deben empezar con **e**. El orden de los nodos de las fuentes de voltaje está determinado por la polaridad que nos indica el dibujo. El de las resistencias es irrelevante.

Como **segundo paso** escribimos la orden de la simulación y la descripción del circuito en la *línea de entrada*.

```
sq\dc("e1,1,0,120; r1,1,2,30; e2,2,3,30; r2,3,0,15")
```

Iniciamos la simulación con ¸. Tras las frases en la pantalla, aparece **`Done`** en el *área de historia*. Mi calculadora tomó 13 segundos en simular este circuito. Almacenadas en la *carpeta actual* están todas las respuestas.

La potencia consumida por la fuente e_1 la obtenemos con `pe1` y es de **`-240`** watts. Esto significa que la fuente entregó 240 watts. Dado que el planteamiento pregunta cuánta potencia consume el elemento, la respuesta es el primer valor. La potencia consumida por la fuente e_2 la obtenemos con `pe2` y es de **`60`** watts. Que este valor sea positivo significa que la fuente no entrega potencia, sino que la consume. La potencia consumida por la resistencia r_1 la obtenemos con `pr1` y es de **`120`** watts. La potencia consumida por la resistencia r_2 la obtenemos con `pr2` y es de **`60`** watts. Estas son las respuestas. Este ejemplo nos da la oportunidad de verificar si el *Symbulator* cumple con la ley de la conservación de la energía. Preguntemos a la calculadora cuánta es la energía neta del circuito, así: `pe1+pr1+pe2+pr2`, y obtendremos la respuesta esperada: **`0`** watts.

2.11.3 Fuentes dependientes

Todos los conceptos que aprendimos para la simulación de fuentes dependientes cuando vimos las fuentes de corriente, son igualmente válidos para las fuentes de voltaje. Veamos algunos ejemplos que lo demuestran. Empecemos con un problema sencillo de fuente de voltaje dependiente de voltaje.

<u>Problema N° 008</u>

Planteamiento. En el siguiente circuito, determine la potencia absorbida por cada uno de los elementos.

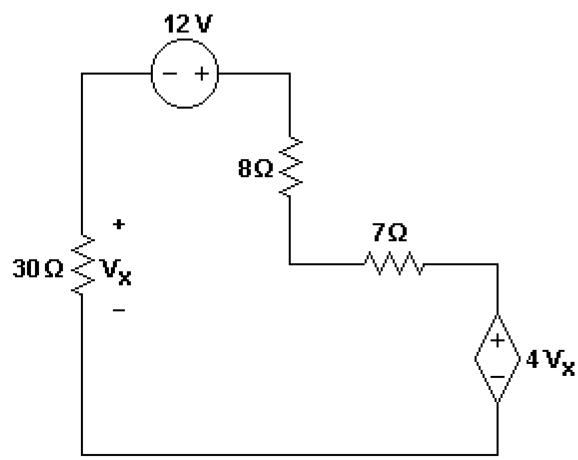

Figura 11. Circuito para el Problema N° 008.

Solución:

Como **primer paso**, nombre todos los nodos y elementos del circuito. El orden de los nodos en la descripción de las fuentes de voltaje está determinado por la polaridad que nos indica el dibujo. El de las resistencias es irrelevante. Con la experiencia adquirida hasta ahora, debe ser fácil notar que si tomamos al nodo inferior como referencia y nombramos *x* al nodo de la esquina superior izquierda, podremos nombrar al voltaje de control *vx*. Los demás nodos fueron nombrados 1, 2 y 3, en el sentido de las manecillas del reloj.

Como **segundo paso** escribimos la orden de la simulación y la descripción del circuito en la *línea de entrada*.

```
sq\dc("r1,x,0,30;   e1,1,x,12;   r2,1,2,8;   r3,2,3,7;
e2,3,0,4*vx")
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **Done** en el *área de historia*. Mi calculadora tomó 18 segundos en simular este circuito. En la *carpeta actual* están todas las respuestas.

Si escribimos `pr1` en la *línea de entrada* y presionamos ¸ , obtendremos en *el área de historia* **96/125** watts. Según hemos explicado anteriormente, las respuestas son exactas porque todos los valores del circuito son exactos. Usualmente, en los libros de texto no encontramos respuestas exactas, como 96/125 watts. Es más frecuente encontrar valores aproximados, como 0.768 watts. El valor exacto es tan correcto como el aproximado; sin embargo, en un examen el profesor podría preferir el valor aproximado. *(Nótese que los términos exacto y aproximado se usan aquí en la acepción que tienen en la jerga de las calculadoras programables.)*

Obtener el valor aproximado es sencillísimo. Basta con presionar la tecla **¥** antes de presionar la tecla ¸ para obtener la respuesta aproximada.

Con `pr1` en la *línea de entrada*, presionamos **¥** y luego ¸ , para obtener en *el área de historia* **.768** watts. El mismo procedimiento se repite para las demás variables.

Con `pe1`, presionamos **¥** y luego ¸ , para obtener **1.92** watts.

Con `pr2`, presionamos **¥** y luego ¸ , para obtener **.2048** watts.

Con `pr3`, presionamos **¥** y luego ¸ , para obtener **.1792** watts.

Con `pe2`, presionamos **¥** y luego ¸ , para obtener **`-3.072`** watts.

Ahora veamos un problema con dos fuentes de voltaje dependientes. Los voltajes de control son más complicados esta vez.

Problema N° 009

Planteamiento. Encuentre la potencia absorbida por cada uno de los elementos del siguiente circuito.

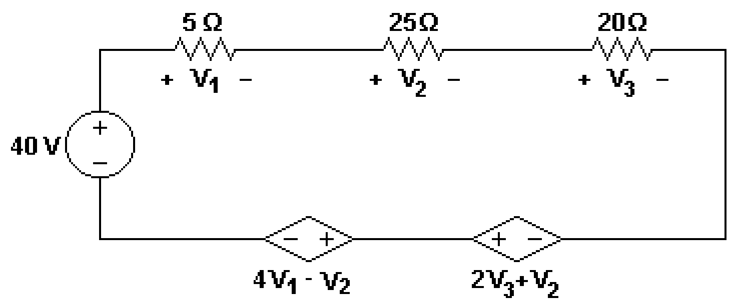

Figura 12. Circuito para el Problema N° 009.

Solución:

Como **primer paso**, nombre todos los nodos y elementos del circuito. El orden de los nodos en la descripción de las tres fuentes de voltaje está determinado por la polaridad que nos indica el dibujo. El de las tres resistencias es irrelevante. No importa que nodo se escoja como referencia, los voltajes de control seguirán siendo complicados. Escogemos como referencia el nodo de la esquina inferior izquierda, y nombramos a los demás *a*, *b*, *c*, *d* y *e*, en el sentido de las manecillas del reloj.

Como **segundo paso** escribimos la orden de la simulación y la descripción del circuito en la *línea de entrada*.

```
sq\dc("e1,a,0,40;   r1,a,b,5;   r2,b,c,25;   r3,c,d,20;
e2,e,d,2*vr3+vr2; e3,e,0,4*vr1-vr2")
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`** en el *área de historia*. Mi calculadora tomó 26 segundos en simular este circuito. En la *carpeta actual* están todas las respuestas.

Con `pe1`, obtenemos **`80`** watts.

Con `pr1`, obtenemos **`20`** watts.

Con `pr2`, obtenemos **`100`** watts.

Con `pr3`, obtenemos **`80`** watts.

Con `pe2`, obtenemos **`-260`** watts.

Con `pe3`, obtenemos **`-20`** watts.

Estas son las respuestas correctas. Veamos ahora un ejemplo de fuente de voltaje dependiente de corriente.

<u>Problema N° 010</u>

Planteamiento. Encuentre la corriente marcada como i_X en el siguiente circuito.

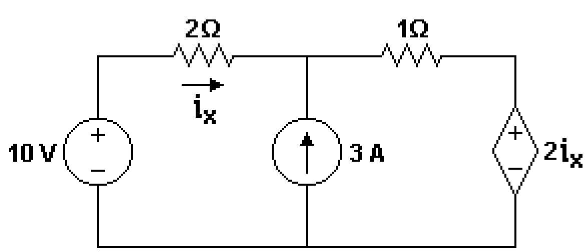

Figura 13. Circuito para el Problema N° 010.

Solución:

Como **primer paso**, nombre todos los nodos y elementos del circuito. El orden de los nodos en la descripción de las tres fuentes está determinado por el dibujo. La dirección de la corriente de la resistencia de 2 ohmios (la cual será nombrada r_X) debe coincidir con la dirección de i_X, y la de la otra resistencia es irrelevante. Escogemos como referencia el nodo inferior, y nombramos a los demás *1*, *2* y *3*, en el sentido de las manecillas del reloj.

Como **segundo paso** escribimos la orden de la simulación y la descripción del circuito en la *línea de entrada*.

```
sq\dc("e1,1,0,10;    rx,1,2,2;    j1,0,2,3;    r1,2,3,1;
e2,3,0,2*irx")
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`** en el *área de historia*. Mi calculadora tomó 16 segundos en simular este circuito. En la *carpeta actual* están todas las respuestas.

Con `irx`, presionamos **¥** y luego ¸ , para obtener **`1.4`** amperios.

Esta respuesta es correcta. Vamos a terminar este capítulo con un problema que combina todo lo que hemos aprendido en él.

<u>Problema N° 011</u>

Planteamiento. Encuentre la corriente, caída de voltaje y potencia absorbida de cada elemento en el siguiente circuito. Demuestre que la suma de las potencias es cero.

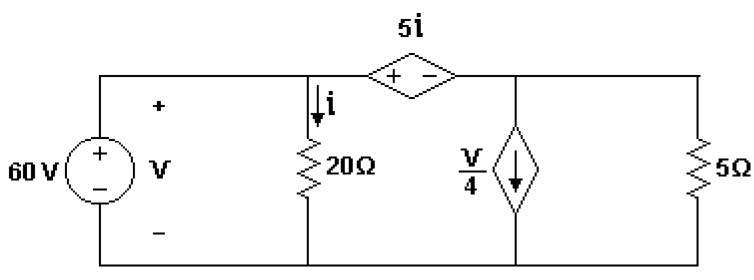

Figura 14. Circuito para el Problema N° 011.

Solución:

Como **primer paso**, nombre todos los nodos y elementos del circuito. Consideramos que en este punto, las explicaciones sobre cómo hacerlo salen sobrando. Como **segundo paso**, escribimos la orden de la simulación y la descripción del circuito en la *línea de entrada*.

```
sq\dc("e1,1,0,60; r1,1,0,20; e2,1,2,5*ir1; j1,2,0,ve1/4; r2,2,0,5")
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`** en el *área de historia*. Mi calculadora tomó 15 segundos en simular este circuito. En la *carpeta actual* están todas las respuestas. Sin más detalles, a continuación presentamos las respuestas: `ve1` y `vr1` son ambas **`60`** voltios, `ve2` es **`15`** voltios, `vj1` y `vr2` son ambas **`45`** voltios, `ie1` es **`-27`** amperios, `ir1` es **`3`** amperios, `ie2` es **`24`** amperios, `ij1` es **`15`** amperios, `ir2` es **`9`** amperios, `pe1` es **`-1620`** watts, `pr1` es **`180`** watts, `pe2` es **`360`** watts, `pj1` es **`675`** watts, `pr2` es **`405`** watts. Nótese que sólo una fuente está entregando potencia. ¿Cuánto tiempo hubiese tomado a un estudiante encontrar, sin errores, estas 15 respuestas? Al Symbulator le tomó, en promedio, 1 segundo por cada una.

2.12 Descripción errada

Supongamos que hemos cometido un error al escribir en la *línea de entrada* la descripción del circuito, y que nos hemos percatado de ello cuando la simulación ya ha terminado. Evidentemente, hay que corregir el error y realizar la simulación otra vez. Lo primero es borrar las variables de la *carpeta actual*. Luego hay que colocar de nuevo en la *línea de entrada* la descripción del circuito y la orden para ejecutar la simulación. Es una tarea tediosa escribir por segunda vez una descripción. Lo más práctico es aprovechar el texto ya escrito. Hay dos maneras de hacer esto:

1) Si la descripción anterior todavía está en la *línea de entrada*, aparecerá como letras blancas en fondo negro. Basta con desplazar el cursor hasta el error, usando las flechas azules, y corregirlo. Nótese que para borrar una letra a la izquierda del cursor se usa la tecla **0** .

2) Si por alguna razón la descripción anterior no está en la *línea de entrada*, entonces es necesario desplazar el cursor hasta la posición en el *área de historia* en que ésta esté. Una vez que la descripción anterior esté seleccionada (o sea como letras blancas con fondo negro), se presiona la tecla ¸ para hacer una copia en la *línea de entrada*. Con esta copia en la línea de entrada, se corrige el error como se describió en el caso anterior.

Una vez corregido el error, iniciamos por segunda vez la simulación con ¸. Con esta explicación, hemos terminado el capítulo de simulación numérica en corriente directa.

Capítulo

3

Simulación simbólica en corriente directa

3.1 Definición de simulación numérica

En el capítulo anterior realizamos una gran cantidad de simulaciones, en las cuales todos los valores de los elementos de circuito eran numéricos. También las respuestas obtenidas en la simulación fueron todas numéricas. Este tipo de simulación se conoce como simulación numérica, y es posible realizarla usando cualquiera de los simuladores numéricos convencionales, tales como el *Electronic Workbench*, *PSpice* y otros. En ellos, son posibles las tareas tales como describir circuitos con fuentes dependientes y obtener las respuestas en todos los nodos y elementos, pero ofrecen un mayor grado de dificultad que en el *Symbulator*.

Una razón por la cual estas tareas resultan más sencillas en el *Symbulator* que en otros simuladores es precisamente que el primero es un simulador simbólico. Por ejemplo, las fuentes dependientes son manejadas simbólicamente al momento de la resolución del circuito, aún cuando al final sus valores terminen siendo numéricos. La otra razón es que fue diseñado y desarrollado por un estudiante, con la idea clara de proporcionar a los estudiantes lo que estos desean recibir como respuesta en una simulación.

El pleno potencial del *Symbulator* se hace verdaderamente manifiesto sólo cuando ejecutamos simulaciones simbólicas.

3.2 Definición de simulación simbólica

Una simulación simbólica es una simulación en la cual al menos uno de los valores del circuito no es un valor numérico – ya sea real o complejo -, sino un valor simbólico. Este *valor simbólico* puede ser una variable desconocida, una expresión algebraica, una función del tiempo, una función de la frecuencia u otro tipo de expresión no numérica.

Aprendimos ya que el tipo de entrada determinará el tipo de salida. La naturaleza de los valores de los elementos del circuito se refleja en la naturaleza de las respuestas de la simulación. Valores exactos producirán respuestas exactas. Valores numéricos producirán respuestas numéricas. Así también, valores simbólicos producirán respuestas simbólicas. Podemos asegurar que una simulación simbólica, por tener algún valor simbólico en el circuito de entrada, tendrá algunas respuestas simbólicas.

3.3 Simulación simbólica para obtener respuesta simbólica

Una simulación es considerada simbólica si la descripción del circuito que se utiliza contiene algún valor simbólico. Hay básicamente dos tipos de simulación simbólica: la que busca una respuesta simbólica y la que busca una respuesta numérica. La simulación del primer tipo, cuyo objetivo es una respuesta simbólica, es muy útil en problemas de diseño, y para la obtención de expresiones que describan simbólicamente el comportamiento de alguna característica del circuito, como por ejemplo la potencia consumida, en función de los valores de sus elementos. También podría usarse con propósitos académicos, para demostrar algunas leyes del analisis de circuitos. La ley de Ohm, por ejemplo, puede demostrarse mediante una simulación simbólica en el *Symbulator*. Para aprender cómo simular simbólicamente, veamos algunos problemas.

Problema N° 012

Planteamiento. Mediante la simulación simbólica del siguiente circuito básico, demuestre la ley de Ohm y la fórmula de la potencia consumida por una resistencia.

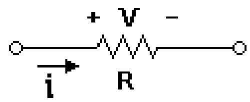

Figura 15. Circuito para el Problema N° 012.

Solución:

Este es el circuito más básico posible. El **primer paso** es nombrar los nodos y elementos del circuito. El **segundo paso** es ordenar la simulación, con la descripción del circuito.

```
sq\dc("r,1,0,r;j,0,1,i")
```

Nótese la manera novedosa en que se han nombrado los elementos: como sólo hay un elemento de cada tipo, hemos usado sólo la letra inicial como nombre de los elementos. Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`** en el *área de historia*. Mi calculadora tomó 6 segundos en simular este circuito. Por ser este el circuito más básico posible, este es el menor tiempo posible para una simulación en el *Symbulator*. En la *carpeta actual* están todas las respuestas.

Podemos demostrar la ley de Ohm solicitando a la calculadora el valor de `vr`. Obtenemos **`i*r`** voltios, como era de esperarse. Para demostrar la fórmula de la potencia consumida por una resistencia, solicitamos el valor de `pr`. Obtenemos $\mathbf{i^2*r}$ vatios, que es correcto.

La verificación de la ley de Ohm también pudo haberse logrado usando una fuente de voltaje en vez de una fuente de corriente, con la siguiente descripción:

```
sq\dc("r,1,0,r;e,1,0,v")
```

Mi calculadora tomó 7 segundos en simular este circuito. Nótese cómo una fuente de voltaje toma más tiempo que una fuente de corriente en ser simulada. Aunque en este caso la diferencia en tiempo es despreciable (sólo 1 segundo), al trabajar con circuitos más complejos esta diferencia puede ser considerable.

Demostramos la ley de Ohm solicitando `ir`. Obtenemos **`v/r`** voltios, que es otra manera de expresar la ley. Demostramos la fórmula de la potencia consumida por una resistencia solicitando `pr`. Obtenemos $\mathbf{v^2/r}$ vatios, que es la otra fórmula para obtener esta potencia.

Este ejercicio nos enseña que la simulación simbólica es una manera fácil de demostrar y aprender las leyes básicas del análisis de circuitos.

3.4 Simulación simbólica para obtener respuesta numérica

Algunas otras simulaciones simbólicas tienen como objetivo el obtener al final una respuesta numérica. En estos casos, se utiliza el comando `solve`, la herramienta `solves` y el *Modo (Experto)*.

Usualmente, para obtener una respuesta numérica, se requiere por cada valor simbólico que tenga el circuito, conocer de antemano el valor numérico de una *respuesta*, es decir un voltaje, corriente o potencia.

3.5 Comando `solve`

Veamos un ejemplo simple del segundo tipo de simulación simbólica, que busca una respuesta numérica, utilizando el comando `solve`.

Problema N° 013

Planteamiento. Determine los valores numéricos de v_x e i_x en el siguiente circuito.

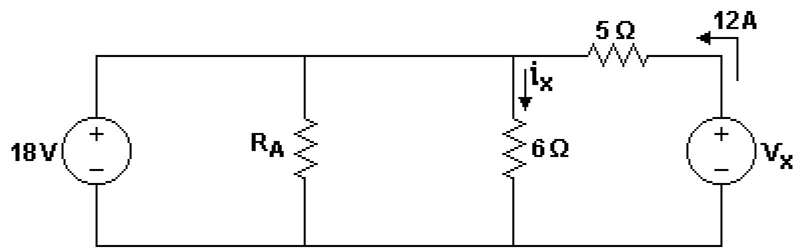

Figura 16. Circuito para el Problema N° 013.

Solución:

Cuando se le compara con otros circuitos resueltos anteriormente, este circuito muestra dos diferencias notorias. La primera es que mezcla elementos con valor simbólico y elementos con valor numérico. La segunda es que tenemos de antemano el valor numérico de una de las corrientes del circuito: 12 amperios a través de la resistencia de 5 ohms.

Nombramos los nodos y elementos. Hemos definido la dirección de corriente de las resistencias de 5 y 6 ohmios en la misma dirección de las flechas en el dibujo. Como valores de los elementos simbólicos, hemos introducido las mismas variables que aparecen en el dibujo. Debemos tener la precaución de que en la carpeta actual no existan ya estas variables. Es decir, debemos asegurarnos de que no existan variables con los nombres r_a y v_x, porque de existir, la calculadora reemplazaría automáticamente el nombre simbólico por este valor numérico almacenado, trastornando la simulación. Para ello, podemos utilizar el comando `DelVar` de la calculadora TI. Así:

```
DelVar ra,vx
```

Con ello, estamos ordenando: "*Borra de la carpeta actual las variables llamadas* `ra` *y* `vx`". Si estas variables existen, serán borradas. Si no existen, nada sucede. Otra forma en que pudimos haber borrado estas variables es entrar al entorno Var-Link, y luego buscarlas y borrarlas manualmente. La forma que hemos mostrado arriba es más rápida.

Habiendo borrado estas variables, damos la descripción del circuito y ordenamos la simulación.

```
sq\dc("e1,1,0,18;ra,1,0,ra;r1,1,0,6;r2,2,1,5;ex,2,0,vx"
)
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`**. Mi calculadora tomó 16 segundos en simular este circuito. En la *carpeta actual* están todas las respuestas.

Encontramos i_x solicitando el valor de `ir1`. Obtenemos **`3`** amperios. Hasta aquí no hay nada nuevo.

La variante se presenta a la hora de encontrar v_x. En primera instancia, nos parece que podríamos encontrar v_x solicitando el valor de `v2`. Lo hacemos y obtenemos **`vx`** voltios. Obviamente, una respuesta así no nos aporta nada nuevo, pues requerimos una respuesta numérica. Entonces recordamos que el dibujo, en el planteamiento del problema, nos advirtió que la corriente a través de la resistencia de 5 ohmios es de 12 amperios. ¿Cuánto vale esta corriente según nuestro simulador? Solicitamos el valor de `ir2`. Obtenemos **`(vx-18)/5`** amperios. Esta expresión simbólica, producto de la simulación simbólica, nos ayudará a encontrar el valor de v_x. ¿Cómo? Veamos.

Sabemos que la corriente vale **`12`** y tenemos su expresión en función de **`vx`**. Basta con unir ambos términos mediante una igualdad para obtener una ecuación. Al resolver esta ecuación para la incógnita **`vx`**, tendremos el valor numérico de la fuente de voltaje desconocida. Podríamos resolver la ecuación manualmente, pero aprovecharemos la tecnología y utilizaremos el comando `solve` de la calculadora TI. Una explicación detallada del uso de este comando puede encontrarse en el Manual de Usuario de la calculadora. Este es un comando muy poderoso, el cual es utilizado extensamente por el *Symbulator* en sus cálculos internos. Ahora lo utilizaremos manualmente nosotros. Introducimos lo siguiente en la *línea de entrada*:

```
solve(ir2=12,vx)
```

Con este comando estamos diciéndole a la máquina: "*Resuelve la ecuación* `ir2=12` *para la incógnita* `vx`*, y dime la respuesta*", o en otras palabras: "*Dime cuánto vale v_x si ir2 vale 12*". Presionamos ¸ , y casi instantáneamente obtenemos la respuesta: **`vx = 78`** voltios. ¡Así de fácil! La incógnita que resolvimos es la respuesta que faltaba.

Habiendo terminado este problema, hay tres aclaraciones que deben hacerse.

1) El comando `solve` de la calculadora nos ha dicho que `vx` vale **`78`**, pero no ha almacenado el valor 78 en la variable v_x. Esto se debe a que este comando resuelve ciertas ecuaciones para ciertas incógnicas y presenta el resultado en el *área de historia*, pero no almacena en la memoria estas respuestas en las variables que resolvió como incógnitas. ¿Desea verificarlo? Preguntemos a la calculadora cuánto vale la variable v_x. Escriba en el área de entrada `vx`, y obtendrá **`vx`** como respuesta. Esto significa que no hay ningún valor almacenado en la variable v_x.

2) En este problema, la incógnita simbólica del circuito era también una de las respuestas que nos solicitaba el problema. En otros problemas, sin embargo, las incógnitas simbólicas del circuito y las respuestas no son la misma cosa. Un ejemplo con esta característica es el que veremos a continuación.

3) El valor de r_a queda indefinido. No hay ninguna manera matemática en que se pueda obtener un valor numérico que lo describa. La razón es simple: para obtener el valor numérico de un elemento simbólico, necesitamos conocer una *pista* que lo involucre, es decir una corriente, voltaje o potencia que esté en función de éste valor simbólico. Por ejemplo, pudimos encontrar el valor numérico de v_x porque teníamos una *pista* que lo involucraba: conocíamos el valor numérico de una corriente que es función de este valor. Pero no podemos encontrar el valor numérico de r_a, porque nos falta una *pista*: no conocemos el valor numérico de ninguna corriente, voltaje o potencia que lo involucre.

Le recomendamos al lector releer este Problema N° 013 hasta que sienta haber entendido a cabalidad los conceptos que en él se exponen. Sólo entonces, prosiga con la lectura.

3.6 Herramienta `solves`

La herramienta `solves` es parte del *Symbulator*. Está programada utilizando tecnología desarrollada para el *Symbulator*, y su propósito es cumplir dos tareas al mismo tiempo: 1) resolver ecuaciones haciendo uso interno del comando `solve`, y 2) almacenar todas las respuestas en sus respectivas variables. Esta herramienta se ejecuta sin colocar ningún argumento de entrada entre los paréntesis, así:

```
sq\solves()
```

Presionamos ¸ . Aparecerá en la pantalla una forma con dos campos y con dos opciones. El primer campo es para escribir la ecuación o las ecuaciones que se desean resolver. El segundo campo es para introducir la incógnita o las incógnitas para las cuales se desea resolver. Y las dos opciones son para escoger entre si las ecuaciones se resolverán como reales o complejas, y si se desea utilizar ecuaciones condicionales, es decir valores previamente conocidos. Presionemos **N** para volver a la *pantalla hogar*.

A continuación veamos un problema, semejante al anterior, que nos permitirá aprender el uso básico de la herramienta `solves`.

Problema N° 014

Planteamiento. Determine los valores numéricos de v_x e i_x en el siguiente circuito.

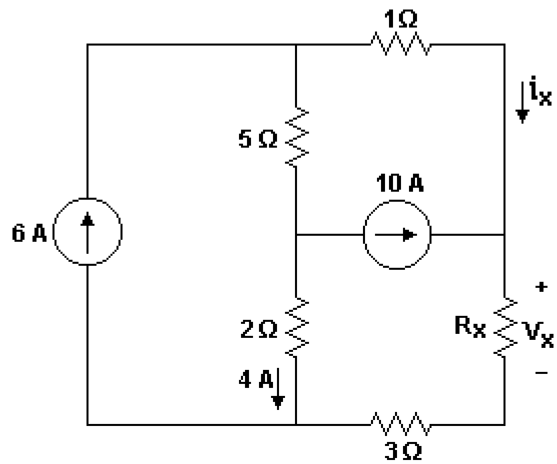

Figura 17. Circuito para el Problema N° 014.

Solución:

Nótese que en este circuito, la incógnita simbólica es r_x, mientras que las respuestas solicitadas son i_x y v_x. Muy probablemente, estas respuestas serán funciones de la incógnita. Por ello, la estrategia que debemos seguir es primero simular el circuito, resolver la incógnita, y luego solicitar las respuestas para obtener sus valores numéricos.

Nombremos los nodos y elementos. Definamos la dirección de corriente de las resistencias de 1 y 2 ohmios en la misma dirección de las flechas en el dibujo. Como valor del elemento simbólico, introduzcamos la misma variable que aparece en el dibujo, para asegurarnos de que no exista ya una variable con el nombre r_x.

```
DelVar rx
```

Tomamos como referencia el nodo de la esquina inferior izquierda. Damos la descripción del circuito y ordenamos la simulación.

```
sq\dc("j1,0,1,6;r1,1,2,1;r2,1,3,5;j2,3,2,10;r3,3,0,2;rx
,2,4,rx;r4,4,0,3")
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`**. Mi calculadora tomó 30 segundos en simular este circuito. A este punto, el lector debe haber notado ya que el tiempo consumido por una simulación es proporcional al número de nodos que tiene el circuito. La razón de esto es que el *Symbulator* “piensa” nodalmente, y creará una ecuación para cada nodo del circuito (excepto el nodo de referencia). Así, este circuito con 4 nodos distintos al de referencia, toma aproximadamente cuatro veces más tiempo que un circuito con 1 nodo distinto al de referencia. Obviamente, hay otros factores que también influyen en el tiempo de ejecución, tales como el tipo de análisis, la presencia de fuentes de voltaje y elementos especiales, y los valores simbólicos.

Volviendo al problema, en la *carpeta actual* están todas las respuestas de nuestra simulación simbólica. Encontramos i_x solicitando el valor de `ir1`. Obtenemos no un valor numérico, sino una función de **`rx`**. También para v_x, al solicitar el valor de `vrx`, obtenemos una función de **`rx`**. Tal como supusimos, las respuestas que necesitamos numéricamente se encuentran en este momento como funciones de la incógnita. Para conocer los valores numéricos de i_x y v_x, primero debemos resolver el valor numérico de r_x. Para permitirnos encontrar este valor, el problema nos ha dicho de antemano que la corriente a través de la resistencia de 2 ohmios (a la cual nosotros hemos llamado `r3`) es de 4 amperios.

El comando `solve` no es el más apropiado para esta tarea. Como hemos visto, este comando es sumamente rápido y eficaz para resolver ecuaciones, y sin duda podría resolver esta pequeña ecuación en unas milésimas de segundo. Sin embargo, a manera de respuesta sólo nos mostraría el valor de **`rx`** en el *área de historia*, sin almacenar el valor numérico en esta variable. Por ello, resulta más apropiado utilizar la herramienta `solves`.

Primero, veamos cuál hubiese sido el procedimiento usando el comando `solve`. Resolvamos la ecuación:

```
solve(ir3=4,rx)
```

Obtenemos como respuesta **`rx = 40`**. Como tenemos que evaluar otros valores que son función de esta incógnita, debemos tomar la respuesta que nos da el comando `solve` y almacenarla manualmente en su variable, utilizando el comando `Define` de la calculadora, así:

```
Define rx = 40
```

O sino, utilizando el comando **!** de la calculadora, el cual es equivalente al comando `Define` y se escribe con la tecla **§** . Así:

```
40!rx
```

Almacenado el valor numérico de r_x, se puede solicitar a la calculadora los valores numéricos de i_x y v_x. Para `ir1` obtenemos **`-8`** amperios. Para `vrx` obtenemos **`80`** voltios.

Esta manera de resolver el problema nos lleva a las respuestas, pero requiere de cuatro pasos para encontrarlas: 1) simular el circuito, 2) resolver la incógnita, 3) almacenar la incógnita y 4) evaluar las respuestas que son funciones de esta incógnita.

Utilizando la herramienta `solves`, encontramos estas respuestas en tres pasos: 1) simular el circuito, 2) resolver y almacenar la incógnita en un paso y 3) evaluar las respuestas que son funciones de esta incógnita. Para ver cómo, primero debemos borrar la variable r_x, para empezar desde el inicio.

```
DelVar rx
```

Recuérdese que las respuestas de la simulación todavía están almacenadas en la memoria. Ejecútese la herramienta:

```
sq\solves()
```

Aparece la forma. En el primer campo, titulado “Equations”, introduciremos la ecuación:

```
ir3=4
```

En el segundo campo, titulado “Unknowns”, introduciremos la incógnita:

`rx`

Como las opciones de abajo están ajustadas a los valores que deseamos, es decir *ecuaciones reales* y *sin condicionales*, los dejamos tal como están. Presionamos ¸ , dos veces si es necesario. Aparecerá un texto que nos indica que la herramienta está resolviendo las ecuaciones, y luego almacenando las respuestas. Abajo aparece la respuesta: **`rx = 40`**. Eso es todo. Presionamos ¸ , para volver a la *pantalla hogar*. En este momento, ya se encuentra almacenado el valor de 40 en la variable r_x. Si lo desea, verifíquelo preguntando a la máquina el valor de `vx`, y verá que responde **`40`**.

El tercer paso es, simplemente, evaluar las respuestas que el problema nos ha solicitado, es decir los valores numéricos de i_x y v_x. Para `ir1` obtenemos **`–8`** amperios. Para `vrx` obtenemos **`80`** voltios.

Existe una manera aún más rápida de encontrar estas mismas respuestas en un sólo paso, utilizando el *Modo (Experto)*. Dado que el uso de este modo requiere mucho más conocimiento y experiencia, le dedicaremos el siguiente capítulo. Veamos, a continuación, otro problema semejante al que recién resolvimos, para afianzar los conceptos aprendidos.

Problema N° 015

Planteamiento. Determine los valores numéricos de v_x e i_x en el siguiente circuito.

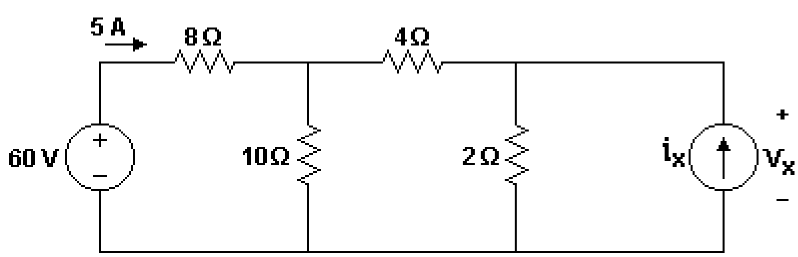

Figura 18. Circuito para el Problema N° 015.

Solución:

Nótese que en este circuito, la incógnita simbólica es i_x, la cual al mismo tiempo es una de las respuestas solicitadas. La otra respuesta solicitada es v_x, la cual seguramente será una función de i_x. Simularemos el circuito, resolveremos la incógnita, y luego solicitaremos las respuestas para obtener sus valores numéricos.

Nombremos los nodos y elementos. Definamos la dirección de corriente de la resistencias de 8 ohmios en la misma dirección de la flecha en el dibujo. Limpiemos el nombre i_x.

```
DelVar ix
```

Tomando como referencia el nodo inferior, damos la descripción del circuito y ordenamos la simulación.

```
sq\dc("e1,1,0,60;r1,1,2,8;r2,2,0,10;r3,2,3,4;r4,3,0,2;jx,0,3,ix")
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece `Done`. Mi calculadora tomó 19 segundos en simular este circuito. En la *carpeta actual* están todas las respuestas de nuestra simulación simbólica. Como sabemos que las respuestas son actualmente simbólicas, debemos resolver la incógnita i_x. Dado que una de las respuestas es una función de `ix`, la mejor forma de resolver este problema es usando la herramienta `solves`.

```
sq\solves()
```

Aparece la forma. Si no hemos borrado la carpeta actual después de la última vez que utilizamos la herramienta, es posible que en los campos veamos el texto que introdujimos anteriormente. En "Equations" introducimos la nueva ecuación, reemplazando la anterior:

```
ir1=5
```

En el segundo campo, titulado “Unknowns”, introduciremos la incógnita:

```
ix
```

Las opciones de abajo están en los valores que deseamos, así que las dejamos tal como están. Presionamos ¸ , dos veces si es necesario. El texto nos indica que la respuesta es: **`ix = 1`**. Esa es una de las respuestas. Presionamos ¸ , para volver a la *pantalla hogar*. Evaluamos la respuesta restante. Para `v3` obtenemos **`8`** voltios. Veamos otro ejemplo de simulación simbólica con respuesta numérica.

<u>Problema N° 016</u>

Planteamiento. Determine el valor de *k* que provocará que el voltaje v_y sea cero.

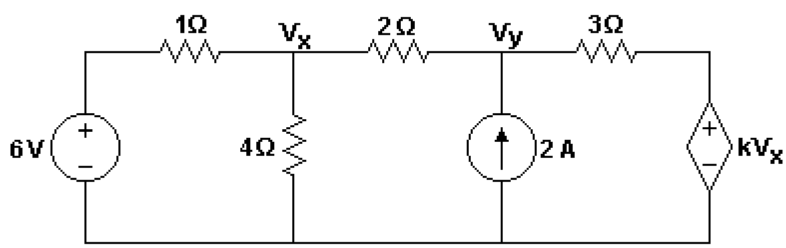

Figura 19. Circuito para el Problema N° 016.

Solución:

Nombremos los nodos y elementos. Nombraremos *x* y *y* a los nodos centrales, por razones obvias. Limpiemos el nombre *k*.

```
DelVar k
```

Tomando como referencia el nodo inferior, damos la descripción del circuito y ordenamos la simulación.

```
sq\dc("e1,1,0,6;r1,1,x,1;r2,x,0,4;r3,x,y,2;j1,0,y,2;r4,
y,2,3;e2,2,0,k*vx")
```

Iniciamos la simulación con ¸ . Tras las frases en la pantalla, aparece **`Done`**. Mi calculadora tomó 36 segundos en simular este circuito. En la *carpeta actual* están todas las respuestas de nuestra simulación simbólica. Como la única incógnita es también la respuesta que se nos ha solicitado, no habrá necesidad de evaluar otras variables en función de esta incógnita. Esto nos permite usar el comanso `solve`.

```
solve(vy=0,k)
```

Obtenemos como respuesta **`k = -13/4`**. Esta es la respuesta correcta. Pero, ¿cómo podemos estar seguros de esto, digamos, en medio de un exámen semestral?

3.7 Verificación de una simulación simbólica con respuesta numérica

Hay una manera muy fácil de verificar si son correctas las respuestas numéricas que hemos obtenido tras nuestra simulación simbólica. Lo único que hay que hacer es tomar del *área de historia* la descripción que se usó (asumiendo que no desea escribir la descripción nuevamente), y reemplazar en ella todos los valores simbólicos por los respectivos valores numéricos que se encontraron. Para tomar la descripción del área de historia, úsese el método para corregir errores en la descripción, aprendido al final del capítulo anterior.

Una vez ejecutada esta simulación con valores numéricos, evaluamos los valores que anteriormente conocíamos, tales como voltajes y corrientes. Si obtenemos los mismos valores que conocíamos anteriormente, entonces las respuestas obtenidas tras la simulación simbólica son correctas.

Como ejemplo, podemos verificar el valor de la *k* que obtuvimos en el problema anterior, mediante la siguiente simulación numérica:

```
sq\dc("e1,1,0,6;r1,1,x,1;r2,x,0,4;r3,x,y,2;j1,0,y,2;r4,
y,2,3;e2,2,0,-13/4*vx")
```

Nótese que lo único que ha cambiado es el valor de `e2`, el cual ahora tiene un valor numérico en vez de la *k* simbólica usada anteriormente. Recuérdese que anteriormente se nos pidió encontrar el valor de *k* para que v_y valiese cero voltios. Por lo tanto, si el valor que encontramos para *k*, es decir `-13/4`, es el valor correcto, entonces el voltaje en el nodo ***y***, debe ser cero voltios. Evaluemos este voltaje. Para `vy` obtenemos **`0`** voltios. Por lo tanto, la respuesta ha sido verificada.

Estar en la capacidad de verificar las respuestas es de especial importancia para un estudiante en medio de un exámen importante. El *Symbulator* ha sido reconocido por muchos de sus usuarios como una excelente herramienta para verificar respuestas, en tareas y exámenes. Esta es, probablemente, una de sus fortalezas más significativas.

Aprendamos ahora a simular como expertos en el *Symbulator*.

Capítulo 4 Simulación experta en corriente directa

4.1 Introducción al Modo (Experto)

Si el lector no está interesado en aprender a simular como un experto, puede omitir la lectura de este capítulo. En él se expone una serie de conocimientos teóricos avanzados sobre la simulación con el *Symbulator*, y luego se muestran ejemplos de cómo este conocimiento, a través del *Modo (Experto)*, puede ahorrar tiempo en simulaciones simbólicas. Este capítulo, de ser leído, debe leerse con atención y detenimiento.

En el capítulo anterior realizamos varias simulaciones simbólicas, y aprendimos a usar el comando `solve` y la herramienta `solves`. En este capítulo vamos a aprender a simular en el *Modo (Experto)*. Este modo apareció en la versión 3 del *Symbulator*, con el propósito de darle más poder al usuario sobre la manera en que se realiza la simulación.

En las simulaciones que hemos visto anteriormente, el *Symbulator* se comporta como una caja negra, en la cual el estudiante introduce una descripción de un circuito como dato de entrada y obtiene una serie de respuestas como datos de salida. Y eso es todo: mientras el usuario se dedica a piropear a la muchacha de al lado o a pensar en la inmortalidad del cangrejo, el simulador se dedica a generar las ecuaciones, resolverlas y almacenar las respuestas. Es decir, el usuario sólo alimenta el circuito, mientras que la responsabilidad total de la simulación recae en el simulador.

El *Modo (Experto)* le da al usuario el poder para desempeñar un papel más importante, pues le habilita para modificar las ecuaciones que el simulador ha generado, antes de que la calculadora las resuelva. A cambio, el usuario asume parte de la responsabilidad de la simulación, por lo que debe actuar con cuidado. Utilizar el *Symbulator* como un experto requiere de un conocimiento más profundo, y no es una necesidad para el usuario regular. Sin embargo, para aquellos dispuestos a aprender un poco más, el simular como experto puede resultar un objetivo atractivo, pues brinda una simulación más rápida.

4.2 Teoría

4.2.1 El procedimiento, de forma general

Una simulación experta, como hemos dicho, requiere que el usuario sea un experto. No sólo en el uso del *Symbulator*, sino también en cómo éste funciona internamente. Aprendamos, entonces, sobre los procesos internos de una simulación. A grandes rasgos, una simulación en el *Symbulator* consiste en lo siguiente:

1) Se generan ecuaciones para describir el circuito.

2) Se resuelven estas ecuaciones.

3) Se almacenan las respuestas.

En el marco de esta generalización, podemos decir que el *Modo (Experto)* nos permite manipular las ecuaciones generadas en el paso 1, antes de resolverlas en el paso 2, para que cuando las respuestas se almacenen en el paso 3 estén en la forma deseada.

4.2.2 Ecuaciones de primer nivel

Las ecuaciones indispensables que el *Symbulator* utiliza para resolver un circuito, se conocen como ecuaciones de primer nivel. El *Symbulator* genera dos tipos de ecuaciones de primer nivel: ecuaciones de nodo y ecuaciones especiales.

Habrá una ecuación de nodo por cada nodo que tenga el circuito. Estas ecuaciones se generan usando, en general, dos leyes básicas: la ley de Ohm y la ley de las corrientes de Kirchhoff. El *Symbulator* generará para cada nodo una ecuación que cumpla con la ley de las corrientes de Kirchhoff, como la suma de las corrientes de todos los elementos en contacto con ese nodo, igualada a cero. Para describir las corrientes de los elementos resistivos, el *Symbulator* usa la ley de Ohm.

Además, habrá una ecuación especial por cada elemento especial que tenga el circuito. Estas ecuaciones se generan de acuerdo a las fórmulas teóricas que describen el comportamiento de estos elementos. Aunque pueda parecer extraño, elementos tales como una fuente de voltaje y un cortocircuito son elementos especiales. Esto se debe simplemente al hecho de que el *Symbulator* usa un análisis de tipo nodal y, por ello, los únicos elementos que no son especiales, de los que hemos visto hasta ahora, son la resistencia y la fuente de corriente. Los demás son considerados especiales y tendrán una ecuación especial asociada, que engrosará la lista de ecuaciones generadas por el *Symbulator* para simularlo.

Por ejemplo, para resolver un circuito con 5 nodos, 2 fuentes de voltaje y 1 cortocircuito, el *Symbulator* generará un total de 8 ecuaciones de primer nivel. No importa si este circuito tiene mil resistencias y setecientas fuentes de corriente: seguirá teniendo 8 ecuaciones de primer nivel. Eso sí: la complejidad de estas 8 ecuaciones será mayor mientras más resistencias y fuentes de corriente haya conectadas a los nodos.

¿Cuántas ecuaciones de primer nivel generará el *Symbulator* para simular un circuito con 3 nodos y 1 fuente de voltaje? Es fácil responder que generará 4 ecuaciones de primer nivel.

Ahora bien, en nuestros cursos de álgebra aprendimos que para resolver completamente un sistema de *n* incógnitas, necesitamos *n* ecuaciones. Esto significa que en el primer circuito, al tener 8 ecuaciones, podremos encontrar el valor de 8 incógnitas. Y en el segundo, al tener 4 ecuaciones, podemos resolver 4 incógnitas. Estas incógnitas que pueden resolverse con las ecuaciones de primer nivel, se conocen como variables de primer nivel.

4.2.3 Variables de primer nivel

Como era de esperarse, el *Symbulator* genera las ecuaciones de primer nivel en función de las variables de primer nivel. Y habrá tantas variables de primer nivel como ecuaciones de primer nivel. Esto significa, indirectamente, que habrá tantas variables de primer nivel como nodos, fuentes de voltaje y cortocircuitos haya en el circuito. Con los elementos que hemos visto hasta ahora, el *Symbulator* genera tres tipos de variables de primer nivel: voltajes en los nodos, corrientes en fuentes de voltaje y corrientes en los cortocircuitos.

El voltaje de cada nodo del circuito es una variable de primer nivel. Al resolver un circuito que tenga tres nodos llamados `1`, `2` y `3`, el *Symbulator* creará para ellas tres variables de primer nivel llamadas `v1`, `v2` y `v3`.

La corriente a través de cada fuente de voltaje del circuito es una variable de primer nivel. Al resolver un circuito que tenga dos fuentes de voltaje, llamadas `e1` y `ex`, el *Symbulator* creará para ellas dos variables de primer nivel llamadas `ie1` e `iex`.

Igualmente, la corriente a través de cada cortocircuito es una variable de primer nivel. Así, al resolver un circuito que tenga un cortocircuito, llamado `s1`, el *Symbulator* creará para él una variable de primer nivel llamada `is1`.

Tras resolver las ecuaciones de primer nivel, en las variables de primer nivel se almacenarán las respuestas. ¿Puede el lector adivinar cómo se llaman estas respuestas? ¡Claro! Se llaman respuestas de primer nivel.

En el ejemplo del circuito con 5 nodos, 2 fuentes de voltaje y 1 cortocircuito, las 8 ecuaciones de primer nivel estarán escritas en función de 8 variables de primer nivel, y al ser resueltas nos darán 8 respuestas de primer nivel. Estas respuestas y variables serán también 5 de voltajes en los nodos, 2 de corrientes en las fuentes y 1 de corriente en el cortocircuito.

¿Cuántas variables de primer nivel generará el *Symbulator* para el circuito con 3 nodos y 1 fuente de voltaje? Obviamente, serán 4 ecuaciones de primer nivel, repartidas así: 1 para la fuente de voltaje y 3 para los nodos. Tras resolver las 4 ecuaciones de primer nivel, se encontrarán 4 respuestas de primer nivel, que serán 1 de la corriente de la fuente de voltaje, y 3 de los voltajes en los nodos.

4.2.4 Variables de segundo nivel

Hemos dicho que las corrientes de las fuentes de voltaje y de los cortocircuitos son consideradas variables de primer nivel. En cambio, las corrientes en otros elementos tales como las fuentes de corriente y resistencias, son llamadas variables de segundo nivel.

Igualmente, las caídas de voltaje en resistencias y fuentes son también variables de segundo nivel.

Así pues, en un circuito con 4 nodos, 3 resistencias, 2 fuentes de corriente, 2 fuentes de voltaje y 1 cortocircuito, habrá las siguientes variables de primer nivel:

- 4 voltajes en los nodos
- 2 corrientes en las fuentes de voltaje
- 1 corriente en el cortocircuito

Y las siguientes variables de segundo nivel:

- 3 corrientes en las resistencias
- 2 corrientes en las fuentes de corriente
- 7 caídas de voltaje en las 3 resistencias y las 4 fuentes

Esto da un total de 4+2+1=7 variables de primer nivel, con sus respectivas ecuaciones de primer nivel, y 3+2+7=12 variables de segundo nivel. Nótese que para el cortocircuito no se genera caída de voltaje (véase el punto 2.6.3 sobre las respuestas relacionadas a un cortocircuito).

Las variables de segundo nivel no se mencionan en las ecuaciones de primer nivel. Entonces, ¿cómo se relacionan estas variables de segundo nivel con las respuestas de la simulación?

4.2.5 Expresiones de segundo nivel

La respuesta es fácil. Para cada variable de segundo nivel, el *Symbulator* genera una expresión de segundo nivel, la cual está dada en función de variables de primer nivel exclusivamente. Así, teniendo las respuestas de primer nivel, todas las variables de segundo nivel quedan definidas por medio de estas expresiones.

Existen dos tipos de expresiones de segundo nivel: las que se reemplazan por sus respuestas luego de la solución, y las que se dejan permanentemente expresadas en términos de variables de primer nivel. Es decir, después de haber resuelto las ecuaciones de primer nivel, y almacenado las respuestas en sus variables, el *Symbulator* evalúa algunas expresiones de segundo nivel, generando así respuestas de segundo nivel que son almacenadas en sus variables. Sin embargo, otras expresiones de segundo nivel no serán evaluadas y seguirán siendo expresiones en vez de respuestas. Las expresiones de segundo nivel que no son evaluadas, son las que corresponden a las corrientes en las fuentes de corriente, y a las caídas de voltaje. Estas últimas seguirán siendo expresiones algebraicas en función de los voltajes de los nodos.

4.2.6 Variables de tercer nivel

En las simulaciones de corriente directa (DC) y corriente alterna (AC), existe otro grupo de variables que se conoce como variables de tercer nivel. Reciben este nombre porque se generan sólo después de la resolución de las ecuaciones de primer nivel y de la evaluación de las expresiones de segundo nivel. Es decir: aparecen sólo cuando ya se han resuelto los dos niveles anteriores de variables.

Las variables de este grupo son, exclusivamente, las potencias consumidas por los elementos. Se definen al final por la naturaleza misma de la potencia. Como potencia es el producto de caída de voltaje y corriente, se define usando variables de primer nivel (los voltajes) y de segundo nivel (las corrientes), las cuales ya deben estar definidas con anterioridad.

4.2.7 El procedimiento normal, en forma detallada

Hemos aprendido lo que son las ecuaciones, variables y respuestas de primer nivel, las expresiones y variables de segundo nivel, y las variables de tercer nivel. Ahora, veamos nuevamente el procedimiento de una simulación, pero esta vez en forma más detallada. Primero, el procedimiento de una simulación normal, es decir aquella que no utiliza el *Modo (Experto)*.

1) Se generan ecuaciones de primer nivel para describir el circuito que el usuario ha introducido, en función de variables de primer nivel. Se generan también expresiones de segundo nivel, en función también de las variables de primer nivel.

2) Se resuelven las ecuaciones de primer nivel, para las variables de primer nivel.

3) Se almacenan las respuestas de primer nivel en las variables de primer nivel. Se evalúan algunas expresiones de segundo nivel, y se almacenan sus respuestas en las variables de segundo nivel.

4) En los análisis de corriente directa y alterna, se generan las variables de tercer nivel, es decir las potencias consumidas por los elementos.

4.2.8 Variaciones del procedimiento en el Modo (Experto)

En el caso de que la simulación se esté ejecutando en el *Modo (Experto)*, existen algunas variantes en este procedimiento.

Al inicio el usuario debe escoger qué tipo de análisis desea realizar. Las opciones son análisis en corriente directa (DC), análisis en corriente alterna (AC), análisis en dominio de la frecuencia (FD) y análisis transitorio (TR).

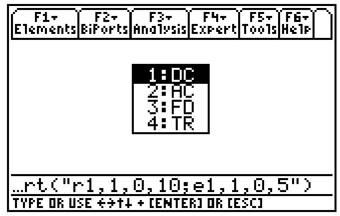

Figura 20. Pantalla ilustrativa, en la TI-89

El usuario escoge el análisis deseado, y presiona ¸ . En este punto, la calculadora mostrará un formulario para que el usuario establezca dos parámetros relacionados con la manera en que la calculadora manejará los números de punto flotante, que son los siguientes:

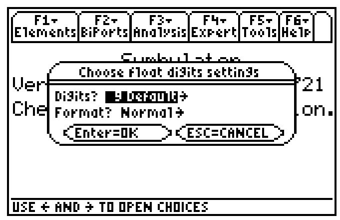

Figura 21. Pantalla ilustrativa, en la TI-89

- Qué nivel de precisión desea en los cálculos. En las simulaciones normales, este valor se ajusta en 9 dígitos. Se recomienda usar 9, a menos que por alguna razón se desee mayor o menor precisión.

- Qué formato desea darle a los números de punto flotante. En las simulaciones normales, se escoge formato Normal. Si lo desea, el usuario puede cambiar este formato a Ingeniería o Científico.

Tras haber seleccionado los valores deseados, el usuario presiona ¸ . En este momento, el *Symbulator* ejecutará el primer paso de la simulación, que hemos detallado arriba. Antes de llegar al segundo paso, el programa hace una pausa y muestra un formulario en la pantalla, que le ofrece al usuario la oportunidad de manipular las ecuaciones e incógnitas de primer nivel. El usuario puede modificarlas o no, según su necesidad.

Esta pantalla se llama *Primer Nivel del Modo (Experto)*. En ella aparecen cuatro campos y un menú de opciones.

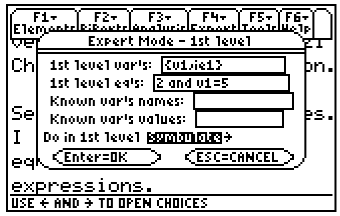

Figura 22. Primer Nivel del Modo (Experto), en la TI-89

El primer campo muestra las incógnitas de primer nivel. Si hay más de una incógnita, las incógnitas aparecerán separadas con comas, y estarán encerradas en llaves `{}`.

El segundo campo muestra las ecuaciones de primer nivel. Si hay más de una ecuación, las ecuaciones aparecerán unidas por operadores `and`.

El tercero y cuarto campos son utilizados cuando queremos especificar el valor de alguna variable de primer nivel, en el caso de que este valor ya sea conocido.

El tercer campo es para colocar el nombre de alguna variable de primer nivel cuyo valor sea conocido. Estas variables se conocen como variables condicionales. Si hay más de una, los nombres se deben introducir separados por comas, así: `nombre1,nombre2,nombre3`

El cuarto campo es para declarar el valor de esta variable de primer nivel ya conocida, que hemos listado arriba. Estos valores declarados se conocen como ecuaciones condicionales. Si hay uno solo, su valor debe introducirse, por ejemplo, así: `v1=3.5`. Si hubiese más de una variable conocida, sus valores deben introducirse usando el operador `and`. Así: `v1=3.5 and v2=5 and ie1=-2`. Téngase claro que sólo pueden declararse aquí los valores conocidos de variables de primer nivel.

El menú de opciones que aparece debajo de estos cuatro campos ofrece tres opciones para que el usuario, ya sea que haya modificado las ecuaciones o no, pueda decidir qué camino desea seguir. Existen tres opciones: simular, simular y guardar las ecuaciones, y simplemente guardar las ecuaciones.

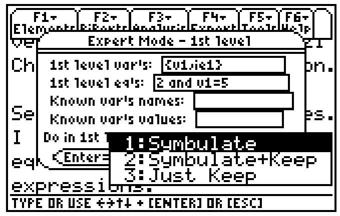

Figura 23. Las tres opciones, en la TI-89

Veamos a continuación en qué consiste cada una.

4.2.9 Sólo almacenar

Si el usuario escoge la tercera opción, `Just Keep`, o sea sólo guardar las ecuaciones, el Symbulator simplemente creará cuatro variables:

1) `Unknown`, en la cual se almacena la lista de las incógnitas de primer nivel.

2) `Equation`, en la cual se almacenan las ecuaciones de primer nivel.

3) `Wheneq`, en la cual se almacenan las ecuaciones condicionales.

4) `Whenvar`, en la cual se almacenan las variables condicionales.

Las cuatro variables se almacenan en forma de cadenas de texto. Cualquier modificación que haya hecho el usuario, aparecerá reflejada en estas variables. La idea de guardar estas ecuaciones es poder resolverlas luego con la herramienta `solves`.

En algunos casos excepcionales, si el circuito es muy grande, las ecuaciones generadas son más de lo que la calculadora puede presentar en la pantalla. De ser así, en vez de aparecer la pantalla del *Primer Nivel del Modo (Experto)*, se presentará un mensaje anunciando que las ecuaciones son muy grandes y que se almacenarán automáticamente. Sin embargo, en la mayoría de los casos, las ecuaciones generadas no serán muy voluminosas y se presentarán en la pantalla sin problemas.

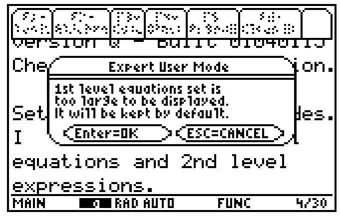

Figura 24. Anuncio del Modo (Experto), en la TI-89

Tras haber creado estas cuatro variables, el *Symbulator* mostrará una nueva pantalla, llamada *Segundo Nivel del Modo (Experto)*. En ella se preguntará al usuario si desea conservar las expresiones de segundo nivel que, como hemos dicho, ya se han generado.

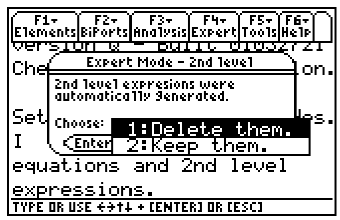

Figura 25. Segundo Nivel del Modo (Experto), en la TI-89

El usuario puede escoger entre borrar estas variables o conservarlas. Si decide borrarlas, el *Symbulator* las borrará y dará por terminado su trabajo. Si el usuario decide conservarlas, el *Symbulator* las conservará, y mostrará una nueva pantalla, llamada *Tercer Nivel del Modo (Experto)*, la cual preguntará al usuario si desea que genere las expresiones de tercer nivel.

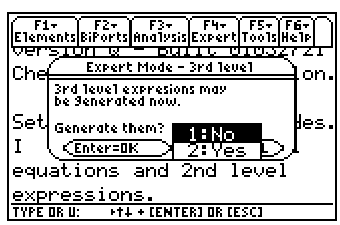

Figura 26. Tercer Nivel del Modo (Experto), en la TI-89

El usuario puede escoger entre generar estas expresiones o no. Si decide no generarlas, el *Symbulator* dará por terminado su trabajo. Si, por el contrario, decide generarlas, el *Symbulator* las generará y dará por terminado su trabajo.

4.2.10 Simular y almacenar

Si el usuario escoge la segunda opción, `Symbulate+Keep`, o sea simular y guardar las ecuaciones, el *Symbulator* creará las mismas cuatro variables mencionadas anteriormente, y seguidamente proseguirá con la simulación normalmente. Es decir, ejecutará los pasos 2, 3 y 4, como lo hubiese hecho normalmente, tomando en consideración los cambios que haya podido hacer el usuario.

4.2.11 Sólo simular

Si el usuario escoge la primera opción, `Symbulate`, o sea simular solamente, el *Symbulator* proseguirá con la simulación normalmente, sin crear las cuatro variables. Ejecutará los pasos 2, 3 y 4, tal y como lo hubiese hecho normalmente, tomando en consideración los cambios que haya podido hacer el usuario, obviamente.

Ahora que hemos aprendido la teoría, veamos todos estos nuevos conceptos aplicados en la práctica. Resolvamos los mismos problemas del capítulo anterior, pero esta vez como expertos.

4.3 Práctica

Problema N° 013 *con el Modo (Experto)*

Planteamiento. Determine los valores numéricos de v_x e i_x en el siguiente circuito.

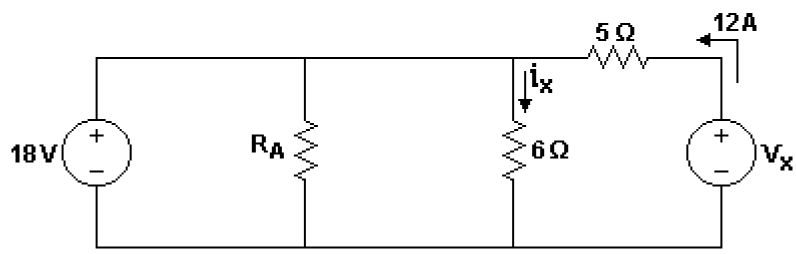

Figura 16. Circuito para el Problema N° 013.

Solución:

4.3.1 Pasos iniciales

Nombramos los nodos y elementos. Limpiamos los nombres simbólicos.

```
DelVar ra,vx
```

Damos la descripción del circuito y ordenamos la simulación experta.

```
sq\expert("e1,1,0,18;ra,1,0,ra;r1,1,0,6;r2,2,1,5;ex,2,0,
vx")
```

Iniciamos la simulación con ¸ . Como tipo de análisis, escogemos `DC` y presionamos ¸ . En cuanto a los formatos de punto flotante, los dejamos como están, y presionamos ¸ . Vemos en la pantalla las frases que nos indican que el *Modo (Experto)* está activado y que se ejecuta el primer paso. Aparece la pantalla del *Primer Nivel de Modo (Experto).* En ella podemos ver las ecuaciones que ha generado el *Symbulator*, y las incógnitas en función de las cuales las pretende resolver.

Recordemos que la resistencia `r5` fue definida de forma tal que su corriente coincida con la corriente de 12A que nos da el problema como dato conocido. Sin embargo, si leemos la lista de incógnitas de primer nivel que aparece en el primero de los campos, veremos que la corriente de esta resistencia no es una variable de primer nivel. Por otro lado, las corrientes de las dos fuentes de voltaje, `ie1` e `iex` sí son variables de primer nivel.

Como la corriente de la fuente `ex` está definida del nodo `2` al nodo `0`, podemos decir que la corriente `iex` es igual a `-12` amperios. El cambio de signo responde a que la corriente de la fuente está definida en el sentido contrario a la flecha del problema.

Ahora ha llegado el momento de analizar qué nos da el problema y qué nos pide. Vemos que el *Symbulator* ha generado para nosotros un sistema de 4 ecuaciones con 4 incógnitas, el cual también posee 1 valor simbólico. Podemos decir que este valor simbólico es una incógnita más, pues debe ser resuelto para poder obtener respuestas numéricas. Entonces, en verdad tenemos un sistema de 4 ecuaciones con 5 incógnitas. Podemos ver esta situación desde dos perspectivas: nos sobra una incógnita, o nos falta una ecuación.

Resolveremos este problema desde la primera perspectiva, y luego desde la segunda.

4.3.2 Eliminando una incógnita

Si nos sobra una incógnita, debemos eliminarla, para poder tener un sistema de 4 ecuaciones y 4 incógnitas.

Para eliminar esta incógnita del sistema, debemos hacer tres cosas. Primero, eliminar a `iex` de la lista de incógnitas de primer nivel que aparece en el primer campo. No debe aparecer en esta lista, pues es un valor conocido. Segundo, agregar a esta lista de incógnitas de primer nivel la variable `vx`, que de ahora en adelante será una incógnita de primer nivel. Y tercero, agregar el nombre de `iex` al tercer campo, y su valor conocido de `iex=-12` al cuarto campo, donde el signo – se escribe con la tecla · . Tal y como nos lo explica el manual de usuario, para escribir letras o números, debemos presionar **j** para activar y desactivar el modo alfabético y numérico de la calculadora. Tras hacer estas modificaciones en los campos, el contenido de estos debe ser como sigue:

- Incógnitas de primer nivel: `{v1,ie1,v2,vx}`
- Las ecuaciones de primer nivel no fueron modificadas.
- Nombres de variables conocidas: `iex`
- Valores de variables conocidas: `iex=-12`

Una vez hecho esto, presionamos ¸ . Las frases en la pantalla nos indican la ejecución de los tres pasos restantes de la simulación. Finalmente, aparece **Done**.

A diferencia de las técnicas anteriores, empleando el comando `solve` y la herramienta `solves`, esta simulación que hemos realizado con el *Modo (Experto)* no requiere más manipulación para obtener respuestas numéricas. Pues lo que hicimos al modificar la lista de incógnitas y agregar el valor conocido, fue simplemente reducir el sistema de 4 ecuaciones con 5 incógnitas, a un sistema de 4 ecuaciones con 4 incógnitas. Así, todas las respuestas que hemos obtenido, y que ahora nos esperan en la *carpeta actual*, serán respuestas numéricas.

Solicitando el valor de `ir1`, encontramos que i_x es **`3`** amperios. Para encontrar el valor de v_x, tenemos tres opciones: solicitar el voltaje del nodo `v2`, solicitar la caída de voltaje en la fuente de voltaje `vex`, o simplemente solicitar el valor de nuestra incógnita de primer nivel `vx`. Recuérdese que habíamos incluído `vx` en la lista de incógnitas de primer nivel, y por ello el *Symbulator* encontró su valor como si fuese cualquier otra incógnita de primer nivel. Por cualquiera de los tres caminos, encontramos que v_x es **`78`** voltios.

Usemos ahora la otra perspectiva desde la cual pudo haberse atacado este problema.

4.3.3 Agregando una ecuación.

Si nos falta una ecuación, debemos agregarla, para poder tener un sistema de 5 ecuaciones y 5 incógnitas.

Como queremos hacer este problema otra vez desde esta nueva perspectiva, ejecutamos todos los pasos del punto 4.3.1, hasta llegar a la pantalla de *Primer Nivel*.

```
DelVar ra,vx

sq\expert("e1,1,0,18;ra,1,0,ra;r1,1,0,6;r2,2,1,5;ex,2,0,
vx")
```

Iniciamos la simulación con ¸ . Como tipo de análisis, escogemos `DC` y presionamos ¸ . Dejamos los formatos de punto flotante como están, y presionamos ¸ . Vemos en la pantalla las frases que nos indican que el *Modo (Experto)* está activado y que se ejecuta el primer paso. Aparece la pantalla del *Primer Nivel de Modo (Experto).*

Para agregar una ecuación al sistema, debemos hacer dos cosas. Primero, agregar a la lista de incógnitas de primer nivel la variable `vx`, que de ahora en adelante será una incógnita de primer nivel. Y segundo, agregar una ecuación a la lista de ecuaciones de primer nivel, así: `and iex=-12`. Tras hacer esto, el contenido de los campos debe ser como sigue:

- Incógnitas de primer nivel: `{v1,ie1,v2,iex,vx}`
- Ecuaciones de primer nivel: `(5*ie1*ra-ra*(vx-33)+90)/ra=0 and 5*iex+vx=18 and v1=18 and v2=vx  and iex=-12`
- Nombres de variables conocidas: *en blanco.*
- Valores de variables conocidas: *en blanco.*

Hemos aumentado el sistema de 4 ecuaciones con 5 incógnitas, a un sistema de 5 ecuaciones con 5 incógnitas. Una vez hecho esto, presionamos ¸ . Las frases en la pantalla nos indican la ejecución de los tres pasos restantes de la simulación. Finalmente, aparece **Done**, y en la *carpeta actual* nos esperan nuestras respuestas numéricas.

Solicitando el valor de `ir1`, encontramos que i_x es **3** amperios. Solicitando el valor de `v2`, `vx` o `vex`, encontramos que v_x es **78** voltios.

Este problema ilustra que el *Modo (Experto)* puede ahorrar tiempo al usuario experto, pero requiere buenos conocimientos sobre circuitos, álgebra y el uso del *Symbulator*.

Veamos ahora otro problema del capítulo anterior.

Problema N° 014 *con el Modo (Experto)*

Planteamiento. Determine los valores numéricos de v_x e i_x en el siguiente circuito.

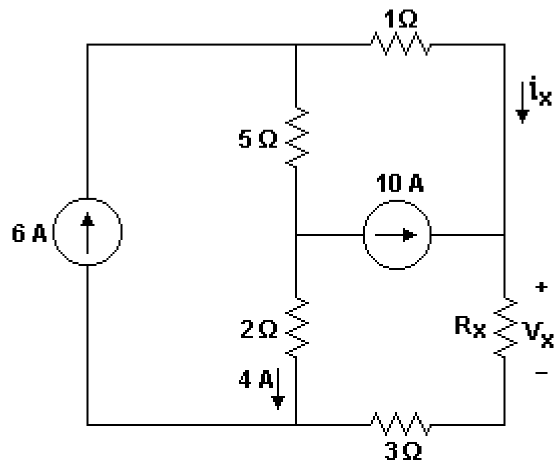

Figura 17. Circuito para el Problema N° 014.

Solución:

Nombramos los nodos y elementos. Limpiamos los nombres simbólicos.

```
DelVar rx
```

Damos la descripción del circuito y ordenamos la simulación experta.

```
sq\expert("j1,0,1,6;r1,1,2,1;r2,1,3,5;j2,3,2,10;r3,3,0,
2;rx,2,4,rx;r4,4,0,3")
```

Iniciamos la simulación con ¸ . Como tipo de análisis, escogemos `DC` y presionamos ¸ . Dejamos los formatos de punto flotante como están y presionamos ¸ . Vemos en la pantalla las frases que nos indican que el *Modo (Experto)* está activado y que se ejecuta el primer paso. Aparece la pantalla del *Primer Nivel de Modo (Experto).*

Tenemos un sistema de 4 ecuaciones con 5 incógnitas (la quinta incógnita es el valor simbólico de `rx`, que requiere ser resuelto para obtener valores numéricos). Resolveremos este problema con la técnica de agregar una ecuación al sistema, pues es el camino más simple. Para convertirlo en un sistema de 5 ecuaciones con 5 incógnitas, agregamos una ecuación al sistema. Primero, agregamos a la lista de incógnitas de primer nivel la variable `rx`, que de ahora en adelante será una incógnita de primer nivel. Y segundo, agregamos una ecuación a la lista de ecuaciones de primer nivel, así: `and ir3=4`. Tras hacer esto, el contenido de los campos debe ser como sigue:

- Incógnitas de primer nivel: `{v1,v2,v3,v4,rx}`
- Ecuaciones de primer nivel: `(rx*(v1-v2+10)-v2+v4)/rx=0 and (rx*v4-3*(v2-v4))/rx=0 and 6*v1-5*v2-v3-30=0 and 2*v1-7*v3-100=0 and ir3=4`
- Nombres de variables conocidas: *en blanco.*
- Valores de variables conocidas: *en blanco.*

Aquí hay que hacer una aclaración sobre lo que es lícito en el *Modo (Experto)* y lo que no lo es. Nótese que ahora `ir3` forma parte de las ecuaciones. Pero, ¿acaso es `ir3` una incógnita de primer nivel? Claro que no. Es una variable de segundo nivel. Pero a este punto ya se ha ejecutado el paso 1 de la simulación, en el cual se han generado expresiones de segundo nivel que expresan las variables de segundo nivel en función de variables de primer nivel. Por lo tanto, es lícito incluir en las ecuaciones de primer nivel a variables de segundo nivel, pues éstas se encuentran expresadas internamente en función de incógnitas de primer nivel.

Por otro lado, no es lícito incluir en el tercero y cuarto campos, variables de segundo nivel cuyo valor sea conocido. Pues el tercer y cuarto campo son para especificar el valor conocido de variables de primer nivel solamente.

Volvamos al problema. Una vez hemos aumentado el sistema de 4 ecuaciones con 5 incógnitas, a un sistema de 5 ecuaciones con 5 incógnitas, presionamos ¸ . Las frases en la pantalla nos indican la ejecución de los tres pasos restantes de la simulación. Finalmente, aparece **`Done`**, y en la *carpeta actual* nos esperan nuestras respuestas numéricas.

Solicitando el valor de `ir1`, encontramos que i_x es **`-8`** amperios. Solicitando el valor de `vrx`, encontramos que v_x es **`80`** voltios.

Es posible resolver este problema con la otra técnica, la de eliminar una incógnita, pero hubiese resultado en un procedimiento un poco más complicado. Tendríamos que agregar el valor de alguna variable de primer nivel como valor conocido, en el tercer y cuarto campos. Para expresar la corriente de 4 amperios como una incógnita de primer nivel, hubiesemos tenido que relacionar el voltaje del nodo 3 y el valor de la resistencia de 2 ohmios mediante la ley de Ohm, para despejar el valor de `v3` así: $v_3 = (4\text{ A})(2\Omega)$ o sea `v3=8`. El procedimiento, pues, sería como sigue.

```
DelVar rx

sq\expert("j1,0,1,6;r1,1,2,1;r2,1,3,5;j2,3,2,10;r3,3,0,
2;rx,2,4,rx;r4,4,0,3")
```

Iniciamos la simulación con ¸ . Como tipo de análisis, escogemos `DC` y presionamos ¸ . Dejamos los formatos de punto flotante como están y presionamos ¸ . Vemos en la pantalla las frases que nos indican que el *Modo (Experto)* está activado y que se ejecuta el primer paso. Aparece la pantalla del *Primer Nivel de Modo (Experto).*

Para eliminar una incógnita del sistema, debemos hacer dos cosas. Primero, eliminar a `v3` de la lista de incógnitas de primer nivel que aparece en el primer campo. No debe aparecer en esta lista, pues es un valor conocido. Segundo, agregar a esta lista de incógnitas de primer nivel la variable `rx`, que de ahora en adelante será una incógnita de primer nivel. Y tercero, agregar el nombre de `v3` al tercer campo, y su valor conocido de `v3=8` al cuarto campo. Tras hacer estas modificaciones, el contenido de los campos debe ser como sigue:

- Incógnitas de primer nivel: `{v1,v2,rx,v4}`
- Las ecuaciones de primer nivel no fueron modificadas.
- Nombres de variables conocidas: `v3`
- Valores de variables conocidas: `v3=8`

Una vez hecho esto, presionamos ¸ . Las frases en la pantalla nos indican la ejecución de los tres pasos restantes de la simulación. Finalmente, aparece **`Done`**.

Solicitando el valor de `ir1`, encontramos que i_x es **`-8`** amperios. Solicitando el valor de `vrx`, encontramos que v_x es **`80`** voltios.

En este problema hemos visto que, cuando sea difícil declarar un valor previamente conocido como una variable de primer nivel, el método de agregar una ecuación es más fácil que el método de eliminar una incógnita. Sin embargo, el método de eliminar una incógnita tiene la ventaja de que el sistema de ecuaciones se reduce y puede ser resuelto más fácilmente por la calculadora. Esto cobra importancia en el caso de circuitos grandes, mucho mayores a los que vemos en estas páginas.

<u>Problema N° 015</u> *con el Modo (Experto)*

Planteamiento. Determine los valores numéricos de v_x e i_x en el siguiente circuito.

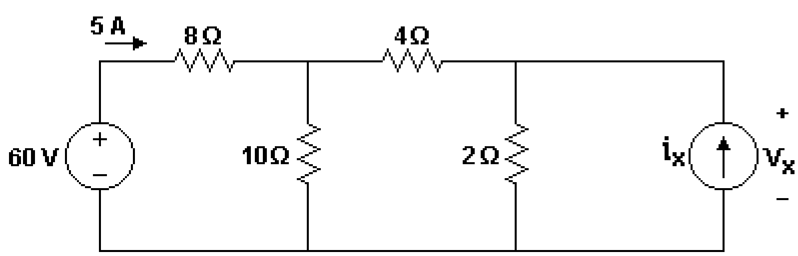

Figura 18. Circuito para el Problema N° 015.

Solución:

Nombramos los nodos y elementos. Limpiamos el nombre i_x.

```
DelVar ix
```

Damos la descripción del circuito y ordenamos la simulación experta.

```
sq\expert("e1,1,0,60;r1,1,2,8;r2,2,0,10;r3,2,3,4;r4,3,0
,2;jx,0,3,ix")
```

Iniciamos la simulación con ¸ . Como tipo de análisis, escogemos `DC` y presionamos ¸ . Dejamos los formatos de punto flotante como están y presionamos ¸ . Vemos en la pantalla las frases que nos indican que el *Modo (Experto)* está activado y que se ejecuta el primer paso. Aparece la pantalla del *Primer Nivel de Modo (Experto)*. Este problema, al igual que los anteriores, puede ser resuelto agregando una ecuación o eliminando una incógnita.

Para resolverlo con la técnica de agregar una ecuación, el contenido de los campos debe modificarse para tener un sistema de 5 ecuaciones con 5 incógnitas, como sigue:

- Incógnitas de primer nivel: `{v1,ie1,v2,v3,ix}`
- Ecuaciones de primer nivel: `8*ie1-v2+60=0 and 4*ix+v2-3*v3=0 and v1=60 and 19*v2-10*(v3+30)=0 and ir1=5`
- Nombres de variables conocidas: *en blanco.*
- Valores de variables conocidas: *en blanco.*

Por otro lado, para resolverlo con la técnica de eliminar una incógnita, el contenido de los campos debe modificarse para tener un sistema de 4 ecuaciones con 4 incógnitas, con el valor de una variable de primer nivel declarado como conocida, así:

- Incógnitas de primer nivel: `{v1,v2,v3,ix}`
- Las ecuaciones de primer nivel no fueron modificadas.
- Nombres de variables conocidas: `ie1`
- Valores de variables conocidas: `ie1=-5`

Escoja el lector la técnica que prefiera entre estas dos. Luego, presionamos ¸ . Las frases en la pantalla nos indican la ejecución de los tres pasos restantes de la simulación. Finalmente, aparece **`Done`**.

Solicitando el valor de `ix`, encontramos que i_x es **1** amperio. Solicitando el valor de `-vjx` o de `v3`, encontramos que v_x es **8** voltios. Veamos un último problema.

Problema N° 016 *con el Modo (Experto)*

Planteamiento. Determine el valor de *k* que provocará que el voltaje v_y sea cero.

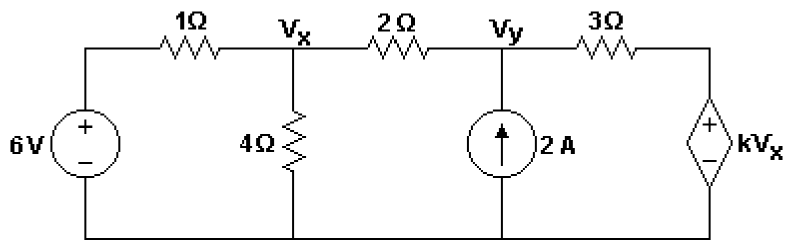

Figura 19. Circuito para el Problema N° 016.

Solución:

Nombramos los nodos y elementos. Limpiamos el nombre *k*.

```
DelVar k
```

Damos la descripción del circuito y ordenamos la simulación experta.

```
sq\experto("e1,1,0,6;r1,1,x,1;r2,x,0,4;r3,x,y,2;j1,0,y,
2;r4,y,2,3;e2,2,0,k*vx")
```

Iniciamos la simulación con ¸ . Como tipo de análisis, escogemos `DC` y presionamos ¸ . Dejamos los formatos de punto flotante como están y presionamos ¸ . Vemos en la pantalla las frases que nos indican que el *Modo (Experto)* está activado y que se ejecuta el primer paso. Aparece la pantalla del *Primer Nivel de Modo (Experto)*. Este problema puede ser resuelto agregando una ecuación o eliminando una incógnita.

Para resolverlo agregando una ecuación, el contenido de los campos debe modificarse para tener un sistema de 7 ecuaciones con 7 incógnitas, como sigue:

- Incógnitas de primer nivel: `{v1,ie1,vx,vy,v2,ie2,k}`
- Ecuaciones de primer nivel: `2*k*vx+3*vx-5*vy+12=0 and k*vx+3*ie2-vy=0 and ie1-vx+6=0 and v1=6 and v2=k*vx and 7*vx-2*(vy+12)=0 and vy=0`
- Nombres de variables conocidas: *en blanco.*
- Valores de variables conocidas: *en blanco.*

Por otra parte, para resolverlo eliminando una incógnita, el contenido de los campos debe modificarse para tener un sistema de 6 ecuaciones con 6 incógnitas, con el valor de una variable de primer nivel declarado como conocida, así:

- Incógnitas de primer nivel: `{v1,ie1,vx,v2,ie2,k}`
- Las ecuaciones de primer nivel no fueron modificadas.
- Nombres de variables conocidas: `vy`
- Valores de variables conocidas: `vy=0`

Escoja el lector la técnica que prefiera entre estas dos. Luego, presionamos ¸ . Las frases en la pantalla nos indican la ejecución de los tres pasos restantes de la simulación. Finalmente, aparece **`Done`**. Solicitando el valor de `k`, encontramos que es **`-13/4`**.

4.4 La herramienta `solves` *tras la simulación experta*

La herramienta `solves` puede ser utilizada en cualquier momento. En el capítulo anterior mostramos su uso tras una simulación simbólica. En ese momento, no existían en la *carpeta actual* las variables `Equation` y `Unknown`. Cuando ejecutamos `solves` por primera vez, los campos de la herramienta aparecían en blanco, sin ningún contenido, y el usuario ingresaba manualmente las ecuaciones que deseaba resolver.

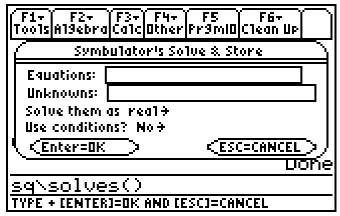

Figura 27. Campos en blanco de `solves`*, en la TI-89*

Las siguientes veces que la ejecutábamos, si no se había borrado el contenido de la *carpeta actual*, los campos aparecían mostrando inicialmente aquellas ecuaciones que habían sido ingresadas manualmente por el usuario la vez anterior.

Hasta ahora hemos realizado nuestras simulaciones expertas utilizando la primera opción del *Modo (Experto)*, solamente simular, sin guardar las ecuaciones. Por eso, no se crean en la *carpeta actual* las variables `Equation` y `Unknown`.

Si, por el contrario, hubiésemos utilizado cualquiera de las otras dos opciones, es decir simular y guardar las ecuaciones, o simplemente guardar las ecuaciones, entonces tendríamos en la *carpeta actual* las variables `Equation` y `Unknown`.

La herramienta `solves` se comporta de manera distinta cuando se le ejecuta justo después de una simulación experta que ha guardado las ecuaciones generadas. En este caso, los campos no aparecen vacíos, sino que muestran las ecuaciones e incógnitas que fueron almacenadas por el *Modo (Experto)* en las variables `Equation` y `Unknown`.

Por ejemplo, si simulamos un circuito en *Modo (Experto)* y escogemos la opción `Just Keep`, el *Symbulator* almacenará las ecuaciones y las incógnitas, y nosotros podemos luego resolverlas con la herramienta `solves`. Nótese que si se pretende insertar alguna variable de segundo nivel en las ecuaciones de primer nivel, debe solicitarse al *Modo (Experto)* que conserve las expresiones de segundo nivel que generó el *Symbulator*.

Al ejecutar la herramienta `solves` después de una simulación experta, en los campos aparecerían las ecuaciones e incógnitas, así:

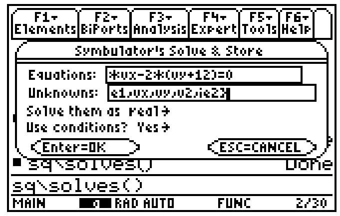

Figura 28. Ecuaciones en campos de `solves`*, en la TI-89*

La herramienta `solves` muestra dos menús de selección múltiple en su parte inferior. El superior es para que el usuario especifique si desea resolver las ecuaciones como reales o como complejas. Excepto si se están resolviendo las ecuaciones de un análisis AC, debe solicitarse siempre que las resuelva como reales. El menú inferior es para especificar si se desean usar condiciones. ¿Qué son las condiciones? No son más que los valores ya conocidos de variables de primer nivel. Si pedimos usar condiciones, la herramienta nos mostrará una nueva pantalla para introducir estos valores y sus nombres.

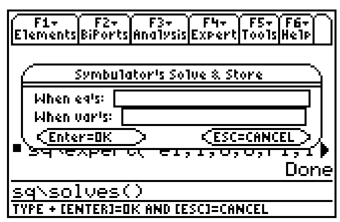

Figura 29. Campos para valores conocidos, en la TI-89

En el campo etiquetado como `When eq’s`, deben introducirse los valores conocidos de las variables de primer nivel. En el campo etiquetado como `When var’s`, deben introducirse los nombres de estas variables de primer nivel.

4.5 Verificación de las respuestas numéricas de una simulación experta

Para verificar si son correctas las respuestas numéricas que hemos obtenido tras nuestra simulación experta, podemos utilizar el mismo procedimiento que explicamos en el punto 3.7.

Antes de cerrar este capítulo, quiero felicitar al lector por haber decidido convertirse en un experto en el uso del *Symbulator*. Ahora que ya conoce los conceptos teóricos, falta que los utilice en la práctica durante algún tiempo. La recompensa, en satisfacciones y ahorro de tiempo, no se hará esperar.

Capítulo 5
Equivalentes de Thévenin y Norton

5.1 Introducción a la herramienta `thevenin`

Los teoremas de Thévenin-Helmholtz y Norton-Mayer son parte importante en los cursos de teoría de circuitos. El *Symbulator* posee una herramienta especial para encontrar los equivalentes de Thévenin y Norton de una red. Esta herramienta se llama `thevenin`, y se le ejecuta así:

```
thevenin(red,nodo 1,nodo 2)
```

Como podemos ver, toma tres datos de entrada: la descripción de la red en formato de texto, y los nombres de los dos nodos entre los cuales se desea encontrar el equivalente.

Como datos de salida, nos entrega tres variables almacenadas en la *carpeta actual*. Estas variables son la impedancia Thévenin `zth`, el voltaje Thévenin `vth` y la corriente Norton `ino`. Con estos tres valores, el usuario puede entonces construir el equivalente Thévenin colocando una impedancia de valor `zth` en serie con una fuente de voltaje de valor `vth`, y el equivalente Norton colocando una impedancia de valor `zth` en paralelo con una fuente de corriente de valor `ino`.

En algunos casos, como veremos más adelante, la herramienta también nos da como respuesta el valor de la máxima potencia que la red puede entregar.

5.2 Teoría

Tras haber introducido en la línea de entrada la orden de ejecutar la herramienta, con sus respectivos datos de entrada, aparecerá una pantalla en la cual el usuario debe escoger dos parámetros:

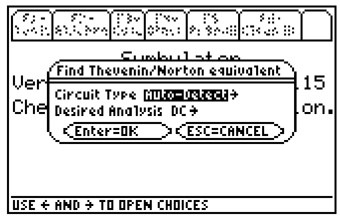

Figura 30. Herramienta `thevenin`*, en la TI-89*

- Qué tipo de red se desea analizar. Las opciones son *Auto-Detectar* y *Pasiva*. Debe usarse *Pasiva* si la red no posee fuentes independientes, y *Auto-Detectar* cuando las posea o cuando el usuario no esté seguro.

Es importante escoger bien el tipo de red que se va a analizar. Si escogemos *Auto-Detectar*, la herramienta ejecutará dos simulaciones, sin importar el tipo de red, y obtendremos los equivalentes correctos. Escoger *Pasiva* para aquellas redes que lo sean, tiene la ventaja de que la herramienta nos ahorrará tiempo, pues obtendrá los equivalentes correctos con una sola simulación. Sin embargo, si por error escogemos *Pasiva* y la red no lo es, obtendremos respuestas equivocadas. Nunca debe escogerse *Pasiva* si la red no es pasiva, o si tenemos dudas. Recuérdese que una red pasiva es aquella que no tiene fuentes independientes.

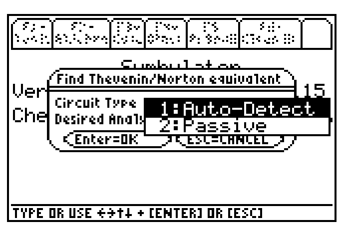

Figura 31. Tipo de red, en la TI-89

- Qué tipo de análisis desea realizar. Las opciones son análisis en corriente directa (DC), análisis en corriente alterna (AC) y análisis en dominio de la frecuencia (FD).

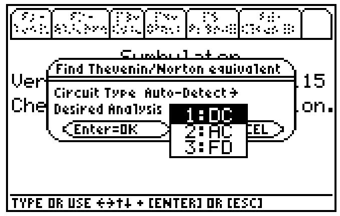

Figura 32. Tipo de análisis, en la TI-89

Tras haber seleccionado los valores deseados, el usuario presiona ¸ . En este momento, el *Symbulator* ejecutará una o dos simulaciones, cuyas frases descriptivas veremos en la pantalla. Luego aparecerá una pantalla de respuestas. En esta pantalla de respuestas, el programa nos mostrará, uno a uno, los valores encontrados: la impedancia Thévenin `zth`, el voltaje Thévenin `vth` y la corriente Norton `ino`. Para ver el siguiente valor, el usuario presiona ¸ . Al final, cuando todos los valores han sido presentados, aparece **`Done`**. Almacenadas en la *carpeta actual*, están todas estos valores en variables con los mismos nombres.

En el caso de que se utilice *Pasivo*, la herramienta no mostrará en la pantalla los valores de `vth` e `ino`, pues siempre serán cero.

En el caso de que se utilice *Auto-Detectar*, la herramienta también entregará, como respuesta a los análisis en corriente directa, la máxima potencia real que puede entregar la red, en la pantalla de respuestas y en una variable llamada `pmax`; y como respuesta a los análisis en corriente alterna, la máxima potencia compleja que puede entregar la red, en la pantalla de respuestas y en una variable llamada `smax`.

5.3 Práctica

Veamos algunos problemas.

Problema N° 017

Planteamiento. Determine los equivalentes de Thévenin y Norton para aquella parte del circuito que está a la izquierda de R_L.

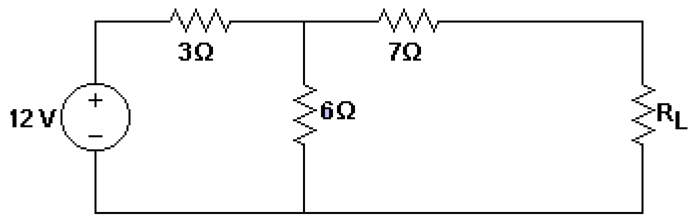

Figura 33. Circuito para el Problema N° 017.

Solución:

Nombramos los nodos y elementos de la parte del circuito que está a la izquierda de R_L. El elemento R_L no desempeña ningún papel en nuestra simulación. Nombramos a los nodos de izquierda a derecha, así: `1`, `2` y `3`. El nodo inferior es el de referencia. Por lo tanto, los nodos entre los cuales deseamos encontrar los equivalentes son el nodo `3` y el nodo `0`. Damos la descripción de la parte del circuito que está a la izquierda de R_L. Ejecutamos la herramienta `thevenin`.

```
sq\thevenin("e1,1,0,12;r1,1,2,3;r2,2,0,6;r3,2,3,7",3,0)
```

Iniciamos la simulación con ¸ . Como la red que deseamos analizar posee una fuente independiente, escogemos `Auto-Detect` como tipo de circuito. Escogemos `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecutan dos simulaciones. En la pantalla de respuestas aparecen los valores de los equivalentes de Thévenin y Norton. Para `zth` obtenemos **`9`**. Para `vth` obtenemos **`8`**. Para `ino` obtenemos **`8/9`**. Con estos valores, el usuario puede construir los equivalentes de Thévenin y Norton de la red en cuestión. En cuanto a la máxima potencia que puede entregar la red, para `pmax` obtenemos **`16/9`**. Finalmente, aparece **`Done`**.

Problema N°018

Planteamiento. Determine los equivalentes de Thévenin y Norton de la red frente al resistor de 1kΩ.

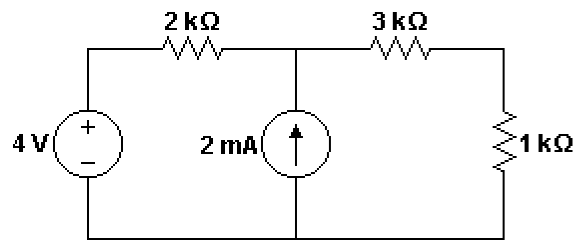

Figura 34. Circuito para el Problema N° 018.

Solución:

Nombramos los nodos y elementos de la red que está a la izquierda del resistor de 1kΩ. Este resistor no desempeña ningún papel en nuestra simulación. Nombramos a los nodos de izquierda a derecha, así: `1`, `2` y `3`. El nodo inferior es el de referencia. Por lo tanto, los nodos entre los cuales deseamos encontrar los equivalentes son el nodo `3` y el nodo `0`. Damos la descripción de la parte del circuito que nos interesa. Ejecutamos la herramienta.

```
sq\thevenin("e1,1,0,4;r1,1,2,2E3;j1,0,2,2E-3;r2,2,3,3E3",3,0)
```

Iniciamos la simulación con ¸ . La red que deseamos analizar posee fuentes independientes, así que escogemos `Auto-Detect` como tipo de circuito. Escogemos `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecutan dos simulaciones. En la pantalla recibimos los valores de los equivalentes de Thévenin y Norton. Para `zth` obtenemos **`5000.`** ohmios, para `vth` obtenemos **`8.`** voltios, para `ino` obtenemos **`.0016`** amperios, y para `pmax` obtenemos **`.0032`** vatios. Finalmente, aparece **`Done`**.

Problema N° 019

Planteamiento. Determine los equivalentes de Thévenin y Norton de la red que se muestra en la figura.

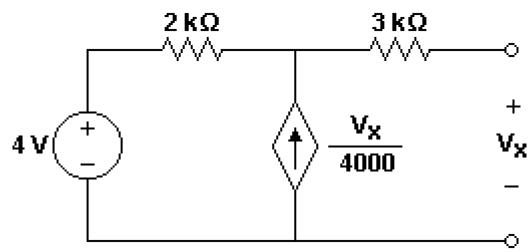

Figura 35. Circuito para el Problema N° 019.

Solución:

Nombramos los nodos y elementos de la red. Nombramos a los nodos de izquierda a derecha, así: `1`, `2` y `x`. El nodo inferior es el de referencia. Por lo tanto, los nodos entre los cuales deseamos encontrar los equivalentes son el nodo `x` y el nodo `0`. Damos la descripción de la red. Ejecutamos la herramienta.

```
sq\thevenin("e1,1,0,4;r1,1,2,2E3;j1,0,2,vx/4E3;r2,2,x,3
E3",x,0)
```

Iniciamos la simulación con ¸ . La red que deseamos analizar posee una fuente independiente, así que escogemos `Auto-Detect` como tipo de circuito. Escogemos `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecutan dos simulaciones. Luego aparecen los valores de los equivalentes de Thévenin y Norton. Para `zth` obtenemos **`10000.`** ohmios, para `vth` obtenemos **`8.`** voltios, para `ino` obtenemos **`.0008`** amperios, y para `pmax` obtenemos **`.0016`** vatios. Finalmente, aparece **`Done`**.

Problema N° 020

Planteamiento. Encuentre: 1) R_{Eq} si R=80Ω, 2) R si R_{Eq}=80Ω, y 3) R si R= R_{Eq}.

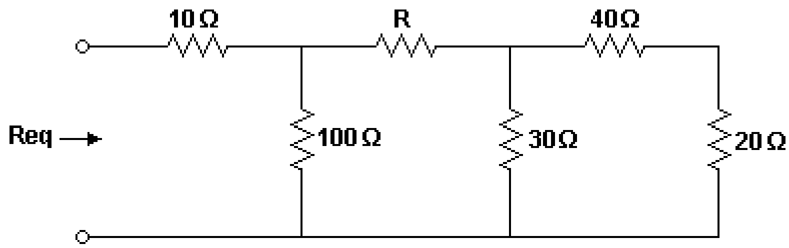

Figura 36. Circuito para el Problema N° 020.

Solución:

Para resolver este problema, usaremos la herramienta `thevenin` y algunos comandos. Nombramos los nodos y elementos de la red. Nombramos a los nodos de izquierda a derecha, así: `1`, `2`, `3` y `4`. El nodo inferior es el de referencia. Por lo tanto, los nodos entre los cuales buscaremos la resistencia equivalente son el nodo `1` y el nodo `0`. Damos la descripción de la red. Ejecutamos la herramienta.

```
sq\thevenin("r1,1,2,10;r2,2,0,100;r,2,3,r;r3,3,0,30;r4,
3,4,40;r5,4,0,20",1,0)
```

Iniciamos la simulación con ¸ . La red que deseamos analizar no posee fuentes independientes, así que escogemos `Passive` como tipo de circuito. Escogemos `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Para `zth`, obtenemos una función de `r`, como era de esperarse. Finalmente, aparece **`Done`**. Se nos ha pedido encontrar: 1) R_{Eq} si R=80Ω, 2) R si R_{Eq}=80Ω, y 3) R si R= R_{Eq}. Para encontrar estas respuestas, usaremos el comando `solve` y el comando | .

Recordemos que `zth` está almacenada en la memoria. Para obtener la primera respuesta, escribimos en la *línea de entrada* lo siguiente:

```
zth|r=80
```

Aquí estamos preguntando: *¿Cuánto vale* `zth` *si* `r=80`*?* Lo preguntamos de esta manera, y no con `solve`, porque `zth` es función de `r`, y no a la inversa. Presionamos ¸ . Obtenemos **`60`**.

Para obtener la segunda respuesta, aprovechamos que `zth` es función de `r`, y escribimos en la *línea de entrada* lo siguiente:

```
solve(zth=80,r)
```

Aquí estamos preguntando: *¿Cuánto vale* `r` *si* `zth=80`*?* Presionamos ¸ . Obtenemos **`r=640/3`**.

Para obtener la tercera respuesta, escribimos en la *línea de entrada* lo siguiente:

```
solve(zth=r,r)
```

Aquí estamos preguntando: *¿Cuánto vale* `r` *si* `zth=r`*?* Presionamos ¸ . Obtenemos dos respuestas. Una de ellas es positiva y la otra negativa. Esto significa que la ecuación tiene más de una solución. Obviamente, no existen resistencias negativas, así que limitaremos el rango que deseamos como respuestas con un operador condicional. Igualmente, aprovecharemos para solicitar la respuesta en forma aproximada.

```
solve(zth=r,r)|r>0
```

Aquí estamos preguntando: *¿Cuánto vale* `r` *si* `zth=r` *y* `r` *es mayor que 0?* Presionamos **¥** y ¸ . Obtenemos **`r=51.7890835`**.

Estas son las respuestas correctas.

Hagamos una pausa y conozcamos una nueva herramienta.

5.4 La herramienta `par`

La herramienta `par` es una función, y fue escrita en su primera versión por mi amigo belga Erwin Baert, para una de las versiones tempranas del *Symbulator*, como una herramienta sencilla y rápida para encontrar el equivalente de dos resistencias conectadas en paralelo. Aunque luego introduje algunas mejoras en el algoritmo, la función sigue siendo muy sencilla en su naturaleza y uso.

Para encontrar el equivalente de dos resistencias, digamos una de 3Ω y otra de 5Ω, que están conectadas en paralelo, se escribe `sq\par(3,5)`.

Esta herramienta se puede utilizar ya sea sola o incluida en una descripción de circuito.

5.4.1 Verificación de la fórmula para reducir resistencias en paralelo

Para mostrar el uso de esta herramienta, verifiquemos la fórmula para reducir resistencias en paralelo. El equivalente de dos resistencias, digamos R_1 y R_2, que están conectadas en paralelo, se encuentra mediante la fórmula $R_{Eq} = R_1R_2/(R_1+R_2)$. Verifiquemos esta fórmula con la herramienta `par`.

`sq\par(r1,r2)`.

Obtenemos **`r1*r2/(r1+r2)`**. Esto es correcto.

5.4.2 Aplicación

Es fácil encontrar en la práctica aplicaciones sencillas para la herramienta `par`. Me interesa aquí mostrar un ejemplo de aplicación un poco más compleja. Resolveremos nuevamente el Problema N° 020, esta vez usando la herramienta `par` en vez de la herramienta `thevenin`.

Problema N°020

Planteamiento. Encuentre: 1) R_{Eq} si R=80Ω, 2) R si R_{Eq}=80Ω, y 3) R si R= R_{Eq}.

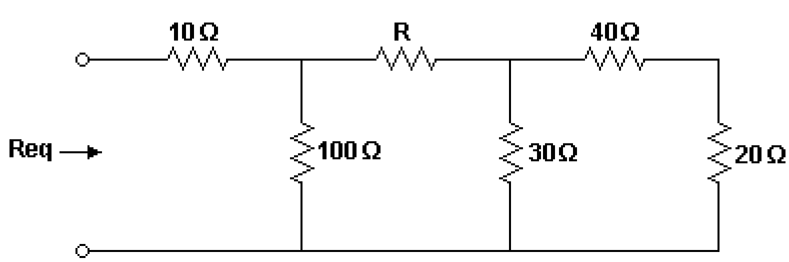

Figura 36. Circuito para el Problema N° 020.

Solución:

Definamos la resistencia equivalente `req`, como la suma de estas resistencias.

```
Define req=10+sq\par(100,r+sq\par(30,40+20))
```

Presionamos ¸ . Obtenemos una función de `r`, como era de esperarse.

```
req|r=80
```

Presionamos ¸ . Obtenemos **60**.

```
solve(req=80,r)
```

Presionamos ¸ . Obtenemos **r=640/3**.

```
solve(req=r,r)|r>0
```

Presionamos **¥** y ¸ . Obtenemos **r=51.7890835**.

Estas son las respuestas correctas. Hemos visto en este problema que la herramienta `par` puede ser muy útil para reducir algunas redes resistivas a su equivalente. Sin embargo, hay algunas redes resistivas que se resisten a ser reducidas con simplificaciones serie-paralelo, y que requieren forzosamente de la herramienta `thevenin`. Este es el caso de las redes que requieren transformaciones delta-estrella, o Δ-Y. Veamos un ejemplo.

Problema N°021

Planteamiento. Encuentre la resistencia equivalente de la red mostrada.

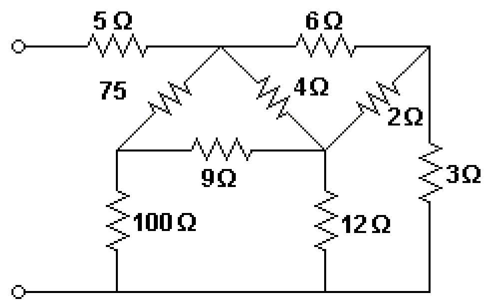

Figura 37. Circuito para el Problema N° 021.

Solución:

Manualmente, este problema requiere de transformaciones Δ-Y. Nosotros usaremos la herramienta `thevenin`. Nombramos los tres nodos superiores, de izquierda a derecha, así: `1`, `2` y `3`. Nombramos los dos nodos intermedios, de izquierda a derecha, así: `4` y `5`. El nodo inferior es el de referencia. Por lo tanto, los nodos entre los cuales buscaremos la resistencia equivalente son el nodo `1` y el nodo `0`. Nombramos los elementos, damos la descripción de la red y ejecutamos la herramienta.

```
sq\thevenin("r1,1,2,5;r2,2,3,6;r3,4,2,75;r4,2,5,4;r5,5,
3,2;r6,4,5,9;r7,3,0,3;r8,4,0,100;r9,5,0,12",1,0)
```

Iniciamos la simulación con ¸ . La red que deseamos analizar no posee fuentes independientes, así que escogemos `Passive` como tipo de circuito. Escogemos `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Como valor de la resistencia equivalente, `zth`, obtenemos **`6365/643`**. Finalmente, aparece **`Done`**. Podemos obtener un valor aproximado pidiendo en la línea de entrada `zth` y **¥** y ¸ , el cual sería **`9.89891135`**.

Nótese que, por tratarse de un circuito pasivo, escogimos la opción `Passive`. Esto implica que no se crea una variable `pmax`, pues un circuito pasivo no puede entregar potencia, por no tener fuentes independientes.

Hemos dicho que los circuitos que no tienen fuentes independientes son considerados pasivos. La presencia de fuentes dependientes no cambia este hecho. Veamos seguidamente cuatro problemas en los cuales los circuitos contienen únicamente fuentes dependientes y por lo tanto son pasivos.

Problema N° 022

Planteamiento. Encuentre el equivalente Thévenin de la red mostrada en la figura.

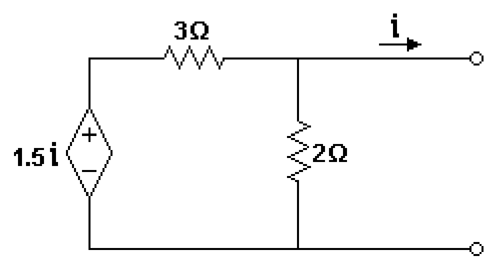

Figura 38. Circuito para el Problema N° 022.

Solución:

Nótese que la fuente dependiente está controlada por una corriente *i* que no atraviesa ningún elemento. Tenemos dos opciones. La primera es colocar un cortocircuito donde está *i*, para luego definir a *i* como la corriente a través de ese cortocircuito. La segunda es entender a *i* como la suma de las corrientes de las dos resistencias, las cuales tendrían que ser definidas en las direcciones apropiadas para que sus corrientes desemboquen hacia *i*. Usaremos esta segunda alternativa, pues no requiere un nuevo elemento.

Nombramos los dos nodos superiores, de izquierda a derecha, así: `1` y `2`. El nodo inferior es el de referencia. Por lo tanto, los nodos entre los cuales buscaremos la resistencia equivalente son el nodo `2` y el nodo `0`. Nombramos los elementos, damos la descripción de la red y ejecutamos la herramienta `thevenin`.

```
sq\thevenin("e1,1,0,1.5*(ir1+ir2);r1,1,2,3;r2,0,2,2",2,
0)
```

Nótese que ambas resistencias *"desembocan"* en el nodo 2. Iniciamos la simulación con ¸ . La red que deseamos analizar no posee fuentes independientes, así que escogemos `Passive` como tipo de circuito. Escogemos `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Para `zth` obtenemos **`.6`** ohmios. Finalmente, aparece **`Done`**.

Problema N°023

Planteamiento. Encuentre el equivalente Norton de la red mostrada en la figura.

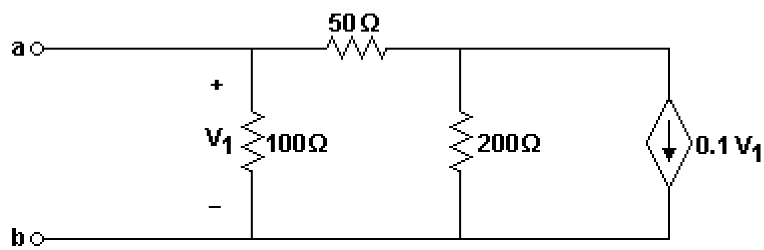

Figura 39. Circuito para el Problema N° 023.

Solución:

Nombramos los dos nodos superiores, de izquierda a derecha, así: `1` y `2`. El nodo inferior es el de referencia. Buscaremos la resistencia equivalente entre el nodo `1` y el nodo `0`. Nombramos los elementos, damos la descripción de la red y ejecutamos la herramienta `thevenin`.

```
sq\thevenin("r1,1,0,100;r2,1,2,50;r3,2,0,200;j1,2,0,0.1
*v1",1,0)
```

Iniciamos la simulación con ¸ . Escogemos `Passive` como tipo de circuito y `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Para `zth` obtenemos **`10.6382979`** ohmios.

Problema N° 024

Planteamiento. Encuentre la impedancia de entrada, Z_{ent}, de la red mostrada en la figura.

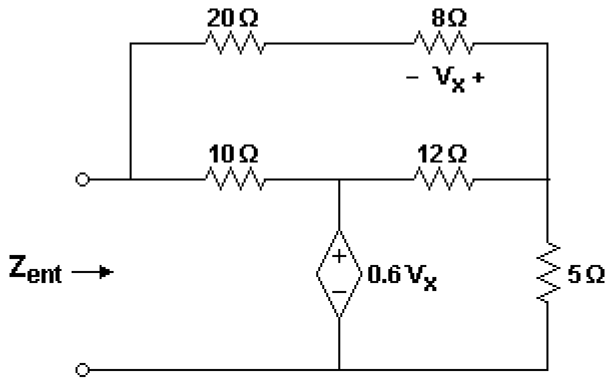

Figura 40. Circuito para el Problema N° 024.

Solución:

Los nodos de la malla superior fueron nombrados en el sentido de las manecillas del reloj, empezando en la izquierda, así: `1`, `2`, `3` y `4`. El nodo inferior es la referencia. Nombramos a la resistencia de 8 ohms como `rx`, y la definimos en la polaridad adecuada, entre `3` y `2`, para usar su caída de voltaje `vrx` como control para la fuente dependiente. Damos la descripción de la red y ejecutamos la herramienta `thevenin`, entre `1` y `0`.

```
sq\thevenin("r1,1,2,20;rx,3,2,8;r2,1,4,10;r3,4,3,12;r4,
3,0,5;e1,4,0,.6*vrx",1,0)
```

Iniciamos la simulación con ¸ . Escogemos `Passive` como tipo de circuito y `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Para `zth` obtenemos **`6.70508119`** ohmios.

Problema N° 025

Planteamiento. Encuentre el equivalente Thévenin de la red mostrada en la figura.

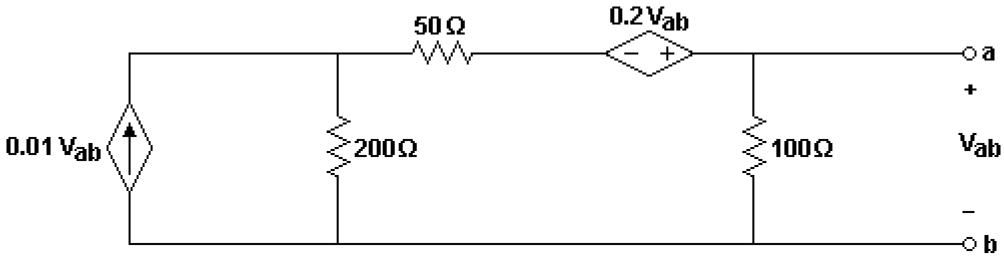

Figura 41. Circuito para el Problema N° 025.

Solución:

Ignoraremos los nombres *a* y *b* que muestra el dibujo. Nombramos los tres nodos superiores, de izquierda a derecha, así: `1`, `2` y `3`. El nodo inferior es el de referencia. Buscaremos la resistencia equivalente entre el nodo `3` y el nodo `0`. Nombramos los elementos, damos la descripción de la red y ejecutamos la herramienta `thevenin`.

```
sq\thevenin("j1,0,1,.01*v3;r1,1,0,200;r2,1,2,50;e1,3,2,
.2*v3;r3,3,0,100",3,0)
```

Iniciamos la simulación con ¸ . Escogemos `Passive` como tipo de circuito y `DC` como tipo de análisis. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Para `zth` obtenemos **`192.307692`** ohmios.

Consideramos que con los problemas mostrados, el lector tendrá en este momento un dominio completo de la herramienta `thevenin`.

Capítulo 6

Simulación numérica en corriente alterna

6.1 Fasores en la calculadora TI

En el análisis en corriente alterna, se utilizan fasores. La calculadora TI es una magnífica herramienta para manejar números complejos, ya sea en operaciones sencillas tales como suma, resta, multiplicación y división, o en operaciones realmente complicadas, tales como manipulación de matrices y resolución de ecuaciones simultáneas no lineales con términos simbólicos y complejos.

El operador imaginario se distingue en la calculadora TI con un símbolo distinto a la i minúscula, que luce algo así: ***i***, y se escribe con las teclas **2)** .

La calculadora puede presentar los números complejos en tres maneras: rectangular, polar y exponencial. La manera que usa en un momento dado depende de los modos de la calculadora en ese momento. Si el modo llamado Complex Format está ajustado en RECTANGULAR, los números complejos se presentarán en forma rectangular, así: `real+`***i***`maginario`. Si Complex Format está ajustado a POLAR, la manera de presentación dependerá entonces de otro modo, llamado Angle. Si el modo Angle está ajustado a RADIAN, los números complejos se presentarán en forma exponencial, así: `valor`$e^{\text{valor}\,i}$. Si por el contrario, Angle está ajustado a DEGREE, se presentarán en forma polar, así: `(magnitud ∠ ángulo en grados)`.

Para más información sobre las capacidades de la calculadora TI, sus modos de operación y las maneras en que puede presentar los números complejos, véase el Manual de Usuario.

Debido a necesidades propias de su operación interna, el *Symbulator* siempre ajusta estos dos modos, durante la simulación, a RECTANGULAR y RADIAN. Sin embargo, el *Symbulator* respeta los modos que el usuario tiene seleccionados antes de la simulación. Así que recordará cómo tenía el usuario ajustados estos modos antes de la simulación, y cuando haya terminado su trabajo, volverá a ajustarlos tal como los encontró en un principio.

Aunque la calculadora puede entregarnos los números complejos en tres presentaciones, normalmente los ingenieros eléctricos los manejamos solamente en dos de ellas: en forma rectangular y en forma polar. Con frecuencia, después de una simulación en corriente alterna, desearemos ver respuestas que están en estos dos formatos. Entonces surge una duda: si deseamos ver los números complejos tanto en forma rectangular como en forma polar, ¿cómo debemos ajustar los modos de la calculadora? La respuesta es que no hay ninguna combinación que permita ver los números en ambos formatos, porque la calculadora no puede adivinar cuándo el usuario los desea en una forma y cuándo en la otra.

Existen alternativas sencillas para ver los números complejos en presentación rectangular y en presentación polar, sin necesidad de cambiar los modos, y sin importar cómo se encuentran los mismos.

6.1.1 El comando `úRect`

Si queremos ver un número complejo en forma rectangular, sin importar cómo estén ajustados los modos de la calculadora, podemos usar el comando `úRect`. Este comando se puede encontrar en el catálogo de funciones de la calculadora.

El catálogo de funciones de la calculadora puede ser accesado a través de la tecla ½ en la TI-89, y de la combinación 2 2 en la TI-92Plus. Al accesar el catálogo, el usuario verá una lista de todos los comandos disponibles en la calculadora. Presionando la tecla [R], el usuario podrá saltar hasta la parte de esta lista en la cual se encuentran los comandos que empiezan con R. Ahí podrá ver el comando `úRect`. Seleccionandolo con el cursor y presionando ¸ podemos llevarlo hasta la *línea de entrada*.

Veamos cómo se utiliza este comando. Supongamos que tenemos una variable llamada `sr5`, que contiene un valor complejo. Queremos ver este valor en presentación rectangular, y para ello vamos a utilizar el comando `úRect`, porque los modos de la calculadora no están ajustados en la forma adecuada. Escribimos en la *línea de entrada*:

`sr5úRect`.

En el área de historia aparecerá el valor de `sr5` en presentación rectangular.

6.1.2 La herramienta `absang`

Existe un comando `úPolar`, que es el equivalente del comando `úRect` para presentación polar. Sin embargo, este comando nos presentará los números complejos en forma exponencial si el modo Angle está ajustado en RADIAN. Así, pues, no es una solución universal.

Mi amigo Erwin Baert programó para el primer *Symbulator* una pequeña función, llamada `absang`, cuyo algoritmo modifiqué para convertirla en una solución universal a la necesidad de presentar un número complejo en forma polar, sin importar cómo estén los modos de la calculadora. Veamos cómo se utiliza. Supongamos que tenemos una variable llamada `sr5`, que contiene un valor complejo. Queremos ver este valor en presentación polar, y como no queremos preocuparnos sobre cómo están ajustados los modos de la calculadora, vamos a utilizar la herramienta `absang`. Escribimos en la *línea de entrada*:

```
sq\absang(sr5)
```

En el área de historia aparecerá el valor de `sr5` en presentación polar, en formato de texto. Esta es una diferencia entre la herramienta `absang` y el comando `úRect`: el número se presenta en forma de texto.

Otra de las características de la herramienta `absang` es que no importa si el valor de entrada es exacto, siempre la salida será aproximada. Esto es así intencionalmente. Su utilidad se verá más adelante, en el Problema N° 031.

6.1.3 Mi recomendación personal

Personalmente, yo prefiero mantener el modo Complex Format ajustado en RECTANGULAR, para que así los números complejos siempre se presenten en forma rectangular. Y cuando deseo verlos en forma polar, utilizo la herramienta `absang`.

***6.2 Puerta para corriente alterna:* ac**

Para realizar una simulación en corriente alterna, se usa la *puerta* llamada **sq\ac**.

6.3 Datos de entrada: el circuito y la frecuencia radial

Al utilizar esta puerta es necesario suministrar, a manera de datos de entrada, el circuito que se quiere simular y la frecuencia angular, en radianes/segundo. Esta frecuencia es la que se designa usualmente como ω, y no debe confundirse con la frecuencia *f* que está dada en Hz. Para obtener ω a partir de *f*, se usa la fórmula $\omega=2\pi f$. La simulación en corriente alterna se ordena así: `sq\ac(circuito,ω)`.

6.4 Respuestas

En el caso de la simulación de corriente alterna, las respuestas son muy parecidas a las que obtenemos con la simulación en corriente directa: los voltajes en los nodos, las caídas de voltaje en los elementos, las corrientes en los elementos y las potencias consumidas por los elementos. Existen solamente dos diferencias. La primera es que en este caso las respuestas serán valores complejos. La segunda es que las potencias consumidas que nos entrega el simulador son las potencias complejas, en voltamperios

(VA). Estas potencias se almacenan en variables cuyo nombre es ***s*** más el nombre del elemento. Por ejemplo, si el elemento se llama ***r5***, el nombre de la variable en la cual se almacena la potencia compleja consumida por el elemento es ***sr5***.

6.5 Comandos `real`*,* `imag` *y* `conj`

La calculadora TI ofrece algunos comandos especiales para trabajar con números complejos. Veamos tres de ellos.

Para obtener la componente real de esta potencia compleja, y en general de cualquier número complejo, podemos utilizar el comando `real` de la calculadora, así:

```
real(sr5)
```

Por otro lado, para obtener la componente imaginaria de esta potencia compleja o de cualquier otro número complejo, podemos utilizar el comando `imag` de la calculadora, así:

```
imag(sr5)
```

Si quisiéramos encontrar el conjugado de una corriente compleja o de cualquier otro número complejo, podemos utilizar el comando `conj` de la calculadora, así:

```
conj(ir5)
```

Ha llegado el momento de conocer la descripción de algunos elementos usados en el análisis en corriente alterna.

6.6 Impedancias

Una impedancia es un elemento resistivo con un valor complejo dado en Ω. En las descripciones de circuito usadas en el *Symbulator*, las impedancias se describen exactamente igual que las resistencias, con la única diferencia de que sus valores serán complejos.

6.7 Admitancias

Una admitancia es un elemento conductivo con un valor complejo dado en siemens. En las descripciones de circuito usadas en el *Symbulator*, las admitancias se describen con la misma técnica usada para las conductancias, con la única diferencia de que sus valores serán complejos.

6.8 Capacitores

Un capacitor es un elemento de circuito que almacena energía en forma de un campo eléctrico y que se opone a cambios bruscos de voltaje.

6.8.1 Notación

En el caso del capacitor, la mínima información necesaria para describirlo por completo es la siguiente:

1. *¿Qué clase de elemento es?* Es un capacitor. La letra que identifica a los capacitores es la **c**.

2. *¿Cómo se llama el capacitor?* Para diferenciar un capacitor dado de cualquier otro capacitor, se le debe dar un nombre cualquiera, que empiece con **c**. Debe recordarse que la variables llamadas c#, en donde # es un número entre el 1 y el 99, son variables reservadas de la calculadora TI y no pueden ser utilizadas como nombre.

3. *¿Dónde se encuentra conectado el capacitor?* Un capacitor tiene dos extremos, cada uno de ellos conectado a un nodo distinto. Es necesario darle al simulador los nombres de los dos nodos a los cuales está conectado el capacitor. El orden de estos nodos establece la dirección que tendrá en la respuesta el flujo de corriente y la caída del voltaje a través del capacitor.

4. *¿Cuánto vale el capacitor?* Un capacitor tiene un valor. Este valor debe entregarse al *Symbulator* en faradios.

A veces nos encontramos en los libros de texto con problemas en corriente alterna, en los cuales aparecen capacitores cuyos valores están dados en Ω. Para efectos de la simulación, estos elementos son impedancias, y deben ser tratados como tales en la descripción del circuito, aunque aparezcan dibujados como capacitancias.

Esto tiende a confundir al usuario novato, quien trata por la fuerza de definir en el *Symbulator* estos elementos como capacitancias, pero declarando su valor en Ω. Esto es un error. Los elementos cuyos valores se vayan a declarar en ohmios deben ser tratados como impedancias. Para definirlos como capacitancias, habría que obtener primero su valor en faradios.

Debe tenerse la precaución de no introducir el valor de la capacitancia en una unidad distinta a los faradios (F). Por ejemplo, si el valor se introduce en microfaradios (μF), las respuestas estarán equivocadas.

Así, los capacitores se definen como se muestra a continuación:

c*nombre, nodo 1, nodo 2, valor en faradios*

En el análisis en corriente alterna, solamente se declaran estos cuatro términos. Veremos en capítulos posteriores que el análisis en dominio de la frecuencia y el análisis transitorio requieren un quinto término, para declarar el voltaje inicial del capacitor. Cuando el capacitor tenga un voltaje inicial, debemos colocarlo en el quinto término, y dar los nodos en el orden adecuado que indique la polaridad que deseamos darle a este voltaje inicial: el primer nodo con respecto al segundo nodo de la descripción. El dibujo del circuito nos indicará, con los signos de positivo y negativo, la polaridad que debemos darle al capacitor. Ejemplo:

c*nombre, nodo 1, nodo 2, valor, voltaje inicial*

Este voltaje inicial es captado por el simulador considerándolo como el voltaje del nodo 1 con respecto al nodo 2.

En el análisis en corriente directa, el *Symbulator* considera a los capacitores como circuitos abiertos.

6.8.2 Respuestas relacionadas

En el caso de un capacitor en análisis en corriente alterna, se entregan tres respuestas relacionadas: la corriente a través del capacitor, la caída de voltaje en el capacitor y la potencia compleja consumida por él. La corriente se almacena en una variable llamada `i`*nombre*, donde *nombre* es el del capacitor. La caída de voltaje se almacena en una variable llamada `v`*nombre*, donde *nombre* es el del capacitor. La potencia compleja consumida, por su parte, se almacena en una variable llamada `s`*nombre*, donde *nombre* es el del capacitor. Son aplicables a las respuestas de este elemento todas las convenciones antes expuestas sobre dirección, polaridad y consumo.

6.9 Inductores

Un inductor es un elemento de circuito que almacena energía en forma de un campo magnético y que se opone a cambios bruscos de corriente.

6.9.1 Notación

En el caso del inductor, la mínima información necesaria para describirlo por completo es la siguiente:

1. *¿Qué clase de elemento es?* Es un inductor. La letra que identifica a los inductores es la **l** (o sea L minúscula).

2. *¿Cómo se llama el inductor?* Para diferenciar un inductor dado de cualquier otro inductor, se le debe dar un nombre cualquiera, que empiece con **l**.

3. *¿Dónde se encuentra conectado el inductor?* Un inductor tiene dos extremos, cada uno de ellos conectado a un nodo distinto. Es necesario darle al simulador los nombres de los dos nodos a los cuales está conectado el inductor. El orden de estos nodos establece la dirección que tendrá en la respuesta el flujo de corriente y la caída del voltaje a través del inductor.

4. *¿Cuánto vale el inductor?* Un inductor tiene un valor. Este valor debe entregarse al *Symbulator* en henrios.

Al igual que con los capacitores, a veces nos encontramos en los libros de texto problemas en corriente alterna que muestran inductores con valores dados en Ω. Para efectos de la simulación, estos elementos deberán ser tratados como impedancias, aunque aparezcan dibujados como inductancias.

Debe tenerse la precaución de no introducir el valor de la inductancia en una unidad distinta a los henrios (H). Por ejemplo, si el valor se introduce en milihenrios (mH), las respuestas estarán equivocadas.

Así, los inductores se definen como se muestra a continuación:

l*nombre, nodo 1, nodo 2, valor en henrios*

En el análisis en corriente alterna, solamente se declaran estos cuatro términos. El análisis en dominio de la frecuencia y el análisis transitorio requieren que se declare también la corriente inicial del inductor. Cuando el inductor tenga una corriente inicial, debemos dar estos nodos en el orden adecuado que indique la dirección que deseamos darle a esta corriente inicial: desde el primer nodo hacia el segundo nodo de la descripción. El dibujo del circuito nos indicará, con una flecha, la dirección que debemos darle al inductor. Ejemplo:

l*nombre, nodo 1, nodo 2, valor, corriente inicial*

Esta corriente inicial es captado por el simulador considerándola como fluyendo a través del inductor desde el nodo 1 hacia el nodo 2.

En el análisis en corriente directa, el *Symbulator* considera a los inductores como cortocircuitos.

6.9.2 Respuestas relacionadas

En el caso de un inductor en análisis en corriente alterna, se entregan tres respuestas relacionadas: la corriente a través del inductor, la caída de voltaje en el inductor y la potencia compleja consumida por él. La corriente se almacena en una variable llamada `i`*nombre*, donde *nombre* es el del inductor. La caída de voltaje se almacena en una variable llamada `v`*nombre*, donde *nombre* es el del inductor. La potencia compleja consumida se almacena en una variable llamada `s`*nombre*, donde *nombre* es el del inductor. Son aplicables a las respuestas de este elemento todas las convenciones antes expuestas sobre dirección, polaridad y consumo.

Veamos algunos ejemplos.

Problema N° 026

Planteamiento. Encuentre la corriente *i(t)* en el circuito.

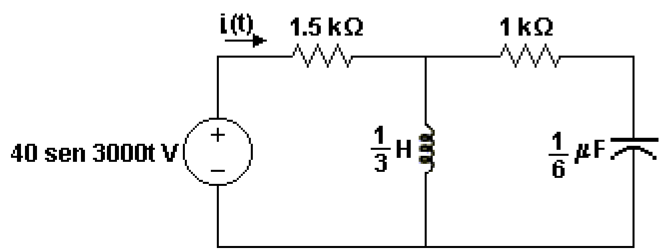

Figura 42. Circuito para el Problema N° 026.

Solución:

El valor de la fuente, *40sen(3000t),* puede ser expresado también como *40cos(3000t-90°).* Al llevar a fasor este valor, es `(40∠-90°)`. Nombramos los tres nodos superiores, de izquierda a derecha, así: `1`, `2` y `3`. El nodo inferior es el de referencia. Nombramos los elementos, damos la descripción del circuito y ejecutamos la simulación con la puerta `sq\ac`.

```
sq\ac("e1,1,0,(40∠-90°);r1,1,2,1.5E3;r2,2,3,1E3;l1,2,0,1/3;
ca,3,0,(1/6)E-6",3000)
```

Nótese que se usó `ca` como nombre de la capacitancia, pues las variable `c1` es una variable reservada. No es importante la dirección en que se definen los elementos en este problema, excepto la resistencia `r1`, que se hizo coincidir con la dirección de *i(t)*. Iniciamos la simulación con ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación, y luego aparece **`Done`**. Para ver el valor de *i(t)*, podemos solicitar el valor de `ir1`, utilizando la herramienta `absang`.

```
sq\absang(ir1)
```

`".015989779∠-126.842333°"`

Esta es la respuesta correcta. Si quisieramos expresar esta corriente en función del tiempo, sería *.015989779 cos(3000t-126.842333°).*

También la herramienta `thevenin` puede hacer uso del análisis en corriente alterna. Veamos dos ejemplos.

<u>Problema N° 027</u>

Planteamiento. Si ω=1000 rad/s, encuentre la impedancia de entrada que se mediría entre las terminales: 1) a y g; 2) b y g; 3) a y b.

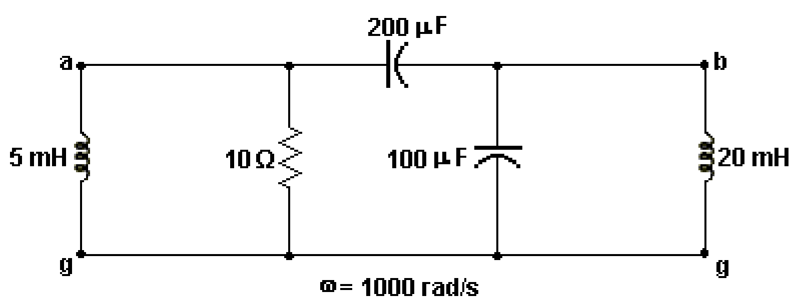

Figura 43. Circuito para el Problema N° 027.

Solución:

Tomaremos al nodo inferior como referencia, y dejaremos a los nodos *a* y *b* con los mismos nombres que les ha dado el dibujo. Nombramos los elementos, damos la descripción de la red y ejecutamos la herramienta `thevenin`.

```
sq\thevenin("l1,a,0,5E-3;r1,a,0,10;ca,a,b,200E-6;
cb,b,0,100E-6;l2,b,0,20E-3",a,0)
```

Presionamos ¸ . Escogemos `Passive` como tipo de circuito y AC como tipo de análisis. Presionamos ¸ . La herramienta `thevenin` nos pregunta ahora el valor de la frecuencia radial ω en rad/s. Presionamos **j** para desactivar el modo alfabético, introducimos `1000`, y presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Como tenemos el modo Complex Format ajustado en RECTANGULAR, no tenemos que usar el comando **ú**`Rect` para ver la respuesta en presentación rectangular. Para `zth` obtenemos **2.80898876+4.49438202*i*** ohmios, que es la respuesta correcta. Sigamos entonces con el siguiente par de nodos.

```
sq\thevenin("l1,a,0,5E-3;r1,a,0,10;ca,a,b,200E-6;
cb,b,0,100E-6;l2,b,0,20E-3",b,0)
```

Nótese que sólo cambiamos el nodo *a* por el nodo *b* en el final de la orden. Presionamos ¸ . Escogemos `Passive` y AC, y presionamos ¸ . Introducimos `1000`, y presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Para `zth` obtenemos **1.79775281-1.1235955*i*** ohmios, que es la respuesta correcta. Sigamos.

```
sq\thevenin("l1,a,0,5E-3;r1,a,0,10;ca,a,b,200E-6;
cb,b,0,100E-6;l2,b,0,20E-3",a,b)
```

Igualmente, sólo cambiamos los nodos del final. Presionamos ¸ . Escogemos `Passive` y AC, y presionamos ¸ . Introducimos `1000`, y presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta una simulación. Para `zth` obtenemos **.112359551-3.82022472*i*** ohmios, que es la respuesta correcta.

Problema N° 028

Planteamiento. Si ω=100 rad/s, encuentre: 1) Z_{ent}; 2) Z_{ent} si se conecta en cortocircuito de `x` a `y`.

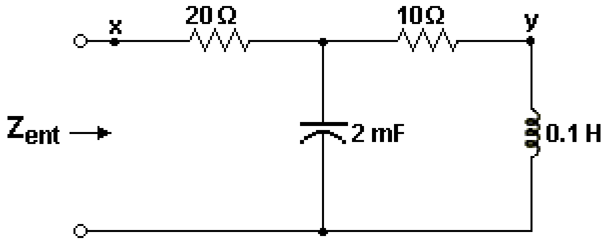

Figura 44. Circuito para el Problema N° 028.

Solución:

Tomaremos al nodo inferior como referencia, dejaremos a los nodos *x* y *y* con los mismos nombres que les ha dado el dibujo y nombraremos al nodo del medio `1`. Nombramos los elementos, damos la descripción de la red y ejecutamos la herramienta `thevenin`.

```
sq\thevenin("r1,x,1,20;ca,1,0,2E-3;r2,1,y,10;l1,y,0,.1"
,x,0)
```

Presionamos ¸ . Escogemos `Passive` y `AC`, y presionamos ¸ . Introducimos `100`, y presionamos ¸ . Para `zth` obtenemos **`22.-6.`*i*** ohmios, que es la respuesta correcta.

Para la segunda parte de este problema, el enunciado nos solicita que conectemos un cortocircuito entre *x* y *y*. El propósito es hacer, de ambos nodos, uno solo. Para efectos de la simulación, podríamos conectar un cortocircuito entre estos dos nodos. Esto agregaría un nuevo elemento a la simulación. Pero podríamos también hacer algo mejor: reemplazar por una `x` toda `y` que aparezca en la descripción. Así, simularemos como si `x` y `y` fuesen el mismo nodo, lo que para todos los efectos prácticos es equivalente a un cortocircuito, pues no nos interesa la corriente a través de este corto. No solamente estamos evitando el agregar un nuevo elemento, sino que estamos eliminando un nodo del circuito.

```
sq\thevenin("r1,x,1,20;ca,1,0,2E-3;r2,1,x,10;l1,x,0,.1"
,x,0)
```

Presionamos ¸ . Escogemos `Passive` y `AC`, y presionamos ¸ . Introducimos `100`, y presionamos ¸ . Para `zth` obtenemos **`9.6+2.8`*i*** ohmios, que es la respuesta correcta.

Hemos visto en estos dos ejemplos que la herramienta `thevenin` utiliza también el análisis en corriente alterna, con éxito. Veamos ahora un ejemplo de circuito en el cual los inductores y capacitores deben ser simulados como impedancias, pues sus valores nos son entregados en Ω.

<u>Problema N° 029</u>

Planteamiento. Encuentre I_1, I_2 e I_3.

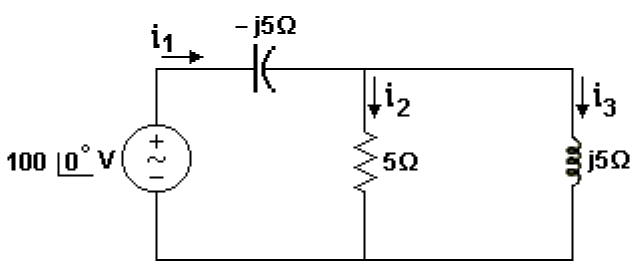

Figura 45. Circuito para el Problema N° 029.

Solución:

El valor de la fuente ya está en fasor, y puede reducirse de $(100\angle 0°)$ a simplemente `100`. El circuito muestra tres elementos pasivos: un capacitor, un resistor y un inductor. Los valores de todos ellos, sin embargo, han sido definidos en el circuito como impedancias. Por lo tanto, deben simularse como impedancias. Tendremos la precaución de hacer coincidir la numeración y la dirección de nuestras impedancias con las corrientes I_1, I_2 e I_3. Con el nodo inferior como referencia, nombramos los otros nodos y los elementos, damos la descripción del circuito y ejecutamos la simulación. Como tenemos que dar al programa una frecuencia radial como dato de entrada y no tenemos ninguna, le damos cualquier valor. Nosotros usamos *x*, por ejemplo.

```
sq\ac("e1,1,0,100;r1,1,2,-5i;r2,2,0,5;r3,2,0,5i ",x)
```

Recuérdese que el signo de negativo debe escribirse con la tecla de negativo, y no con la de resta. El operador imaginario ***i*** se escribe como lo indicamos en el punto 6.1. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación, y luego aparece **`Done`**. Para el valor de I_1, con `absang(ir1)` obtenemos **`"28.2842712∠45.°"`** amperios. Para el valor de I_2, con `absang(ir2)` obtenemos **`"20.∠90.°"`** amperios. Para el valor de I_3, con `absang(ir3)` obtenemos **`"20.∠0.°"`** amperios. Estas son las respuestas correctas.

En este problema hemos aprendido que, cuando un problema nos presente impedancias, no importa si estas fueron obtenidas de capacitores o inductores, deben ser simuladas como impedancias.

6.10 El separador **:**

En la calculadora TI, dos órdenes separadas pueden unirse en una sola línea, mediante el separador **:** (o sea dos puntos). Por ejemplo, las dos órdenes:

```
sq\ac("e1,1,0,100;r1,1,0,30i",x)
```

```
absang(ir1)
```

...pueden unirse en una sola línea, así:

```
sq\ac("e1,1,0,100;r1,1,0,30i",x):absang(ir1)
```

Este separador será utilizado en el siguiente problema.

Como vimos en el punto 6.3, cuando la frecuencia se nos entrega en Hz, hay que convertirla primero a radianes/segundo para alimentarla al simulador. El siguiente problema ilustra este punto, y también demuestra que el *Symbulator* maneja con éxito las fuentes dependientes en el análisis en corriente alterna.

<u>Problema N° 030</u>

Planteamiento. Sea $V_s=100\angle 0°$V en f=50Hz en el circuito que se muestra en la figura. Encuentre el voltaje fasorial V_x, si el elemento X es un: a) resisitor de 30Ω; b) inductor de 0.5H; c) capacitor de 50μF.

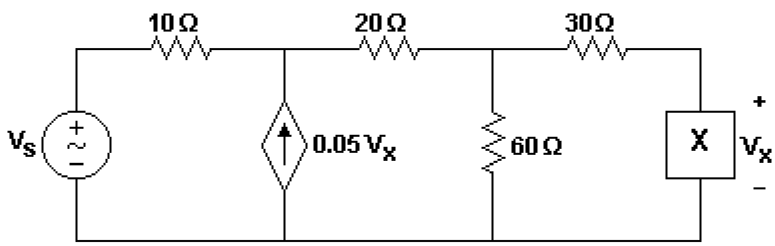

Figura 46. Circuito para el Problema N° 030.

Solución:

De ahora en adelante, redondearemos las respuestas a cuatro cifras. Realicemos la primera simulación:

```
sq\ac("e1,1,0,100;r1,1,2,10;r2,2,3,20;r3,3,4,30;j1,0,2,.05*
v4;r4,3,0,60;rx,4,0,30",2π50):sq\absang(v4)
```

Nótese que la frecuencia no se introdujo en Hz, sino que se convirtió a radianes/segundo. Nótese también el uso del separador **:** para unir ambas órdenes en una línea. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación. Al terminar la simulación, en vez de aparecer **`Done`**, aparece la respuesta a la última orden de la línea, es decir el voltaje `v4`. Este es el valor de V_x, **`28.57∠0.°`** voltios (redondeado). Esta es la respuesta correcta. Realicemos la segunda simulación, modificando el último elemento de la descripción:

```
sq\ac("e1,1,0,100;r1,1,2,10;r2,2,3,20;r3,3,4,30;j1,0,2,.05*
v4;r4,3,0,60;lx,4,0,.5",2π50):sq\absang(v4)
```

Presionamos ¸ . Para el valor de V_x, obtenemos **90.24∠25.52°** voltios, que es la respuesta correcta. Realicemos la tercera simulación, modificando el último elemento de la descripción:

```
sq\ac("e1,1,0,100;r1,1,2,10;r2,2,3,20;r3,3,4,30;j1,0,2,.05*
v4;r4,3,0,60;cx,4,0,50E-6",2π50):sq\absang(v4)
```

Presionamos ¸ . Para el valor de V_x, obtenemos **64.71∠-49.67°** voltios, que es la respuesta correcta.

Conozcamos dos nuevos elementos del *Symbulator*, relacionados con el acoplamiento magnético: la inductancia mutua y el transformador ideal.

6.11 Inductancia mutua

Una inductancia mutua es un acoplamiento magnético mutuo entre dos inductores. No es un "elemento" físico, pero sí es un elemento para los efectos de la simulación.

6.11.1 Notación

En el caso de la inductancia mutua, la mínima información necesaria para describirla por completo es la siguiente:

1. *Identificación.* La letra que identifica a las inductancias mutuas es la **m**.

2. *Nombre.* Es válido cualquier nombre que empiece con **m**.

3. *Nombre de las dos inductancias acopladas.* En el *Symbulator*, una inductancia mutua acopla dos inductancias solamente, cada una de ellas con un nombre distinto. Es necesario darle al simulador los nombres de las dos inductancias que están acopladas por la inductancia mutua.

Es importante que estas dos inductancias que están acopladas por la inductancia mutua, estén definidas en la dirección correcta. En el dibujo del circuito, cuando dos inductancias están acopladas, se utiliza un punto en un extremo de cada inductancia para indicar el sentido del acoplamiento. Así, cada inductancia tiene un extremo con punto y el otro extremo sin punto.

Al momento de definir estas dos inductancias en el Symbulator, sus direcciones deben definirse de manera consistente con los puntos. Si al definir la primera inductancia, se introdujo primero el nodo que tiene el punto en el dibujo, entonces al definir la segunda inductancia, el primer nodo que se defina también debe ser el que tiene el punto en el dibujo. Es muy importante definir los nodos de las inductancias en el orden correcto. O viceversa.

4. *Valor.* Para cada acoplamiento, existe un coeficiente de inductancia mutua.

Este coeficiente se simboliza por *M* si su valor está dado en henrios, y por *k* si su valor está dado adimensionalmente. Existe una fórmula que relaciona a estas dos presentaciones: $k=M/\sqrt{(L_1L_2)}$. Así, conociendo uno de ellos, siempre es posible conocer el otro. Para efectos de la descripción de una inductancia mutua en el *Symbulator*, el valor utilizado es *M* y debe entregarse en henrios. Por lo tanto, si el problema nos entrega el valor de *k*, debemos convertirlo a *M*, antes de utilizarlo en la descripción.

Debe tenerse la precaución de no introducir el valor de una inductancia mutua en una unidad distinta a los henrios (H). Por ejemplo, si el valor se introduce en milihenrios (mH), las respuestas estarán equivocadas.

Así, una inductancia se define como se muestra a continuación:

```
mnombre, inductancia 1, inductancia 2, valor en henrios
```

De haber necesidad, se rellena el quinto espacio con 0.

En el análisis en corriente directa, el *Symbulator* ignora las inductancias mutuas.

6.11.2 Respuestas relacionadas

No hay respuestas relacionadas a la inductancia mutua en sí misma. Sin embargo, cada una de las dos inductancias involucradas tendrá las respuestas relacionadas propias de cualquier inductancia, y que vimos en el punto 6.9.2.

Veamos algunos ejemplos.

Problema N° 031

Planteamiento. Con el circuito de la figura, presente en forma polar la corriente en el resistor de 400Ω y la relación V_2/V_1.

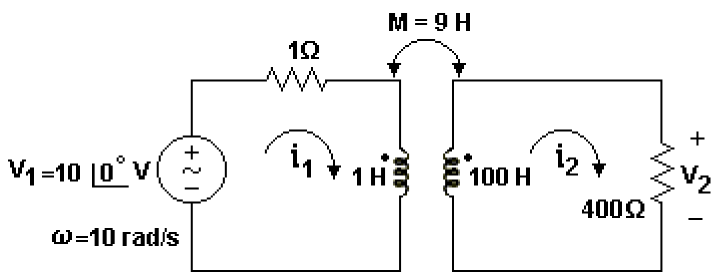

Figura 47. Circuito para el Problema N° 031.

Solución:

Nombramos los nodos de izquierda a derecha, `1`, `3` y `2`, con el propósito de que v_1 y v_2 conserven en las respuestas los mismos nombres que en el dibujo. Léase la descripción de circuito que está bajo estas líneas, y los comentarios que siguen:

```
sq\ac("e1,1,0,10;r1,1,3,1;l1,3,0,1;l2,2,0,100;r2,2,0,400;m1
,l1,l2,9",10)
```

Nótese que el primer nodo en la descripción de cada inductancia es el nodo que tiene el punto en el dibujo. Una alternativa igualmente correcta hubiese sido hacer lo inverso: que el primer nodo en la descripción de cada inductancia hubiese sido el nodo que no tiene el punto en el dibujo. Lo importante es que en ambas inductancias se utilice la misma convención. Nótese que la inductancia mutua no tiene que definirse inmediatamente a continuación de las inductancias que relaciona. Nótese también que el valor de la inductancia mutua se introdujo en henrios. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación. Al terminar la simulación, aparece **`Done.`** Para encontrar las respuestas, pedimos `absang(ir2)` y obtenemos **`.1724∠-16.7°`** amperios (redondeado), y `absang(v2/v1)` y obtenemos **`6.896∠-16.7°`** (redondeado). Estas son las respuestas correctas.

Problema N° 032

Planteamiento. En el circuito de la figura, encuentre la potencia real promedio absorbida por la fuente y los dos resistores.

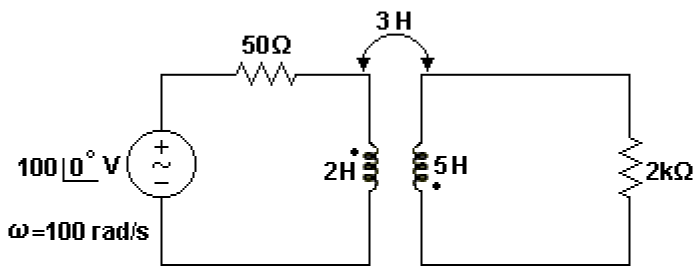

Figura 48. Circuito para el Problema N° 032.

Solución:

He aquí la descripción que usé:

```
sq\ac("e1,1,0,100;r1,1,2,50;l1,2,0,2;l2,0,3,5;m1,l1,l2,3;r2
,3,0,2–3",100)
```

Nótese que el primer nodo en la descripción de cada inductancia es el nodo que tiene el punto en el dibujo. Nótese también que el valor de la inductancia mutua se introdujo en henrios. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación. Al terminar, aparece **`Done.`** Se nos solicita la potencia real promedio absorbida. En el análisis de corriente alterna, las variables en las cuales el *Symbulator* nos da la potencia absorbida empiezan con `s`. Tal y como nos enseña la teoría, la potencia promedio es la mitad de la potencia máxima. Como se nos solicita la potencia real, usamos el comando `real` de la calculadora. Para encontrar las respuestas, pedimos `real(se1)/2` y obtenemos **`–10.4`** vatios; pedimos `real(sr1)/2` y obtenemos **`5.63`** vatios; pedimos `real(sr2)/2` y obtenemos **`4.77`** vatios. Estas son las respuestas correctas, redondeadas a tres cifras.

<u>Problema N° 033</u>

Planteamiento. Si ω=100 rad/s, encuentre la potencia real promedio: a) entregada a la carga de 10Ω, b) a la de 20Ω, y c) entregada por la fuente.

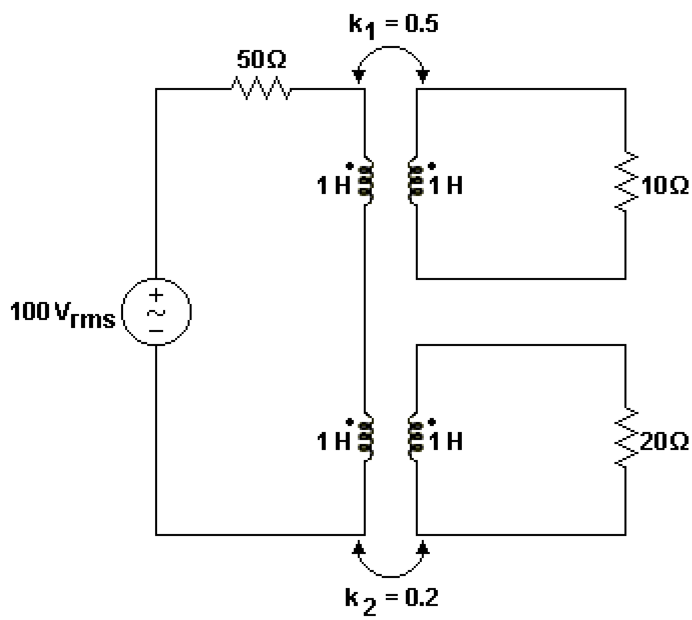

Figura 49. Circuito para el Problema N° 033.

Solución:

He aquí la descripción que usé:

```
sq\ac("e1,1,0,100√(2);l1,1,2,1;l2,2,0,1;l3,3,0,1;r1,3,0,10;
m1,l1,l3,.5√(1*1);l4,4,0,1;r2,4,0,20;m2,l2,l4,.2√(1*1)"
,100)
```

Nótese lo siguiente:

1. En un tratamiento riguroso, en el *Symbulator*, se debe introducir el valor normal de la fuente y no su valor RMS. Por ello, multiplicamos el valor RMS por raíz de dos, para convertirlo en su valor normal. Es perfectamente válido hacer esta multiplicación directamente en el campo de entrada.

2. El primer nodo en la descripción de cada inductancia es el nodo que tiene el punto en el dibujo.

3. El valor de la inductancia mutua se convirtió a henrios, directamente en el campo de entrada. Ahora bien, este problema no es el mejor para ilustrar esta conversión, porque obviamente `.2√(1*1)` es igual a `.2`. A pesar de que en este problema resulte innecesario colocar la conversión, en otros casos no será así. Por ello, coloqué la conversión para ilustrar el proceso.

4. El circuito del ejemplo está compuesto por tres pequeños circuitos. Estos tres circuitos están acoplados por inductancias mutuas. Sin embargo, cada uno de ellos tiene un nodo de referencia. Estos tres nodos de referencia deben ser el nodo `0`. Es necesario que los tres compartan el mismo nodo de referencia, aunque en el dibujo los nodos no estén conectados físicamente.

Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación. Al terminar, aparece **`Done.`** Al igual que en el problema anterior, se nos solicita la potencia real promedio absorbida. Pedimos `real(sr1)/2` y obtenemos **`.842`** vatios; pedimos `real(sr2)/2` y obtenemos **`.262`** vatios; pedimos `real(se1)/2` y obtenemos **`-1.104`** vatios. Estas son las respuestas correctas (redondeadas).

<u>Problema N° 034</u>

Planteamiento. Encuentre I_L en el circuito que se muestra.

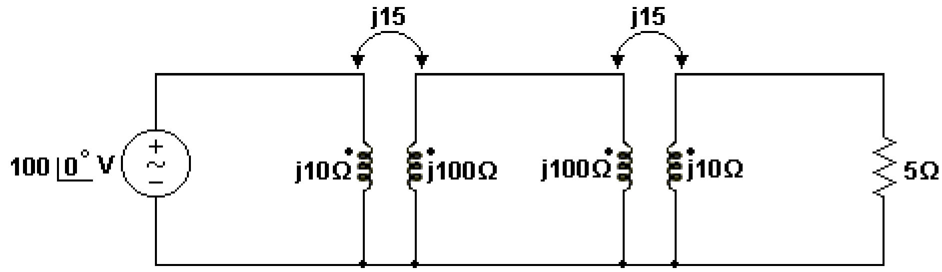

Figura 50. Circuito para el Problema N° 034.

Solución:

Este es un problema muy particular, y por ello lo guardé para el final. En él se presenta algo que el *Symbulator* no maneja por sí solo: inductancias acopladas cuyos valores están dados en Ω. Antes de resolver este problema, tenemos que llevar el circuito a una forma que el *Symbulator* pueda manejar. Tenemos dos alternativas:

La primera alternativa es simular usando impedancias. ¿Cómo? Reemplazando las inductancias mutuas por un equivalente hecho de impedancias. Aprovechamos el hecho de que los inductores acoplados están funcionando como un transformador lineal, para reemplazarlos por su equivalente T. Para un transformador lineal con valores de L_1, L_2 y M, el equivalente T se obtiene haciendo una T con elementos con valores de L_1-M a un lado, L_2-M al otro lado, y M como base de la T. En el caso de nuestra simulación, esta T se hará con impedancias, y no habrá inductores en el circuito. A continuación, la descripción que usé para resolver el problema con esta alternativa. Nótese que puede usarse cualquier frecuencia radial, y cualquier orden de los nodos de las impedancias.

```
sq\ac("e1,1,0,100;r1,1,2,-5i;r2,2,3,85i;rm1,2,0,15i;
r3,3,4,85i;r4,4,5,-5i;rm2,4,0,15i;rL,5,0,5",x)
```

La segunda alternativa es simular usando inductores. ¿Cómo? Dado que el problema no nos ha especificado un valor para la frecuencia radial, inventamos un valor cualquiera. Yo usé un valor unitario: 1 rad/seg. Luego, convierto todos los valores de las inductancias e inductancias mutuas de Ω a H, usando la frecuencia radial. Si la frecuencia es unitaria, el valor en ohmios será el mismo valor en henrios (sin el operador imaginario, obviamente). A continuación, la descripción que se debe usar para resolver el problema con esta alternativa. Nótese que el orden de los nodos de las inductancias se escogió cuidadosamente, para cumplir la convención de los puntos.

```
sq\ac("e1,1,0,100;l1,1,0,10;l2,2,0,100;m1,l1,l2,15;l3,2,0,1
00;l4,3,0,10;m2,l3,l4,15;rL,3,0,5",1)
```

En ambos casos, llamé `rL` a la resistencia de 5Ω. Escoja el lector cuál de las alternativas prefiere. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación. Al terminar, aparece **`Done.`** Pedimos `absang(irL)` y obtenemos **`1.26∠-60.2°`** amperios (redondeado), que es la respuesta correcta.

En los ejemplos anteriores, hemos visto cómo simular en el *Symbulator* las inductancias mutuas, en diferentes tipos de problemas. Aprendamos ahora sobre un nuevo elemento.

6.12 Transformador ideal

Un transformador ideal es un modelo idealizado de acoplamiento unitario de un transformador físico. Se supone que no tiene pérdidas.

6.12.1 Notación

En el caso del transformador ideal, la mínima información necesaria para describirlo por completo es la siguiente:

1. *Identificación.* La letra que identifica a un transformador ideal es la **t**.

2. *Nombre.* Se le debe dar un nombre cualquiera, que empiece con **t**.

3. *Nodos.* Un transformador ideal tiene cuatro puntos de conexión, dos superiores y dos inferiores. Para una simulación en el *Symbulator*, los dos inferiores se asumen siempre conectados al nodo de referencia. Los superiores deben estar cada uno conectado a un nodo distinto. Es necesario darle al

simulador los nombres de estos dos nodos superiores a los cuales está conectado el transformador.

4. *Relación de vueltas.* Un transformador ideal tiene una relación de vueltas de un lado con respecto al otro. Este valor debe entregarse al *Symbulator,* en la forma de dos números, uno para cada lado.

Cuando un transformador tenga los puntos de ambos lados en los nodos superiores, o los dos puntos en los extremos inferiores, los dos números de la relación de vueltas deben ser del mismo signo, por ejemplo ambos positivos. Por el contrario, cuando el transformador tenga el punto de un lado en el nodo superior pero el punto del otro lado en el nodo inferior, los dos números de la relación de vueltas deben tener distinto signo, por ejemplo uno positivo y uno negativo.

Así, los transformadores ideales se definen como se muestra a continuación:

t*nombre, nodo 1, nodo 2, vueltas 1, vueltas 2*

La descripción del transformador ideal siempre tiene 5 términos, por lo que nunca habrá necesidad de agregar un cero al final.

En el análisis en corriente directa, el *Symbulator* ignora los transformadores ideales.

6.12.2 Respuestas relacionadas

En el caso de un transformador ideal, se entregan dos respuestas relacionadas: la corriente que entra al transformador por el primer nodo, y la corriente que entra al transformador por el segundo nodo. Recuérdese que estos dos nodos son los superiores, pues el inferior en ambos lados es el nodo de referencia. Estas corrientes se almacenan en variables llamadas `i`*nombreNODO,* donde *nombre* es el nombre del transformador ideal, y *NODO* es el nombre del nodo. Así, para un transformador ideal que se llame `t1`, y que esté conectada a los nodos `1` y `2`, las respuestas relacionadas serán `it11` e `it12`. Nótese que ambas corrientes se consideran entrando al transformador por los nodos superiores.

Por lo tanto, si una de ellas vale, por ejemplo, `-5` amperios, significa que están entrando `-5` amperios, o sea que están saliendo `5` amperios.

Es importante notar que si alguno de los extremos del transformador ideal se encuentra en circuito abierto, ambas corrientes resultarán nulas.

Veamos algunos ejemplos.

Problema N° 035

Planteamiento. Para el circuito que se muestra, encuentre el valor RMS de I_1, I_2, I_3, y las potencias reales promedio consumidas en las tres resistencias.

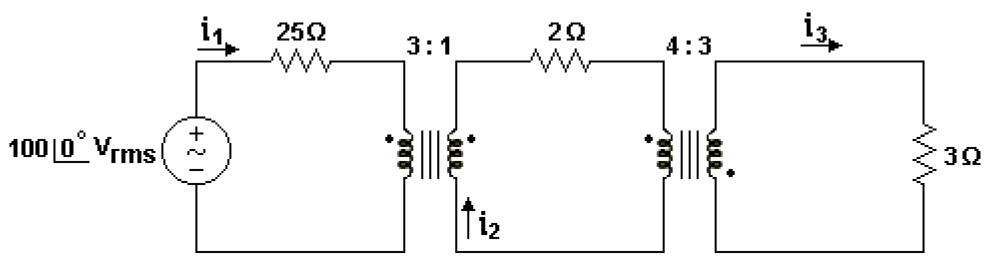

Figura 51. Circuito para el Problema N° 035.

Solución:

A continuación, la descripción que usé para resolver el problema.

```
sq\ac("e1,1,0,100√(2),0;r1,1,2,25,0;t1,2,3,3,1;r2,3,4,2,0;t2
,4,5,4,-3;r3,5,0,3,0",x)
```

Nótese lo siguiente:

1) El valor de la fuente se convirtió de valor RMS a valor normal.

2) Puede usarse cualquier frecuencia radial.

3) Rellené con ceros las descripciones de aquellos elementos cuyas descripciones no tienen 5 términos. Si nos olvidamos de hacer esto, obtendremos un *error de sintaxis*.

4) En el caso del transformador `t2`, como los puntos están uno arriba y otro abajo, uno de los dos números de la relación de vueltas en la descripción se colocó negativo.

Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación. Al terminar, aparece **`Done.`** Para obtener el valor RMS de I_1, tenemos tres alternativas: pedir ya sea `absang(-ie1/√(2))`, `absang(ir1/√(2))` o `absang(it12/√(2))`, y obtenemos **`1.099∠0.°`** amperios. Para obtener el valor RMS de I_2, también tenemos tres alternativas: pedir ya sea `absang(-it13/√(2))`, `absang(ir2/√(2))` o `absang(it24/√(2))`, y obtenemos **`1.30∠0.°`** amperios. Para el valor RMS de I_3, tenemos dos alternativas: pedir ya sea `absang(-it25/√(2))` o `absang(ir3/√(2))`, y obtenemos **`4.40∠180.°`** amperios.

Recuérdese que el *Symulator* siempre definirá las dos corrientes de un transformador ideal entrando al transformador por los dos nodos superiores, y el nombre de cada una es `i` seguido del nombre del transformador y del nombre del nodo por el cual entra la corriente.

Se nos solicita la potencia real promedio absorbida por las tres resistencias. Con `real(sr1)/2.` obtenemos **`30.2`** vatios; con `real(sr2)/2.` obtenemos **`21.7`** vatios; y con `real(sr3)/2.` obtenemos **`58.0`** vatios. Estas son las respuestas correctas (redondeadas). Usamos `/2.` en vez de `/2` porque deseábamos obtener respuestas en formato aproximado.

Veamos dos problemas similares, pero antes aprendamos algunos conceptos nuevos sobre los valores RMS.

6.13 Problemas con valores RMS

En dos problemas que hemos visto en este Capítulo, el valor de la fuente independiente se nos entregó como valor RMS. Como respuestas, nos pidieron las corrientes RMS y las potencias promedio. Son muy frecuentes en análisis de corriente alterna este tipo de problemas que usan valores de voltaje y corriente en RMS, y potencias promedio. Estos problemas pueden recibir dos tratamientos en el *Symbulator*: el riguroso y el informal.

6.13.1 Tratamiento riguroso

Los dos problemas que hemos visto anteriormente han sido resueltos con el tratamiento riguroso. Este tratamiento asume que los valores de las fuentes independientes se introducen en forma normal, y los voltajes, corrientes y potencias de las respuestas también están en forma normal. Por ello:

1) Cuando el problema nos entregue el valor de alguna fuente en RMS, debemos convertirlo a valor normal al introducirlo en la descripción.

2) Si deseamos los valores en RMS de las corrientes y voltajes que fueron entregados como respuestas, debemos dividir el valor normal entre raíz de dos para tenerlo en RMS.

3) Si deseamos los valores promedios de las potencias que fueron entregadas como respuestas, debemos dividir el valor normal entre dos para tenerlo en promedio.

6.13.2 Tratamiento informal

Los dos problemas que veremos a continuación serán resueltos con el tratamiento informal. Este tratamiento se fundamenta en el hecho de que, si los valores de todas las fuentes independientes se introducen en RMS, entonces todos los voltajes y corrientes de las respuestas estarán en RMS, y todas las potencias de las respuestas estarán en valor promedio. Por ello:

1) Cuando el problema nos entregue el valor de alguna fuente en RMS, debemos introducirlo tal como está. Si, por el contrario, nos lo entrega en valor normal, debemos convertirlo a RMS antes de introducirlo.

2) Recibiremos las corrientes y voltajes de las respuestas en valores RMS.

3) Recibiremos las potencias de las respuestas en valores promedio.

Veamos ahora dos problemas, resueltos con el *tratamiento informal*.

<u>Problema N° 036</u>

Planteamiento. Para el circuito que se muestra, encuentre las potencias reales promedio consumidas en las resistencias.

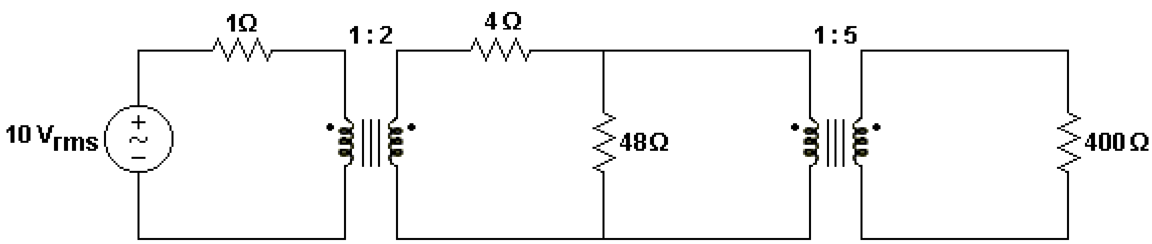

Figura 52. Circuito para el Problema N° 036.

Solución:

A continuación, la descripción que usé para resolver el problema, con el *tratamiento informal*.

```
sq\ac("e1,1,0,10,0;r1,1,2,1,0;t1,2,3,1,2;r2,3,4,4,0;r3,4,0,
48,0;t2,4,5,1,5;r4,5,0,400,0",x)
```

Nótese que el valor de la fuente se dejó en RMS. Presionamos ¸ . Vemos en la pantalla las frases que nos indican que se ejecuta la simulación. Al terminar, aparece **`Done.`** Podríamos obtener el valor RMS de cualquier corriente instantáneamente. Sin embargo, el problema no nos solicita ninguna corriente. Se nos solicita la potencia real promedio absorbida por las tres resistencias. Como estamos usando el método informal, las potencias ya están en promedio. Con `real(sr1)` obtenemos **4** vatios; con `real(sr2)` obtenemos **4** vatios; con `real(sr3)` obtenemos **3** vatios; y con `real(sr4)` obtenemos **9** vatios. Nótese que no hubo necesidad de usar `/2` porque las respuestas ya estaban en promedio.

Problema N° 037

Planteamiento. Para el circuito que se muestra, encuentre las potencias reales promedio consumidas por las resistencias de 10Ω. Repita conectando A y C, y B y D.

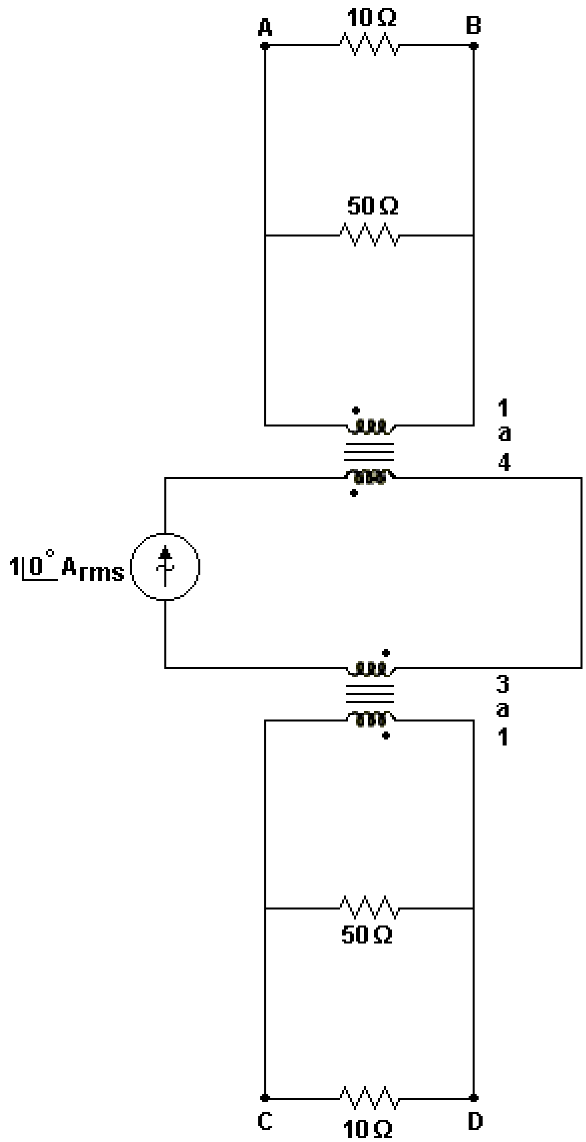

Figura 53. Circuito para el Problema N° 037.

Solución:

Por la disposición vertical del dibujo de este circuito, y por la posición de la fuente en él, podría resultar algo complicado el escoger el nodo de referencia para cada sección del circuito. Es necesario usar los nodos de la derecha del circuito (entre ellos B y D) como referencia en cada uno de los tres circuitos, porque los dos transformadores deben tener cada uno los dos extremos inferiores conectados en el nodo de referencia. Los nodos de la izquierda, de abajo hacia arriba, los nombré C=1, 2, 3 y A=4. A continuación, la descripción que usé para resolver la primera parte del problema, con el *tratamiento informal*.

```
sq\ac("r1,1,0,10,0;r2,1,0,50,0;t1,1,2,1,3;j1,2,3,1,0;t2,3,4
,4,1;r3,4,0,50,0;r4,4,0,10,0",x)
```

El valor de la fuente se dejó en RMS. Presionamos ¸ . Al terminar la simulación, aparece **`Done`**. Las potencias ya están en promedio. Con `real(sr1)`, ¥+ ¸ obtenemos **`62.5`** vatios; y con `real(sr4)`, ¥+ ¸ obtenemos **`111.1`** vatios. A continuación, la descripción que usé para resolver la segunda parte del problema, con el *tratamiento informal*.

```
sq\ac("r1,1,0,10,0;r2,1,0,50,0;t1,1,2,1,3;j1,2,3,1,0;t2,3,4
,4,1;r3,4,0,50,0;r4,4,0,10,0",x)
```

Nótese que no utilicé un cortocircuito para conectar A con C, sino que reemplacé el nodo 4 por el nodo 1 en la descripción. No hay necesidad de conectar B con D, porque ambos son el nodo de referencia. Presionamos ¸ . Al terminar la simulación, aparece **`Done`**. Con `real(sr1)`, ¥+ ¸ obtenemos **`1.736`** vatios; y con `real(sr4)`, ¥+ ¸ obtenemos el mismo valor de `sr1`.

Hemos visto en estos dos problemas la manera en que se pueden utilizar, con el *tratamiento informal*, los valores de las fuentes en RMS para obtener directamente respuestas en RMS, y las potencias en promedio.

Veamos un último ejemplo, de la herramienta `thevenin` aplicada a un circuito con transformadores ideales.

<u>Problema N° 038</u>

Planteamiento. Encuentre el equivalente de Thévenin en las terminales *a* y *b* para la red mostrada en la figura.

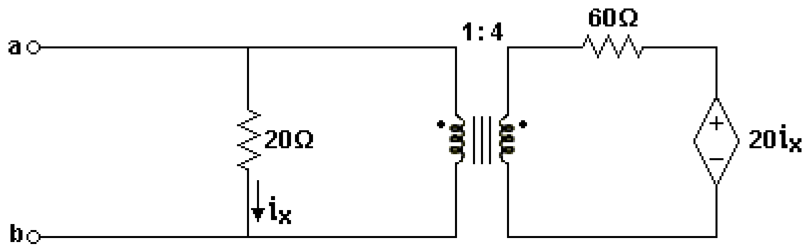

Figura 54. Circuito para el Problema N° 038.

Solución:

A continuación, la descripción:

```
sq\thevenin("rx,a,0,20,0;t1,a,1,1,4;r1,1,2,60,0;e1,2,0,20*i
rx,0",a,0)
```

Presionamos ¸ . Escogemos `Passive` y `AC`, y presionamos ¸ . Introducimos `x`, o cualquier otro valor como frecuencia radial y presionamos ¸ . Para `zth` obtenemos **`4`** ohmios, que es el valor correcto.

Con este ejemplo, hemos terminado el capítulo de simulación numérica en corriente alterna.

Capítulo
7
Bipuertos

7.1 Introducción

Un bipuerto es una red que tiene dos pares de terminales. En el *Symbulator*, se pueden simular los bipuertos como elementos de circuito, asumiendo que:

- la red está compuesta sólo de elementos lineales,
- la red no contiene fuentes independientes (puede contener fuentes dependientes),
- los dos nodos superiores de las terminales están conectados a nodos distintos del circuito
- los dos nodos inferiores de las terminales se consideran conectadas al nodo de referencia, `0`

El *Symbulator* acepta bipuertos con parámetros de 6 tipos diferentes:

- Parámetros de admitancia
- Parámetros de impedancia
- Parámetros híbridos
- Parámetros de ganancia
- Parámetros de transmisión
- Parámetros de transmisión inversa

Hay tres procedimientos que se pueden realizar en el *Symbulator*, relacionados con los bipuertos:

- **Simular** circuitos que contienen bipuertos. Para ello, se usan los bipuertos como elementos de circuito.
- Encontrar las **ganancias** en un bipuerto. Para ello, se usa la herramienta `gain` tras una simulación de un circuito.
- Encontrar el bipuerto **equivalente** de una red. Para ello, se usa la herramienta `port` y la descripción de una red.

En este capítulo aprenderemos todo lo relacionado con los bipuertos en el *Symbulator*.

7.2 Notación

En el caso de los bipuertos, la mínima información necesaria para describirlos por completo es la siguiente:

1. *Identificación.* Las letras que identifican a los diferentes bipuertos son:
    - Parámetros de impedancia, la letra `z`
    - Parámetros de admitancia, la letra `y`
    - Parámetros híbridos, la letra `h`
    - Parámetros de ganancia, la letra `g`
    - Parámetros de transmisión, la letra `a`
    - Parámetros de transmisión inversa, la letra `b`
2. *Nombre.* Es válido cualquier nombre que empiece con la letra que identifica al bipuerto que se quiere describir, y que no sea una variable reservada, como aquellas listadas en el Manual de Usuario y el punto 1.8.5 de este texto. Ejemplo: `zc` y `yc` son variables reservadas que no pueden ser usadas como nombres.
3. *Nodos.* Un bipuerto tiene cuatro puntos de conexión, dos superiores y dos inferiores. Para una simulación en el *Symbulator*, los dos inferiores se asumen

siempre conectados al nodo de referencia. Los superiores deben estar conectados a nodos distintos entre sí. Sólo es necesario darle al simulador los nombres de estos dos nodos superiores a los cuales está conectado el bipuerto.

Así, los bipuertos se definen como se muestra a continuación:

`letra``*nombre, nodo 1, nodo 2*`

Según sea necesario, se agregan uno o dos ceros al final.

4. *Valor de los parámetros.* Los bipuertos son elementos únicos, porque requieren algo que ningún otro elemento requiere en el *Symbulator*: variables almacenadas previas a la simulación. Cada bipuerto que se simule como parte de un circuito, requiere que los valores de sus 4 parámetros estén almacenados en 4 variables, una para cada parámetro, de la siguiente forma:
   - Parámetro `11`, en la variable ***letra****nombre*`11`
   - Parámetro `12`, en la variable ***letra****nombre*`12`
   - Parámetro `21`, en la variable ***letra****nombre*`21`
   - Parámetro `22`, en la variable ***letra****nombre*`22`

Debe tenerse la precaución de introducir estos valor en unidades enteras, y no en sus múltiplos. Esto significa que no se deben introducir en milis (m), micros (μ), kilos (k), megas (M), etc.

7.3 Respuestas relacionadas

En el caso de un bipuerto, se entregan dos respuestas relacionadas. Se trata de dos corrientes: la corriente que entra al bipuerto por el primer nodo, y la corriente que entra por el segundo nodo. Recuérdese que estos dos nodos son los superiores, pues el inferior en ambos lados es el nodo de referencia. Estas corrientes se almacenan en variables llamadas `i`***letra****nombreNODO*, donde ***letra****nombre* es el nombre del bipuerto, y *NODO* es el nombre del nodo. Así, para un bipuerto que se llame `yp`, y que esté conectada a los nodos `1` y `2`, las respuestas relacionadas serán `iyp1` e `iyp2`. Recuérdese que ambas corrientes se consideran entrando al bipuerto por los nodos superiores.

Veamos nuestro primer problema de bipuertos.

Problema N° 039

Planteamiento. En el siguiente circuito, los parámetros del bipuerto mostrado son zp11=20/3, zp12=1/3, zp21=2500/3, zp22=200/3. Encuentre: a) el voltaje en el nodo de carga y las corrientes que entran al bipuerto por ambos extremos, b) la impedancia de entrada y las ganancias de corriente, voltaje y potencia del bipuerto, c) la impedancia de salida de la red, vista por la carga, y d) los equivalentes de este bipuerto en los otros cinco tipos de parámetros.

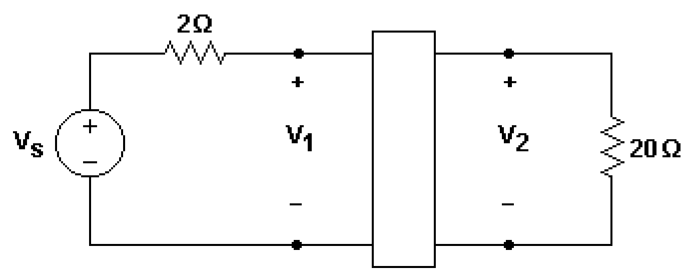

Figura 55. Circuito para el Problema N° 039.

Solución:

Este problema, especialmente diseñado para inaugurar nuestra experiencia con los bipuertos, será resuelto en cuatro partes.

La pregunta **a)** será respondida simulando el circuito usando un análisis de corriente directa. Llamaremos al bipuerto `zp`. Primero, definimos las variables que describen los parámetros del bipuerto.

```
20/3!zp11:1/3!zp12:2500/3!zp21:200/3!zp22
```

Tras insertar esta línea y presionar ¸ , veremos en el área de historia el valor del último de estos parámetros como respuesta a nuestro comando. Lo que a nosotros nos interesa es que se almacenaron los valores en las variables. Nótese que usamos el operador `:` y el comando `!`. Una vez que están almacenadas las variables de los parámetros del bipuerto, procedemos a ordenar la simulación. A continuación, la descripción que utilizamos:

```
sq\dc("e1,1,0,1;r1,1,2,2;r2,3,0,20;zp,2,3,0")
```

Nótese que, como la descripción del bipuerto sólo require tres términos, rellenamos con un cero para mantener la simetría. Presionamos ¸ . Tras las frases, aparece **`Done`**. El problema nos ha preguntado el voltaje en el nodo de carga y las corrientes que entran al bipuerto por ambos extremos. Obtenemos estos valores así: el voltaje en el nodo de carga lo obtenemos con `v3`, y vale **`2500/71`** voltios; la corriente que entra al bipuerto por la izquierda la obtenemos con `izp2`, y vale **`13/71`** amperios; y la corriente que entra al bipuerto por la derecha la obtenemos con `izp3`, y vale **`-125/71`** amperios. El signo negativo de esta última corriente nos indica que la corriente realmente está saliendo del bipuerto por la derecha, y no entrando.

Hemos aprendido que un bipuerto puede simularse fácilmente en el *Symbulator* como un elemento de circuito más. Para responder a la pregunta **b)**, usaremos una nueva herramienta.

7.4 La herramienta `gain`

Esta herramienta nos ayuda a encontrar las ganancias de corriente, voltaje y potencia, y la impedancia de entrada de una red. Esta red puede contener o no bipuertos. Aprendamos a usar esta herramienta al mismo tiempo que respondemos la segunda pregunta. La herramienta `gain` se ejecuta así:

```
sq\gain()
```

No toma argumentos de entrada entre los paréntesis. Sin embargo, al ejecutarla, se abre un formulario en el cual debemos introducir cuatro valores de entrada.

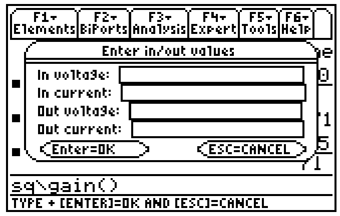

Figura 56. Formulario de entrada de la herramienta `gain`*, en la TI-89.*

Los cuatro valores solicitados son:

- El voltaje en el nodo de la terminal de entrada. En el caso de nuestro problema, este voltaje es `v2`, pues éste es el voltaje en el nodo de entrada al bipuerto, pues es el que está hacia la fuente.
- La corriente que entra por la terminal de entrada. En el caso de nuestro problema, esta corriente es `izp2`, pues ésta es la corriente que entra al bipuerto en el lado de entrada, o sea el lado que está hacia la fuente.
- El voltaje en el nodo de la terminal de salida. En el caso de nuestro problema, este voltaje es `v3`, pues éste es el voltaje en el nodo de salida del bipuerto, pues es el que está hacia la carga.
- La corriente que entra por la terminal de salida. En el caso de nuestro problema, esta corriente es `izp3`, pues ésta es la corriente que entra al bipuerto en el lado de salida, o sea el lado que está hacia la carga. Nótese que el decir corriente de salida no significa corriente que está saliendo. La corriente se debe definir como entrando al bipuerto, pero en el lado de salida.

El formulario, una vez lleno, debe lucir así:

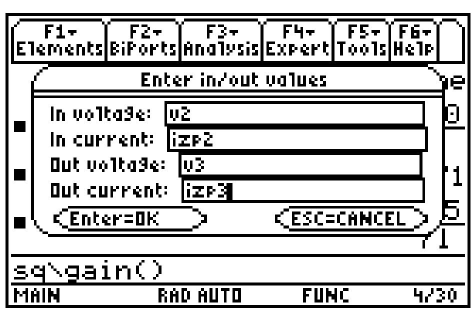

Figura 57. El formulario lleno, en la TI-89.

Presionamos ¸, dos veces si es necesario. A continuación aparece una pantalla, que funciona de forma semejante a la de la herramienta `thevenin`, en la cual debemos presionar ¸ tantas veces como valores buscamos.

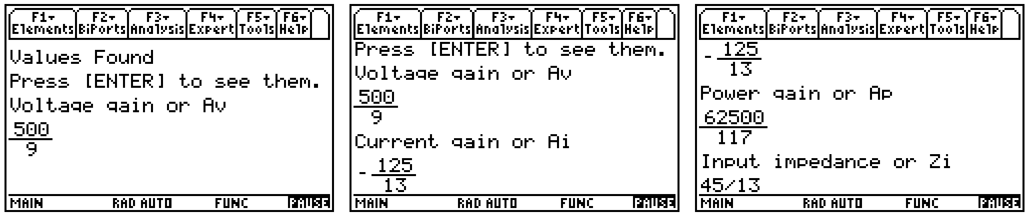

Figura 58. Pantalla de respuestas de `gain`*, en la TI-89, tras presionar* ¸ *una, dos y cuatro veces.*

Vemos en la pantalla que la ganancia de voltaje, llamada Av o Gv dependiendo del libro de texto, vale **`500/9`**; la ganancia de corriente, llamada Ai o Gi, vale **`–125/13`**; la ganancia de potencia, llamada Ap o Gp, vale **`62500/117`**; y la impedancia de entrada, llamada Zi o Ri, vale **`45/13`**. Además de mostrar estos valores en pantalla, la herramienta también los almacena en variables con los nombres `av`, `ai`, `ap`, y `zi`, respectivamente. Todos estos valores, por estar almacenados en la memoria, pueden solicitarse luego, incluso en forma de valores aproximados si se desea.

Hemos aprendido el uso de la herramienta `gain`. Respondamos ahora la siguiente pregunta.

7.5 Impedancia de salida

La pregunta **c)** requiere de algunos conocimientos teóricos. La teoría nos enseña que la impedancia de salida de la red, vista por la carga, es la impedancia equivalente de la red, desde las terminales en donde está conectada la carga, cuando se ha eliminando la carga y se han *matado* las fuentes independientes. *Matar* una fuente independiente de voltaje es convertirla en un cortocircuito. *Matar* una fuente independiente de corriente es convertirla en un circuito abierto. Ya definimos los parámetros de este bipuerto cuando respondimos a la primera pregunta, y estos se encuentran en la memoria. Por ello, no hay necesidad de volver a definirlos. A continuación, la descripción que utilizamos:

```
sq\thevenin("e1,1,0,0;r1,1,2,2;zp,2,3,0",3,0)
```

Nótese que la carga ha sido removida de la descripción, y que la fuente se ha *matado*, es decir que su valor se ha hecho nulo. Esto es lo mismo que haberla convertido en cortocircuito. Presionamos ¸ , escogemos `Passive` y `DC`, y presionamos ¸ . En la pantalla de respuestas, aparece que **`zth`** es **`450/13`** ohmios, que es la impedancia de salida correcta.

Hemos aprendido cómo encontrar la impedancia de salida de una red. También hemos visto que la herramienta `thevenin` trabaja también con bipuertos. Respondamos ahora la última pregunta, usando una nueva herramienta.

7.6 La herramienta **`port`**

Esta herramienta nos entrega los parámetros del bipuerto equivalente de cualquier red con dos terminales. Aprendamos a usar esta herramienta, al mismo tiempo que respondemos la cuarta pregunta. La herramienta `port` se ejecuta así:

```
sq\port(descripción de la red,nodo1,nodo2)
```

Se entregan tres argumentos de entrada entre los paréntesis:

- La descripción de la red de dos terminales, cuyo equivalente en bipuerto se desea encontrar. Esta red debe ser un circuito compuesto sólo de elementos pasivos, y no puede contener fuentes independientes. Esta red puede estar compuesta, digamos, de resistencias solamente. También puede estar compuesta de uno o más bipuertos. O de una mezcla de elementos resistivos y bipuertos. Como esta red puede contener otros bipuertos, la herramienta `port` es un poderoso aliado para encontrar el equivalente de un bipuerto dado en otros parámetros distintos, y para reducir un conjunto de varios bipuertos (en serie, en paralelo o en combinación) a un único bipuerto equivalente.

- El nodo superior de la terminal de **entrada** de la red.

- El nodo superior de la terminal de **salida** de la red.

Se asume que los nodos inferiores de las dos terminales es el nodo de referencia.

Una vez que hemos escrito la línea mostrada, presionamos ¸ . Al ejecutarla, se abre un formulario en el cual debemos hacer algunas selecciones.

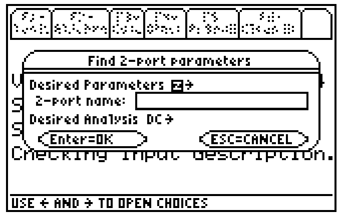

Figura 59. Formulario de la herramienta `port`*, en la TI-89.*

Las tres selecciones son:

- El tipo de parámetro que queremos encontrar. Las opciones son `z`, `y`, `h`, `g`, `a` y `b`, cuyos significados ya listamos en el punto 7.2.

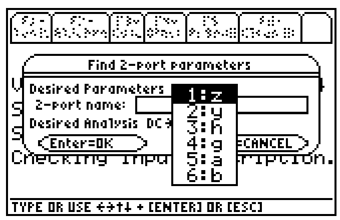

Figura 60. Selección del tipo de parámetro, en la TI-89.

- El nombre que deseamos darle al bipuerto, sin incluir la letra que identifica el tipo de bipuerto. Por ejemplo, si queremos que el bipuerto se llame `zp1`, el nombre que debemos introducir aquí es sólo `p1`, pues la `z` ya la hemos escogido arriba. Podemos usar cualquier nombre, excepto aquellos que, al combinarse con la letra del bipuerto que hallamos escogido, resulte ser el nombre de una variable reservada. Por ejemplo, si escogemos un bipuerto tipo `z` o `y`, no podemos usar un nombre como `c`, porque el nombre completo del bipuerto sería `zc` o `yc`, que son ambas variables reservadas, como lo señalamos en el punto 1.8.5. Existe un gran número de variables reservadas que empiezan con `z` y con `y`. Se puede encontrar un listado completo de ellas en el Manual de Usuario. Como recomendación general, debe usarse una letra o un número como nombre, y que esta letra no sea la letra c.

Figura 61. Introducción del nombre del bipuerto, en la TI-89.

- El tipo de análisis que deseamos. Podemos escoger entre análisis en corriente directa DC, en corriente alterna AC y en dominio de la frecuencia FD.

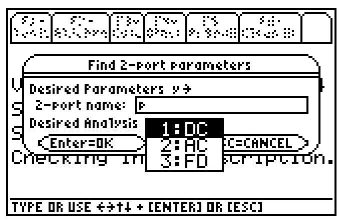

Figura 62. Selección del tipo de análisis, en la TI-89.

A continuación, se ejecuta una simulación de la red. Luego aparece una pantalla de respuestas que funciona de forma semejante a las anteriores. Debemos presionar ¸ cuatro veces para ver los cuatro parámetros buscados presentados en la pantalla.

Estos parámetros quedarán también almacenados en la memoria de la calculadora, en variables con los nombres que corresponden al nombre del bipuerto, según la convención de nombres que expusimos en el punto 7.2.

Con este conocimiento, veamos ahora cómo podemos responder la pregunta **d)** utilizando la herramienta `port`. La pregunta nos solicita encontrar los equivalentes del bipuerto en todos los demás tipos de parámetros. A continuación, la solución a esta pregunta.

Encontremos primero el equivalente en parámetros de admitancia del bipuerto dado. Introdúzcase la siguiente línea de comando en el *área de entrada*.

```
sq\port("zp,1,2,0",1,2)
```

Esta línea ordena utilizar la herramienta `port` para encontrar el equivalente de la red descrita, que es el bipuerto zp, cuyos parámetros de impedancia ya son conocidos y se llaman zp11, zp12, zp21 y zp22. Estos parámetros ya se encuentran almacenados en la memoria, desde que respondimos la primera pregunta. Si el usuario tiene alguna duda sobre si estos parámetros están o no almacenados en la memoria, puede usar el entorno Var-Link para verificar. Si estos parámetros fueron borrados por el usuario, se hace necesario introducirlos nuevamente, como se mostró en la pregunta anterior. El hecho es que los parámetros de un bipuerto deben estar almacenados antes de ejecutar cualquier simulación que involucre a este bipuerto.

El bipuerto se encuentra conectado entre los nodos 1 y 2, y esto se indica en la línea del comando. Aunque parezca innecesario en este caso, los dos nodos extremos de la red que quiere reducirse deben especificarse, pues no siempre esta red estará compuesta por un elemento únicamente, como veremos más adelante.

Así, presionamos ¸ . Especificamos `y` como tipo de bipuerto deseado, `p` como nombre y `DC` como análisis. A continuación mostramos las respuestas.

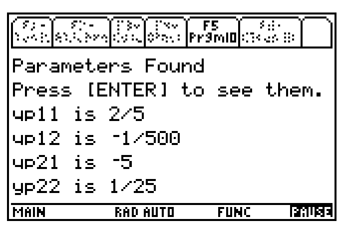

Figura 63. Parámetros Y encontrados, en la TI-89.

Este es el equivalente correcto. Recuérdese que estos parámetros de la respuesta, además de aparecer en pantalla, también se almacenan en variables con los nombres correspondientes. Encontremos ahora otro equivalente del bipuerto en otro tipo de parámetros.

Para encontrar el equivalente en parámetros híbridos del bipuerto dado, el procedimiento es idéntico. Sólo por propósitos didácticos, usaremos ahora como bipuerto de partida el recién encontrado bipuerto `yp` cuyos parámetros están almacenados ya en la memoria, en lugar del bipuerto `zp` que utilizamos en la línea anterior. Introdúzcase la siguiente línea de comando en el *área de entrada*.

```
sq\port("yp,1,2,0",1,2)
```

Presionamos ¸ . Especificamos `h` como tipo de bipuerto deseado, `p` como nombre y `DC` como análisis. A continuación mostramos las respuestas.

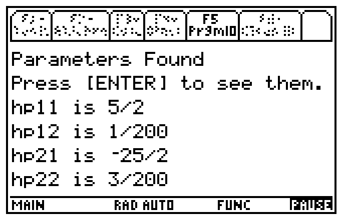

Figura 64. Parámetros H encontrados, en la TI-89.

Para encontrar el equivalente en parámetros de ganancia del bipuerto dado, el procedimiento es igual. Usaremos como bipuerto de partida el bipuerto `hp`, cuyos parámetros están en la memoria. Introdúzcase la siguiente línea de comando en el *área de entrada*.

```
sq\port("hp,1,2,0",1,2)
```

Presionamos ¸ . Especificamos `g` como tipo de bipuerto deseado, `p` como nombre y `DC` como análisis. A continuación mostramos las respuestas.

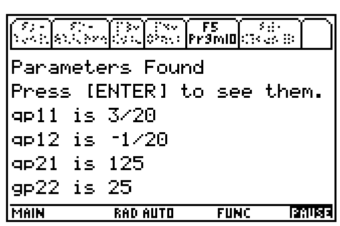

Figura 65. Parámetros G encontrados, en la TI-89.

Encontremos el equivalente en parámetros de transmisión del bipuerto, usando como bipuerto de partida el bipuerto `gp`. Introdúzcase la siguiente línea de comando en el *área de entrada*.

```
sq\port("gp,1,2,0",1,2)
```

Presionamos ¸ . Especificamos `a` como tipo de bipuerto deseado, `p` como nombre y `DC` como análisis. A continuación mostramos las respuestas.

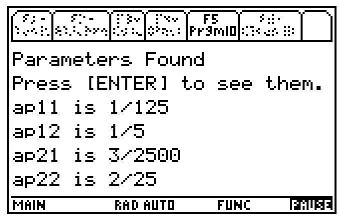

Figura 66. Parámetros A encontrados, en la TI-89.

Encontremos el equivalente en parámetros de transmisión inversa del bipuerto, usando como bipuerto de partida el bipuerto `ap`. Aquí encontramos una variante en el procedimiento. Cuando resolvimos la pregunta **a)**, hace un momento, la herramienta `gain` generó una variable llamada `Ap`, la cual contenía la ganancia de potencia. Esta variable todavía está en la memoria. Si nos olvidamos de esto, y tratamos de ejecutar la herramienta usando el `ap` como nombre del bipuerto, recibiremos un error del *Symbulator* indicándonos que el nombre que le hemos asignado al elemento #1 de nuestra descripción de circuito no es apropiado, y que por favor corrijamos esta descripción. El nombre es inapropiado porque es el nombre de una variable en uso. El mensaje se muestra a continuación:

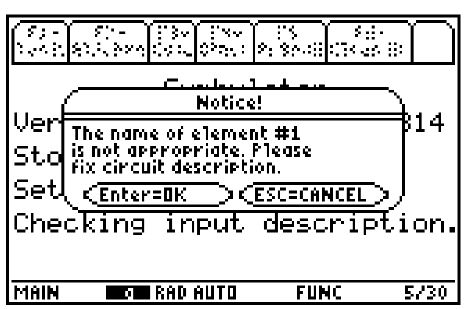

Figura 67. Mensaje de error por nombre de elemento inapropiado, en la TI-89.

Por ello, debemos borrar la variable antes de ejecutar la herramienta. Podemos poner ambas órdenes en una sola línea, usando el separador. Por ejemplo, introdúzcase la siguiente línea de comando en el *área de entrada*.

```
DelVar ap:sq\port("ap,1,2,0",1,2)
```

Como la variable `ap` ha sido borrada, ahora sí podemos emplear esta variable como nombre del bipuerto. Presionamos ¸. Especificamos `b` como tipo de bipuerto deseado, `p` como nombre y `DC` como análisis. A continuación mostramos las respuestas.

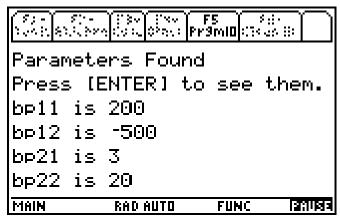

Figura 68. Parámetros B encontrados, en la TI-89.

Así, hemos encontrado los cinco equivalentes del bipuerto que nos fue entregado.

7.7 Verificación

Pero, ¿cómo podríamos verificar que estos bipuertos son los equivalentes correctos de nuestro bipuerto original? Sencillo. Encontremos el equivalente en parámetros de impedancia, es decir en el tipo original, usando como bipuerto de partida el bipuerto `bp`, que fue el último equivalente encontrado. Cualquier error que se haya generado en el camino, impactará nuestra respuesta final. Así, si obtenemos ahora el equivalente correcto, significará que todos los equivalentes intermedios fueron correctos. Introdúzcase la siguiente línea de comando en el *área de entrada*.

```
sq\port("bp,1,2,0",1,2)
```

Presionamos ¸. Especificamos `z` como tipo de bipuerto deseado, `p` como nombre y `DC` como análisis. A continuación mostramos las respuestas.

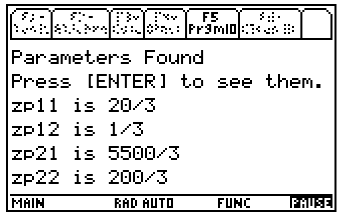

Figura 69. Parámetros Z encontrados, en la TI-89.

Podemos ver que estos parámetros de impedancia coinciden perfectamente con los que nos entregó el problema como datos iniciales. Esto significa que todos los equivalentes que hemos encontrado son correctos. Veamos más problemas resueltos, como práctica para dominar el uso de los bipuertos en el *Symbulator*.

Problema N° 040

Planteamiento. Los parámetros del bipuerto mostrado son zp11=20, zp12=3, zp21=100, zp22=5. Encuentre: a) la impedancia de entrada y las ganancias de corriente, voltaje y potencia del bipuerto, b) la impedancia de salida de la red, vista por la carga.

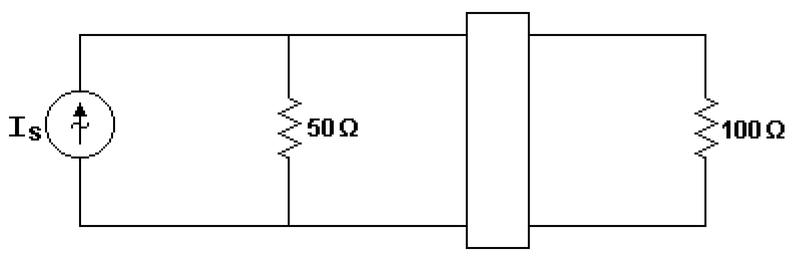

Figura 70. Circuito para el Problema N° 040.

Solución. A continuación, la descripción que utilizamos para la primera pregunta:

```
20.!zp11:3.!zp12:100.!zp21:5.!zp22:sq\dc("j1,0,1,is;r1,
1,0,50.;zp,1,2,0;r2,2,0,100."):sq\gain()
```

En esta línea, está la definición de los parámetros, la órden de ejecutar la simulación y la órden de ejecutar la herramienta `gain`. Presionamos ¸. Tras las frases, aparece el formulario de la herramienta, en el cual introducimos `v1`, `izp1`, `v2`, `izp2`. Presionamos ¸ . Recibimos en la pantalla, y en la memoria, las respuestas: `ai=-.952`, `av=5.56`, `ap=5.29`, `zi=17.1`.

A continuación, la descripción para la segunda pregunta:

```
sq\thevenin("r1,1,0,50.;zp,1,2,0",2,0)
```

Nótese que la fuente de corriente se eliminó. Esta forma de *matarla* es equivalente a haberla introducido con valor nulo, y tiene la ventaja de ahorrarle molestias al simulador. Presionamos ¸ . Seleccionamos `Passive` y `DC`, y presionamos ¸ . Obtenemos: `zth=.714`. Estas son las respuestas correctas.

Problema N° 041

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de admitancia.

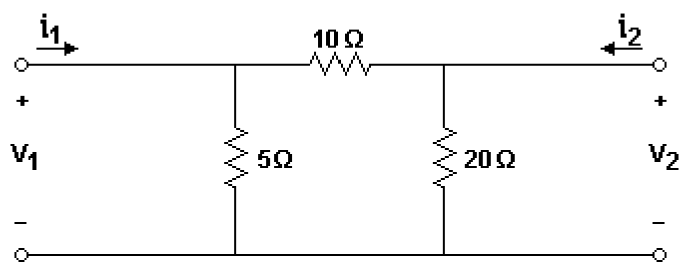

Figura 71. Circuito para el Problema N° 041.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,2,10.;r2,1,0,5.;r3,2,0,20.",1,2)
```

Presionamos ¸ . Seleccionamos `y`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de admitancia: `.3`, `-.1`, `-.1`, `.15`.

Problema N° 042

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de admitancia.

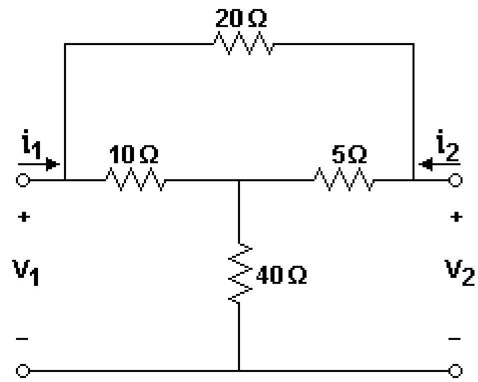

Figura 72. Circuito para el Problema N° 042.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,2,10.;r2,2,3,5;r3,1,3,20;r4,2,0,40",1,3)
```

Presionamos ¸ . Seleccionamos `y`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de admitancia: `.1192`, `-.1115`, `-.1115`,`.1269`.

Problema N° 043

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de admitancia.

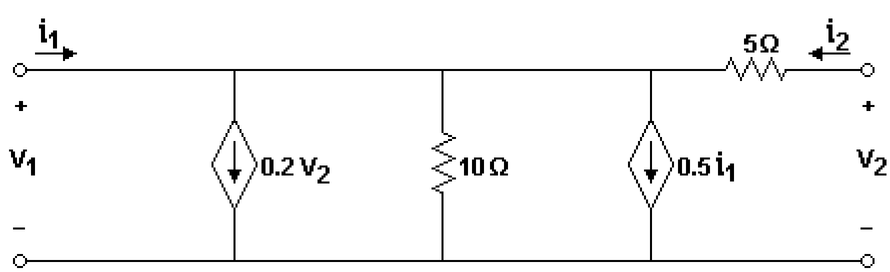

Figura 73. Circuito para el Problema N° 043.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("s1,x,1,0;j1,1,0,.2*v2;r1,1,0,10.;j2,1,0,.5*is1
;r2,2,1,5.",x,2)
```

Nótese que introdujimos un cortocircuito con el propósito de utilizar su corriente como corriente de control para la fuente `j2`. Presionamos ¸ . Seleccionamos `y`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de admitancia: `.6`, `0.` , `-.2`,`.2`.

Problema N° 044

Planteamiento. El siguiente circuito es el equivalente lineal de un transistor en la configuración de emisor común con retroalimentación resistiva entre el colector y la base. Encuentre el equivalente, en parámetros de admitancia.

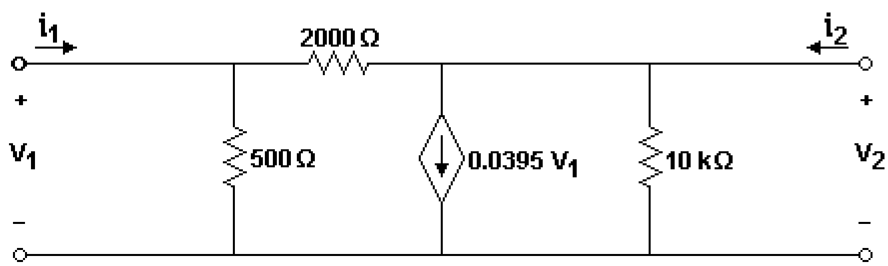

Figura 74. Circuito para el Problema N° 044.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,0,5E2;r2,1,2,2E3;j1,2,0,.0395*v1;
r3,2,0,1E4",1,2)
```

Presionamos ¸ . Seleccionamos `y`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de admitancia: `.0025`, `-5.E-4`, `.039`, `6.E-4`.

Problema N° 045

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de impedancia.

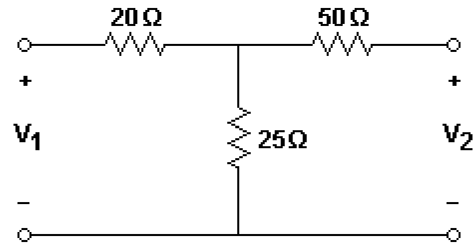

Figura 75. Circuito para el Problema N° 045.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,3,20.;r2,3,2,50.;r3,3,0,25.",1,2)
```

Presionamos ¸ . Seleccionamos `z`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de impedancia: `45.`, `25.`, `25.`, `75.`.

Problema N° 046

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de impedancia.

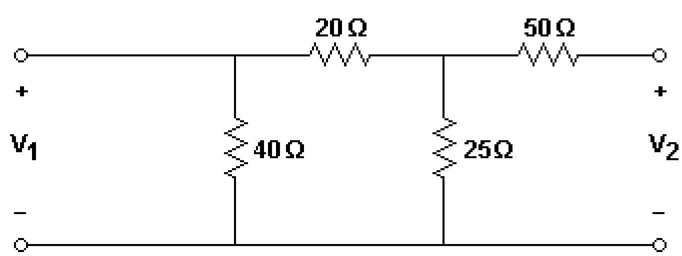

Figura 76. Circuito para el Problema N° 046.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,0,40.;r2,1,3,20.;r3,3,0,25.;r4,3,2,50.",1,2)
```

Presionamos ¸ . Seleccionamos `z`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de impedancia: `21.2`, `11.8`, `11.8`, `67.6`.

Problema N° 047

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de impedancia.

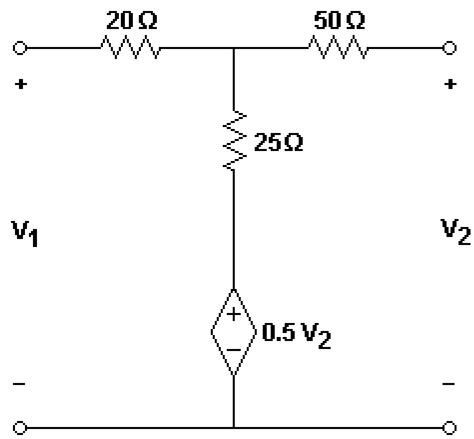

Figura 77. Circuito para el Problema N° 047.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,3,20.;r2,3,2,50.;r3,3,4,25.;e1,4,0,.5*v2",1,2)
```

Presionamos ¸ . Seleccionamos `z`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de impedancia: `70.`, `100.`, `50.`, `150.`.

Problema N° 048

Planteamiento. Este problema no tiene figura. Encuentre la impedancia de entrada y las ganancias de corriente, voltaje y potencia de un bipuerto, con parámetros de impedancia z11=4, z12=1.5, z21=10 and z22=3, si tiene una fuente de entrada Vs en su entrada en serie con una resistencia de 5Ω, mientras que la carga es de 2Ω.

Solución. A continuación, la orden que utilizamos:

```
4!zp11:1.5!zp12:10!zp21:3!zp22:sq\dc("e1,1,0,vs;r1,1,2,5;r2,3,0,2;zp,2,3,0"):sq\gain()
```

En esta línea está la definición de los parámetros, la órden de ejecutar la simulación y la órden de ejecutar la herramienta `gain`. Presionamos ¸ . Tras las frases, aparece el formulario de la herramienta, en el cual introducimos `v2`, `izp2`, `v3`, `izp3`. Presionamos ¸ . Recibimos en la pantalla y en la memoria, las respuestas: `ai=-2.`, `av=4.`, `ap=8.`, `zi=1.`

Problema N° 049

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros híbridos.

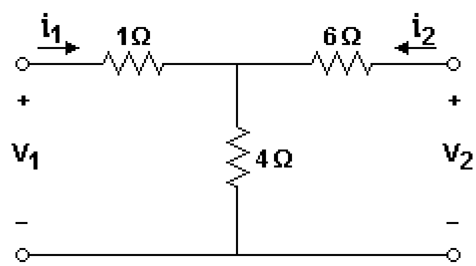

Figura 78. Circuito para el Problema N° 049.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,3,1.;r2,3,2,6.;r3,3,0,4.",1,2)
```

Presionamos ¸ . Seleccionamos `h`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros híbridos: `3.4`, `.4`, `-.4`, `.1`.

Problema N° 050

Planteamiento. El siguiente circuito es el equivalente de un transistor de alta frecuencia. Encuentre el equivalente, en parámetros de impedancia.

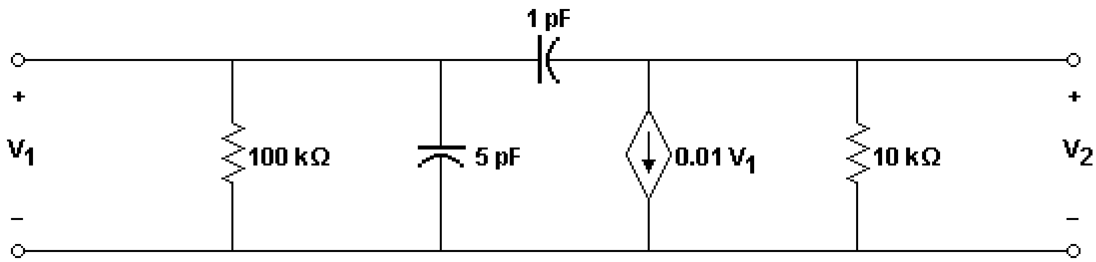

Figura 79. Circuito para el Problema N° 050.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,0,1E5;ca,1,0,5E-12;cb,1,2,1E-12;
j1,2,0,.01*v1;r2,2,0,1E4",1,2)
```

Presionamos ¸. Seleccionamos `z`, algún nombre y `AC`, y presionamos ¸. Como frecuencia, introducimos `1E8`, y presionamos ¸. Obtenemos los siguientes parámetros de impedancia: `89.7-98.4*i`, `94.1-4.34*i`, `528.+9401.*i`, `564.-35.5*i`. Como estos valores están almacenados en la memoria, podemos usar la herramienta `absang` para verlos en forma polar: `133.1`∠`-47.6`°, `94.2`∠`-2.64`°, `9416.`∠`86.8`°, `565.`∠`-3.60`°.

<u>Problema N° 051</u>

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de impedancia.

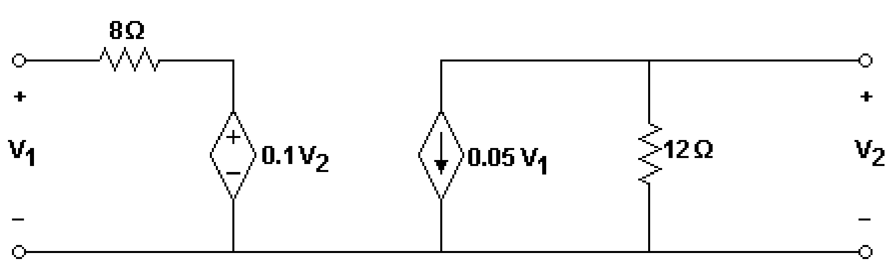

Figura 80. Circuito para el Problema N° 051.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,3,8.;e1,3,0,.1*v2;j1,2,0,.05*v1;r2,2,0,12
.",1,2)
```

Presionamos ¸ . Seleccionamos `z`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de impedancia: `7.55`, `1.13`, `-4.53`, `11.3`.

Problema N° 052

Planteamiento. El siguiente circuito es el equivalente de un transistor de alta frecuencia. Encuentre el equivalente, en parámetros híbridos.

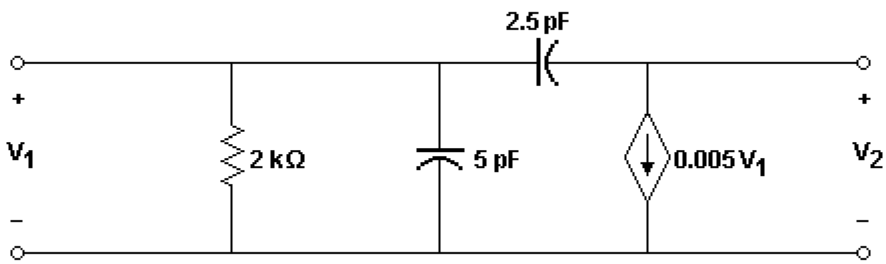

Figura 81. Circuito para el Problema N° 052.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,0,2E3;ca,1,0,5E-12;cb,1,2,2.5E-12;
j2,2,0,.005*v1",1,2)
```

Presionamos ¸ . Seleccionamos `h`, algún nombre y `AC`, y presionamos ¸ . Como frecuencia, introducimos `1E8`, y presionamos ¸ . Obtenemos los siguientes parámetros híbridos: `615.-923.*i`, `.231+.154*i`, `2.85-4.77*i`, `.00119+9.62E-4*i`. Como estos valores están almacenados en la memoria, podemos usar la herramienta `absang` para verlos en forma polar: $1109\angle{-56.3}°$, $.277\angle 33.7°$, $5.55\angle{-59.2}°$, $.00153\angle 38.9°$.

Problema N° 053

Planteamiento. Encuentre a) el equivalente de la siguiente red, en parámetros híbridos, y b) la impedancia de salida de la red, vista desde la perspectiva de la carga, si la entrada contiene Vs en serie con Rs=200Ω.

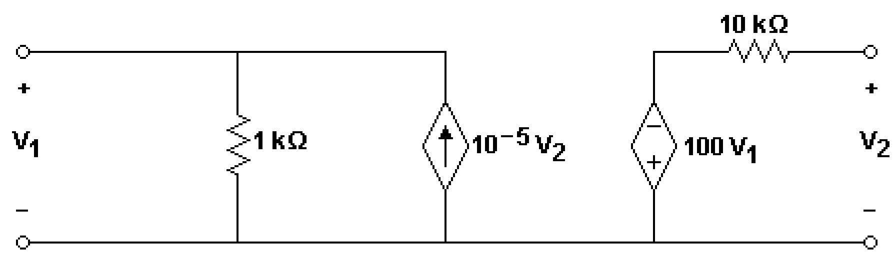

Figura 82. Circuito para el Problema N° 053.

Solución:

A continuación, la descripción que utilizamos para la primera parte:

```
sq\port("r1,1,0,1E3;j1,0,1,1E-5*v2;e1,0,3,100*v1;
r2,3,2,1E4",1,2)
```

Presionamos ¸ . Seleccionamos `h`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros híbridos: `1000.`, `.01`, `10.`, `2.E-4`.

A continuación, la descripción que utilizamos para la segunda parte:

```
sq\thevenin("rs,0,1,200;r1,1,0,1E3;j1,0,1,1E-
5*v2;e1,0,3,100*v1;r2,3,2,1E4",2,0)
```

Nótese que hemos agregado únicamente la resistencia `rs`. No se agregó la fuente `vs` pues, para encontrar la impedancia de salida, la fuente de voltaje debe estar muerta, y una fuente de voltaje muerta es simplemente un cortocircuito. No es necesario colocar este cortocircuito en serie con la resistencia, pues colocar sólo la resistencia tiene el mismo efecto. Presionamos ¸ . Seleccionamos `Passive` y `DC`, y presionamos ¸ . Vemos que `zth` es `8571.`

Problema N° 054

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de transmisión.

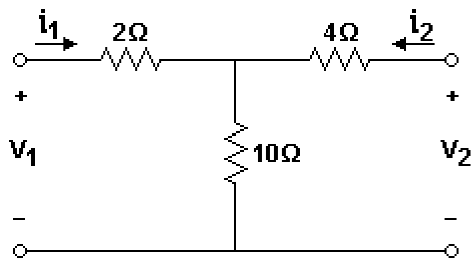

Figura 83. Circuito para el Problema N° 054.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,3,2.;r2,3,2,4.;r3,3,0,10.",1,2)
```

Presionamos ¸ . Seleccionamos `a`, algún nombre y `DC`, y presionamos ¸ . Obtenemos los siguientes parámetros de transmisión: `1.2`, `6.8`, `.1`, `1.4`.

Problema N° 055

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de transmisión.

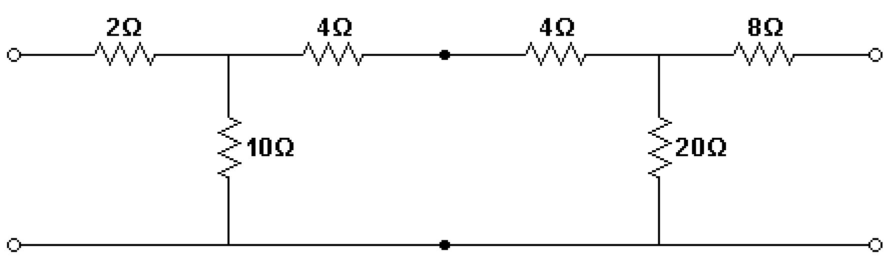

Figura 84. Circuito para el Problema N° 055.

Solución. Nótese que la red, en este caso, está compuesta por dos grupos de resistencias en arreglo T. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,2,2.;r2,2,0,10.;r3,2,3,4.;r4,3,4,4.;r5,4,
0,20.;r6,4,5,8.",1,5)
```

Presionamos ¸ . Seleccionamos a, algún nombre y DC, y presionamos ¸ . Obtenemos los siguientes parámetros de transmisión: 1.78, 25.84, .19, 3.32. Este es un ejemplo de cómo, con la herramienta port, se pueden reducir a un sólo bipuerto una red que usualmente hubiese tomado dos bipuertos.

<u>Problema N° 056</u>

Planteamiento. Encuentre el equivalente de la siguiente red, en parámetros de transmisión.

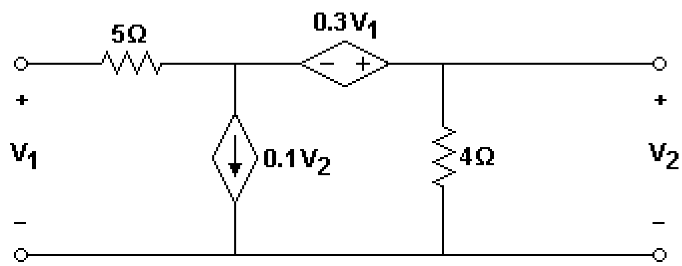

Figura 85. Circuito para el Problema N° 056.

Solución. A continuación, la descripción que utilizamos:

```
sq\port("r1,1,3,5.;j1,3,0,.1*v2;e1,2,3,.3*v1;r2,2,0,4."
,1,2)
```

Presionamos ¸ . Seleccionamos a, algún nombre y DC, y presionamos ¸ . Obtenemos los siguientes parámetros de transmisión: 2.12, 3.85, .35, 1..

Capítulo 8

Simulación simbólica en corriente alterna

8.1 Conceptos generales

La simulación simbólica en corriente alterna utiliza las mismas técnicas que la simulación simbólica en corriente directa, con dos únicas excepciones.

8.2 Comando `cSolve`

Para resolver ecuaciones complejas, la calculadora TI posee el comando `cSolve`, el cual funciona de forma idéntica al comando `solve`. El comando `cSolve` es un aliado poderoso en la obtención de respuestas numéricas a partir de una simulación simbólica, cuando las ecuaciones involucradas son complejas. En otros casos, cuando las ecuaciones que se usan son reales, se puede usar el comando `solve`.

8.3 Incógnitas complejas

Para encontrar el valor numérico de una incógnita compleja, debemos encontrar dos valores numéricos: un valor real y un valor imaginario. Por lo tanto, una incógnita compleja es en verdad dos incógnitas: una real y una imaginaria.

Igualmente, al tener un valor complejo como valor dado conocido, en verdad tenemos dos valores conocidos: un valor real y un valor imaginario. En nuestro primer problema de este capítulo, veremos como encontrar el valor numérico de dos incógnitas simbólicas, usando sólo un valor complejo conocido.

Problema N° 057

Planteamiento. Este problema no tiene figura. Un circuito consiste de una resistencia en serie con una capacitancia. Encuentre sus valores, si al aplicar un voltaje de 240V RMS y 200Hz, la corriente tomada de la fuente es de 1.2+1.6***i***. Nótese que esta corriente no está en RMS.

Solución. Primero, simulamos el circuito según el tratamiento riguroso:

```
sq\ac("e,1,0,240.√(2);r,1,2,r;c,2,0,c",2π200.)
```

Como sólo hay un elemento de cada tipo, hemos aprovechado para nombrarlos e, r y c, que es una notación lícita y muy elegante. Si hubiese más de un elemento de cada tipo, habría que nombrarlos de forma más completa. Presionamos ¸ . Tras las frases, aparece **`Done`**. A continuación, ordenamos a la calculadora encontrar los valores de `r` y `c` que cumplan tanto con la parte real como con la parte imaginaria, de la corriente que nos entregó el problema como dato conocido.

```
solve(real(ir)=1.2 and imag(ir)=1.6,{r,c})
```

Esta línea de comando es muy interesante. Veamos qué significa. Acabamos de realizar una simulación, usando `r` y `c` como valores simbólicos. Por lo tanto, todas las respuestas, incluyendo la corriente a través de los elementos, están expresadas como funciones de `r` y `c`. Estas respuestas tienen una parte real y una parte imaginaria. También la corriente que nos ha dado el circuito como valor conocido tiene una parte real y una parte imaginaria. Lo que la línea de comando está ordenando a la calculadora es: *"Utiliza el comando* `solve` *para encontrar qué valores de* `r` *y* `c` *cumplen con las siguientes dos condiciones: que la parte real de la corriente sea igual a 1.2 amperios, y la parte imaginaria de la corriente sea igual a 1.6 amperios."* Usamos el comando `solve` y no el comando `cSolve,` porque las dos ecuaciones involucradas están dadas en términos de números reales. ¿Por qué? La parte real de un número complejo es un número real. La parte imaginaria de un número complejo, una vez que se ha extraído con el comando `imag`, también es un número real. Por lo tanto, las dos ecuaciones dadas están compuestas de números reales, y se puede usar el comando `solve` para resolverlas.

Presionamos ¸ . La calculadora nos entrega dos pares de respuestas. Uno de estos pares tiene un valor de r que es negativo. El otro par de respuestas tiene un valor de r positivo. Obviamente, el par de respuestas correcto es aquel que tiene el valor de r positivo. Estos valores, redondeados a tres cifras, son: `r=102. and c=5.86E-6`. El valor de `r` está dado en Ω, y el de `c` en F. Si queremos obtener únicamente las respuestas correctas, señalamos que r>0, con esta línea:

```
solve(real(ir)=1.2 and imag(ir)=1.6,{r,c})|r>0
```

Problema N° 058

Planteamiento. Si ω=500 rad/seg y I_L=2.5∠40°, encuentre *vs*(t).

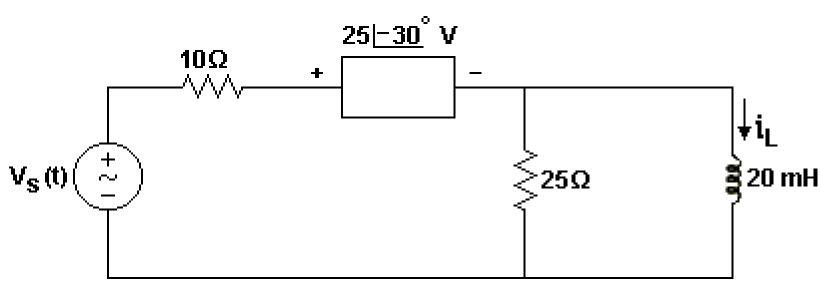

Figura 86. Circuito para el Problema N° 058.

Solución. Usaremos una simulación simbólica y luego la herramienta `solves`:

```
sq\ac("es,1,0,vs;r1,1,2,10;e2,2,3,(25∠-30°);r2,3,0,25;
l1,3,0,.02",500):sq\solves()
```

Presionamos ¸ . Tras las frases, aparece **Done**. A continuación, se ejecuta la herramienta. Introducimos `il1=(2.5∠40°)` como ecuación y `vs` como incógnita.

Obtenemos el valor de `vs` en pantalla, pero está en forma rectangular y así no nos es útil. Presionamos ¸ para volver a la *pantalla hogar*. Para ver el valor en forma polar, usamos `sq\absang(vs)` en la línea de entrada. Obtenemos `35.5∠58.9°`. Por lo tanto, el valor de *vs*(t) es 35.5cos(500t+58.9°) V.

<u>Problema N° 059</u>

Planteamiento. Este problema no tiene figura. Una corriente fasorial de 1∠0° A fluye a través de la combinación en serie de 1Ω, 1H y 1F. ¿A qué frecuencia la amplitud del voltaje a través de la red es dos veces la amplitud del voltaje a través del resistor?

Solución. Primero, simulamos el circuito:

```
sq\ac("j,0,1,1;r,1,2,1;l,2,3,1;c,3,0,1",ω)
```

Como sólo hay un elemento de cada tipo, hemos aprovechado para nombrarlos sólo con una letra. Presionamos ¸. Tras las frases, aparece **`Done`**. En este problema, es posible encontrar más de una respuesta, por lo tanto lo más adecuado es usar el comando `solve` en vez de la herramienta `solves`, porque el comando nos dará las múltiples respuestas que encuentre, mientras que la herramienta sólo nos dará una respuesta. Así, usamos el comando.

```
solve(abs(v1)=2*abs(vr),ω)|ω>0
```

Nótese que suprimimos la posibilidad de obtener frecuencias negativas, con la condicional al final de la línea del comando. Obtendremos dos respuestas, y ambas son válidas. Si presionamos ¸, obtendremos las respuestas en forma exacta: `ω=(√(7)-√(3))/2 or ω=(√(7)+√(3))/2`. Si presionamos ¥ ¸, obtendremos las respuestas en forma aproximada: `ω=2.19 or ω=.457`.

<u>Problema N° 060</u>

Planteamiento. Este problema no tiene figura. Un inductor de 20mH y un resistor de 30Ω están en paralelo. Encuentre la frecuencia en que: a) abs($Z_{entrada}$)=25Ω, b) ang($Z_{entrada}$)=25°, c) real($Z_{entrada}$)=25Ω, d) imag($Z_{entrada}$)=10Ω.

Solución. Usamos la herramienta `thevenin`:

```
sq\thevenin("r,1,0,30;l,1,0,.02",1,0)
```

Presionamos ¸ . A continuación, se ejecuta la herramienta. Seleccionamos `Passive`, `AC` y presionamos ¸ . Introducimos una frequencia simbólica `w` y presionamos ¸ . Tras las frases, aparece **`zth`** como función de `w`. Presionamos ¸ para volver a la *pantalla hogar*. Para responder las preguntas, usamos:

```
solve(abs(zth)=25,w)|w>0
```

Obtenemos `w=2261.`

```
solve(angle(zth)=25°,w)|w>0
```

Obtenemos `w=3217.`

```
solve(real(zth)=25,w)|w>0
```

Obtenemos `w=3354.`

```
solve(imag(zth)=10,w)|w>0
```

Obtenemos `w=3927. or w=572.9`

<u>Problema N° 061</u>

Planteamiento. En la red mostrada, encuentre la frecuencia en que a) $R_{entrada}$=550Ω, b) $X_{entrada}$=50Ω, c) $G_{entrada}$=1.8mS, d) $B_{entrada}$=-150µS. Recuérdese que Z=R+***i***X y Y=G+***i***B

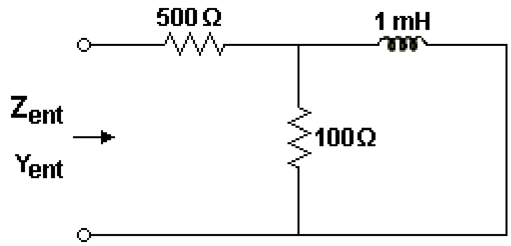

Figura 87. Circuito para el Problema N° 061.

Solución. Usamos la herramienta `thevenin`:

```
sq\thevenin("r1,1,2,500;r2,2,0,100;l1,2,0,1E-3",1,0)
```

Presionamos ¸ . A continuación, se ejecuta la herramienta. Seleccionamos `Passive`, `AC` y presionamos ¸ . Introducimos una frequencia simbólica `w` y presionamos ¸ . Tras las frases, aparece **`zth`** como función de `w`. Presionamos ¸ para volver a la *pantalla hogar*. Para responder las preguntas, usamos:

```
solve(real(zth)=550,w)|w>0
```

Obtenemos `w=100000.`

```
solve(imag(zth)=50,w)|w>0
```

Obtenemos `w=100000.`

```
solve(real(1/zth)=.0018,w)|w>0
```

Obtenemos `w=102062.`

```
solve(imag(1/zth)=-1.5E-4,w)|w>0
```

Obtenemos `w=132953. or w=52232.`

<u>Problema N° 062</u>

Planteamiento. Encuentre $Y_{entrada}$ como función de ω para la red mostrada.

Encuentre también la frecuencia de resonancia ω_0 y el valor de $Z_{entrada}$ para esa frecuencia de resonancia.

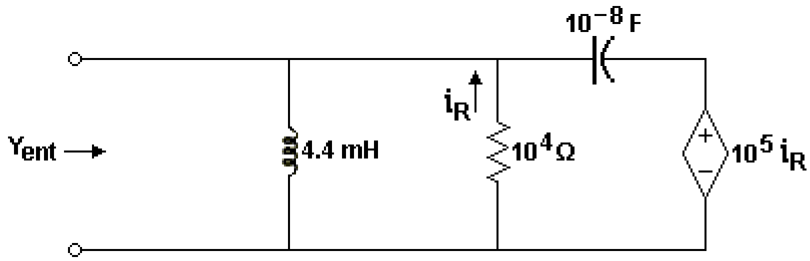

Figura 88. Circuito para el Problema N° 062.

Solución. La teoría nos enseña que la frecuencia de resonancia es aquella frecuencia en la cual la parte imaginaria de la admitancia de entrada es igual a cero. Como sabemos, la admitancia de entrada es el inverso de la impedancia de entrada. Usamos la herramienta `thevenin`:

```
sq\thevenin("l,1,0,4.4E-3;r,0,1,1E4;c,1,2,1E-8;
e,2,0,1E5*ir",1,0)
```

Presionamos ¸ . A continuación, se ejecuta la herramienta. Seleccionamos `Passive`, `AC` y presionamos ¸ . Introducimos una frequencia simbólica `w` y presionamos ¸ . Tras las frases, aparece **`zth`** como función de `w`. Presionamos ¸ para volver a la *pantalla hogar*. Para responder las preguntas, usamos:

```
solve(imag(1/zth)=0,w)|w>0
```

Con la orden que está sobre estas líneas, estamos preguntántole a la calculadora: *"Usando el comando* `solve`*, dime cuánto vale* `w` *cuando la parte imaginaria del inverso de* `zth` *es cero, considerando a* `w` *como positiva."* Presionamos ¸ . Obtenemos `w=45454.5` como frecuencia de resonancia.

```
zth|ans(1)
```

Con la orden que está sobre estas líneas, estamos preguntántole a la calculadora: *"Dime cuánto vale* `zth` *cuando la respuesta inmediatamente anterior es cierta."* Colocar `zth|ans(1)` es equivalente a colocar `zth|w=45454.5`. El comando `ans(#)` evoca la $\#^{va}$ respuesta del área de historia. Presionamos ¸ . Obtenemos `w=9999.9`. Esta respuesta puede redondearse a 10KΩ.

Problema N° 063

Planteamiento. Para la red mostrada, encuentre la frecuencia resonante ω_0 y el valor de $Z_{entrada}$ para esa frecuencia.

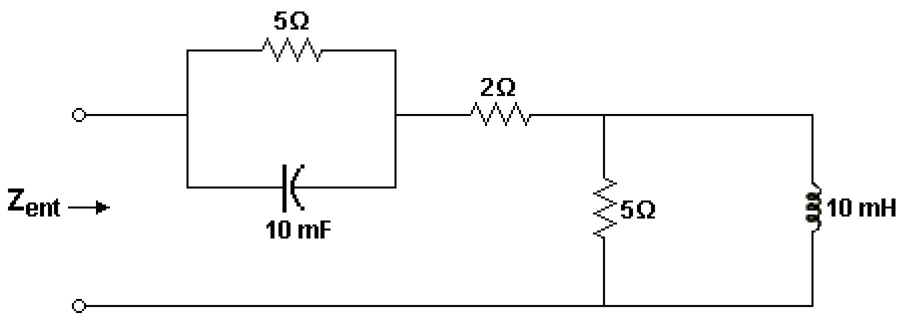

Figura 89. Circuito para el Problema N° 063.

Solución. La técnica ya la conocemos:

```
sq\thevenin("r1,1,2,5;c,1,2,.01;r2,2,3,2;r3,3,0,5;l,3,0,.01",1,0)
```

Presionamos ¸ . A continuación, se ejecuta la herramienta. Seleccionamos `Passive`, `AC` y presionamos ¸ . Introducimos una frequencia simbólica `w` y presionamos ¸ . Tras las frases, aparece **`zth`** como función de `w`. Presionamos ¸ para volver a la *pantalla hogar*. Para responder las preguntas, usamos:

```
solve(imag(1/zth)=0,w)|w>0
```

Presionamos ¸ . Obtenemos `w=100.` como frecuencia de resonancia.

```
zth|ans(1)
```

Presionamos ¸ . Obtenemos `2.38`. En el siguiente capítulo, aprenderemos sobre la simulación experta en corriente alterna.

Capítulo

9

Algunos conceptos nuevos

9.1 Problemas con gráficas

En ocasiones, es necesario construir una grafica con las respuestas. Existen, básicamente, cuatro tipos de gráficas relacionadas con una simulación del *Symbulator*:

1. Valor de una respuesta obtenida en un análisis AC, en función de la frecuencia $j\omega$.
2. Valor de una respuesta obtenida en un análisis FD, en función de la frecuencia compleja *s*.
3. Valor de una respuesta obtenida en un análisis TR, en función del tiempo *t*.
4. Valor de cualquier respuesta, en función de un valor de circuito simbólico.

La calculadora TI tiene extensas capacidades de gráficación. La descripción detallada de su diversidad y uso está más allá del propósito de este escrito; sin embargo, está disponible en el Manual de Usuario de la calculadora.

En el Capítulo 12 veremos un ejemplo del segundo tipo de gráfica usando la herramienta `bode`, y un ejemplo del tercer tipo de gráfica, usando la herramienta `plot`.

A continuación, veremos un ejemplo del primer tipo de gráfica, usando las capacidades propias de la calculadora. La técnica que se expondrá es aplicable también para el cuarto tipo de gráficas.

<u>Problema N° 064</u>

Planteamiento. Grafique abs(V_{Salida}) contra ω para el circuito que se muestra en la figura. Cubra un intervalo de frecuencia de ω=800 a ω=1200 rad/seg.

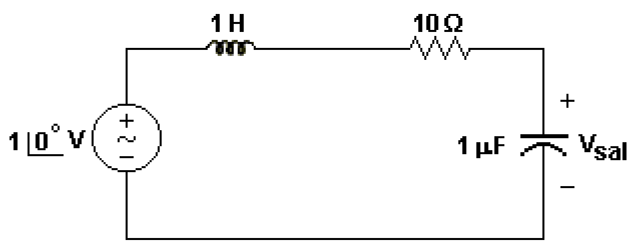

Figura 90. Circuito para el Problema N° 064.

Solución. Simplemente usaremos el comando `solve`:

```
sq\ac("e,1,0,1;l,1,2,1;r,2,3,10;c,3,0,1E-6",x)
```

Se usó una frecuencia x con la intención de que el voltaje de salida quede como función de la variable x. El propósito de esto es satisfacer el requerimiento de la calculadora TI, la cual requiere que las funciones y# que se vayan a graficar, donde # es un número, estén dadas como función de x, donde x es la variable independiente.

Presionamos ¸ . La simulación toma algo de tiempo. Finalmente aparece **`Done`**. El siguiente paso es hacer aparecer en el *área de historia*, el valor de abs(V_{Salida}). Podemos usar como V_{Salida} a `v3` o a `vc`.

```
abs(v3)
```

Presionamos ¸ . Obtenemos una función de x, la cual almacenaremos en cualquiera de las funciones y#(x), donde # es un número.

```
ans(1)!y1(x)
```

Presionamos ¸ y la función se almacena en la variable especificada. Presionando ¥ f podemos verificar que así es. Ahora podemos especificar el rango de frecuencias para el cual deseamos graficar. Presionamos ¥ „ y especificamos para `xmin` un valor de `800` y para `xmax` un valor de `1200`. Presionamos ¥ … y vemos la gráfica. Podemos presionar „ j A en la TI-89 y „ A en la TI-92+ para ajustar el alto de la gráfica al mejor valor:

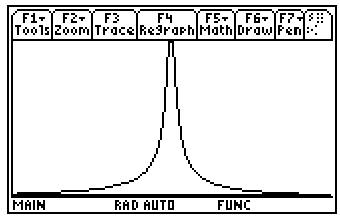

Figura 91. Gráfica obtenida para el problema, en una TI-89.

Hemos aprendido cómo graficar una respuesta como función de la frecuencia, usando las capacidades propias de la calculadora.

9.2 Problemas con múltiples incógnitas

Para resolver un circuito con valores simbólicos, necesitamos tener una respuesta conocida de antemano por cada valor simbólico cuyo valor numérico es desconocido y se desee obtener. Para resolver un sistema, requerimos el mismo número de ecuaciones que de incógnitas. Veremos un problema en el cual ambos axiomas se cumplen al mismo tiempo, pero antes, una aclaración teórica. Explicamos en el Capítulo 7 que antes de simular un circuito que contenga un bipuerto, debemos haber almacenado en la memoria los valores de los cuatro parámetros de este bipuerto, en los nombres correspondientes. Pero, ¿qué sucede si no almacenamos estos valores? El *Symbulator* simulará el circuito usando los parámetros que no estén almacenados antes de la simulación, como fueran variables simbólicas.

Problema N° 065

Planteamiento. Complete la tabla dada, y también dé los valores de los parámetros ***y***.

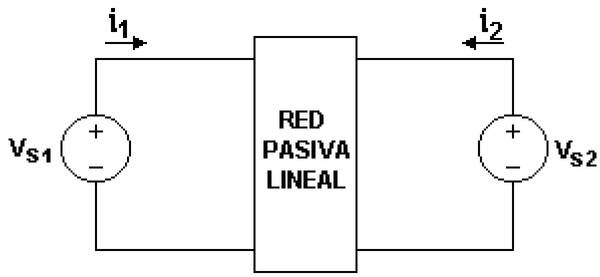

Figura 92. Circuito para el Problema N° 065.

Solución:

Este es un problema muy especial, el cual nos permite disfrutar de las capacidades simbólicas del *Symbulator*. Existen varias formas de resolverlo. Aquí presentaremos la manera más lineal y simple, que sin embargo no es la más sofisticada.

Aunque el planteamiento nos presente las dos preguntas en ese orden, para responderlas nosotros las invertiremos. Es decir, primero encontraremos los valores de los parámetros ***y***, y luego completaremos la tabla. Para encontrar los parámetros ***y***, aprovecharemos las dos primeras filas de la tabla, las cuales están completas ya. Es precisamente gracias a estas filas completas que podremos generar las cuatro ecuaciones, necesarias para encontrar los cuatro valores numéricos de los parámetros ***y11, y12, y21*** y ***y22***. Los pasos que seguiremos son los siguientes:

1. Simular el circuito, sin parámetros definidos.
2. Usar las dos filas completas de la tabla para generar las cuatro ecuaciones que necesitamos.
3. Obtener los cuatro valores numéricos de los parámetros, resolviendo estas cuatro ecuaciones.
4. Completar la tabla, usando los parámetros que hemos encontrado y los valores que nos dan las filas incompletas.

El procedimiento es como sigue:

```
sq\dc("j1,0,1,j1;j2,0,2,j2;yp,1,2,0")
```

Presionamos ¸ . La simulación se ejecuta y aparece **Done**. Generamos la primera ecuación con la siguiente orden:

```
v1=100|j1=5 and j2=-32.5
```

Con ¸ obtenemos la ecuación. Generamos la segunda ecuación así:

```
v2=50|j1=5 and j2=-32.5
```

Con ¸ obtenemos la ecuación. Generamos la tercera ecuación así:

```
v1=50|j1=-20 and j2=-5
```

Con ¸ obtenemos la ecuación. Generamos la cuarta ecuación así:

```
v2=100|j1=-20 and j2=-5
```

Habiendo generado las cuatro ecuaciones, procedemos a resolverlas para encontrar los cuatro parámetros ***y***.

```
solve(ans(1) and ans(2) and ans(3) and ans(4), {yp11,yp12,yp21,yp22})
```

Con ¸ obtenemos los parámetros: `yp11=.2 and yp12=-.3 and yp21=-.4 and yp22=.15`. Ahora, procedemos a completar la tabla. Para completar la tercera fila, usamos la siguiente orden:

```
solve(v1=20 and v2=0,{j1,j2})|ans(1)
```

Con ¸ obtenemos los valores faltantes en la fila: `j1=4. and j2=-8.` Para completar la cuarta fila, usamos las siguientes órdenes:

```
v1|ans(2) and j1=5 and j2=0

v2|ans(3) and j1=5 and j2=0
```

Nótese cómo el número # que aparece en `ans(#)`cambia para reflejar a cada momento la nueva posición de las respuestas de los parámetros ***y***. Con ¸ obtenemos los valores `-8.33333` y `-22.22222`, respectivamente. Para completar la quinta fila, usamos las siguientes órdenes:

```
v1|ans(4) and j1=5 and j2=15

v2|ans(5) and j1=5 and j2=15
```

Con ¸ obtenemos los valores `-58.33333` y `-55.55555`, respectivamente. Así, hemos completado la tabla.

Personalmente, considero que éste es uno de los problemas más interesantes que he resuelto jamás, usando el *Symbulator*.

Aprendamos a usar un nuevo elemento.

*9.3 Amplificador operacional ideal (***op-amp** *ideal)*

Un op-amp ideal es, en esencia, una fuente de voltaje dependiente de voltaje. Al decir que es ideal, nos referimos a que sus parámetros se consideran como R_i=:, R_o=0 y A=:.

9.3.1 Notación

En el caso del op-amp ideal, la mínima información necesaria para describirlo por completo es la siguiente:

1. *Identificación.* La letra que identifica a un op-amp ideal es la **o**.

2. *Nombre.* Se le debe dar un nombre cualquiera, que empiece con la letra **o**.

3. *Nodos.* Un op-amp ideal, tal y como lo entiende el *Symbulator*, tiene tres puntos de conexión: uno positivo, uno negativo y uno de salida.

Así, los op-amp ideales se definen como se muestra a continuación:

o*nombre, nodo +, nodo -, nodo salida*

La descripción del op-amp ideal siempre tiene 4 términos, por lo que podría haber necesidad de agregar un cero al final.

9.3.2 Respuestas relacionadas

En el caso de un op-amp ideal, se entrega una respuesta relacionada: la corriente que sale del op-amp por el nodo de salida. Esta corriente se almacena en una variable llamada `i`*nombre*, donde *nombre* es el nombre del op-amp ideal. Así, para un op-amp ideal que se llame `o1`, la respuesta relacionada será `io1`. Recuérdese que esta corriente se considera saliendo del op-amp.

Veamos algunos ejemplos.

Problema N° 066

Planteamiento. Dado el siguiente circuito, encuentre el voltaje en el nodo de salida del op-amp ideal.

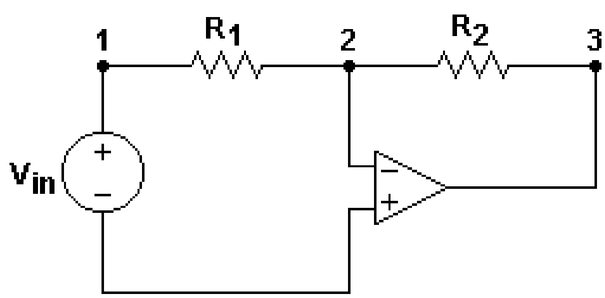

Figura 93. Circuito para el Problema N° 066.

Solución. Tras limpiar las variables simbólicas que vamos a emplear, ordenamos la simulación, y solicitamos el voltaje en el nodo de salida del op-amp.

```
DelVar r1,r2,vin:sq\dc("e1,1,0,vin;r1,1,2,r1;r2,2,3,r2;
o1,0,2,3"):v3
```

Con ¸ , inicia la simulación y obtenemos la respuesta: `-r2*vin/r1`

Problema N° 067

Planteamiento. Si se supone que el op-amp mostrado es ideal, encuentre R_{ent}.

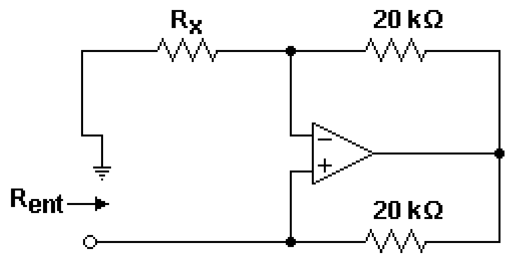

Figura 94. Circuito para el Problema N° 067.

Solución. Usemos la herramienta `thevenin`.

```
DelVar rx:sq\Thevenin("rx,1,0,rx;r1,1,2,2E4;r2,2,3,2E4;
o1,3,1,2",3,0)
```

Con ¸ , se ejecuta la herramienta. Escogemos `Passive` y `DC`. ¸ . Obtenemos la respuesta en la pantalla (y en la memoria): `zth=-rx`

Estos ejemplos ilustran la facilidad con que se simulan los op-amps ideales en el *Symbulator*.

9.4 Consideraciones finales para el Modo (Experto)

Este punto debe ser leído por aquellos usuarios que hayan leído ya el Capítulo 4.

9.4.1 Restricciones del Modo (Experto) *con op-amps*

Como hemos dicho, el *Modo (Experto)* permite al usario escoger si desea resolver las ecuaciones, resolverlas y guardarlas, o simplemente guardarlas. Aquellos circuitos que contengan amplificadores operacionales ideales generarán ecuaciones no comprensibles por el usuario, pues contienen un término que se usará luego para la aplicación del comando de límite de la calculadora. Por lo tanto, cuando el circuito que se resuelva en el *Modo (Experto)* incluya op-amps, debe escogerse siempre Symbulate, y nunca Just Keep, cuando el diálogo nos solicite el procedimiento que deseamos seguir.

9.4.2 Restricciones del Modo (Experto) *en análisis transitorio*

En el Capítulo 10 aprenderemos a realizar simulaciones en dominio de la frecuencia FD. En el Capítulo 11 aprenderemos a realizar simulaciones de análisis transitorio TR. Durante un análisis transitorio, el simulador generará ecuaciones en dominio de la frecuencia. Tras haber resuelto estas ecuaciones, todas las respuestas son convertidas a dominio del tiempo. Por lo tanto, cuando la simulación en el *Modo (Experto)* sea en análisis transitorio, debemos escoger siempre Symbulate, y nunca Just Keep, si deseamos obtener respuestas en el dominio del tiempo. Guardar las ecuaciones generadas y resolverlas luego, nos llevaría a respuestas en el dominio de la frecuencia.

9.4.3 Más variables de primer nivel

Algunos elementos de circuito fueron presentados después del Capítulo 4. Conozcamos cuáles tienen variables de primer nivel relacionadas a ellos.

Inductores. La corriente a través de ellos es una variable de primer nivel. Su caída de voltaje es una variable de segundo nivel que no es reemplazada por su valor al final de la simulación. Su potencia es una variable de tercer nivel.

Capacitores. La corriente a través de ellos es una variable de segundo nivel que sí es reemplazada por su valor al final de la simulación. Su caída de voltaje es una variable de segundo nivel que no es reemplazada por su valor al final de la simulación. Su potencia es una variable de tercer nivel.

Transformadores ideales. La corriente que entra al transformador por el lado primario es una variable de primer nivel. La corriente que entra al transformador por el lado secundario es una variable de segundo nivel que no es reemplazada por su valor al final de la simulación.

Bipuertos. Las corrientes que entran al transformador por ambos extremos son variables de primer nivel.

Amplificadores operacionales ideales. La corriente que sale del op-amp por el nodo de salida es una variable de primer nivel.

Capítulo 10
Simulación en el dominio de la frecuencia

10.1 Puerta para el dominio de la frecuencia: `fd`

Para realizar una simulación en el dominio de la frecuencia compleja *s*, se usa la *puerta* llamada **sq\fd**.

10.2 Dato de entrada

El único dato de entrada que hay que proporcionar a esta herramienta es la descripción del circuito, usando la misma notación que hemos visto hasta ahora. Existe una única diferencia: si el circuito que se va a simular tiene algún inductor o capacitor, entonces la descripción de cada elemento debe tener cinco términos. Aquellas descripciones que sólo requieran de cuatro términos, deberán tener un 0 al final, para mantener la simetría. La simulación en dominio de la frecuencia se ordena así: `sq\fd(circuito)`.

10.3 Condiciones iniciales

El quinto término en la descripción de un inductor o un capacitor es su condición inicial. Inicial quiere decir en tiempo $t=0^+$.

La condición inicial de un inductor es su corriente inicial, y por lo tanto su valor debe darse en amperios. Se considera esta corriente como fluyendo a través del elemento del primer nodo hacia el segundo nodo en la descripción.

La condición inicial de un capacitor es su voltaje inicial, y por lo tanto su valor debe darse en voltios. Se considera este voltaje como del primer nodo con respecto al segundo nodo en la descripción.

Si el circuito no especifica la condición inicial del elemento ni proporciona medios para obtenerla, se asume que su condición inicial es nula, o sea 0.

10.4 Respuestas

En el caso de la simulación en el dominio de la frecuencia, las respuestas son: los voltajes en los nodos, las caídas de voltaje en los elementos y las corrientes en los elementos. No se entregan las potencias consumidas por los elementos. Existe una diferencia: las respuestas serán expresiones simbólicas en función de la frecuencia compleja *s*.

10.5 Función de transferencia

Los problemas que nos solicitan encontrar la función de transferencia, H(s), son muy comunes cuando se estudia el análisis en el dominio de la frecuencia compleja. En el caso de que el problema sólo nos dé la red pasiva cuya función de transferencia se desee encontrar, debemos agregar nosotros una fuente a la entrada, preferiblemente una fuente de corriente de valor simbólico o unitario.

Veamos algunos ejemplos.

Problema N° 068

Planteamiento. Encuentre **H**(s) = V_{sal}/V_{ent} para la red de la figura, y localice todas sus frecuencias críticas.

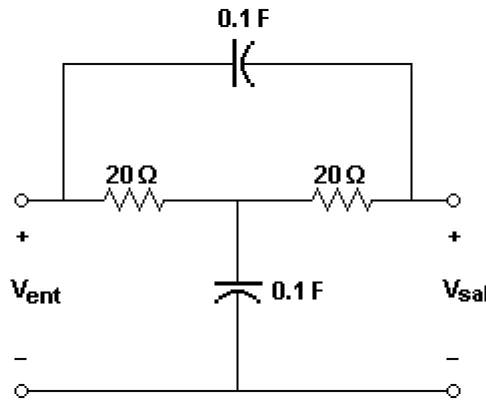

Figura 95. Circuito para el Problema N° 068.

Solución. Agregamos una fuente, y ordenamos la simulación. Nótese que empleamos cinco términos por elemento.

```
sq\fd("j1,0,1,1,0;r1,1,2,20,0;r2,2,3,20,0;ca,1,3,.1,0;cb,2,0,.1,0")
```

Con ¸ , se ejecuta la simulación. Solicitamos la función de transferencia `v3/v1` y obtenemos `(s^2+s+.25)/(s^2+1.5*s+.25)`. Para encontrar las frecuencias críticas, o sea los polos y ceros, usaremos un comando de la calculadora TI.

10.6 Polos, ceros y el comando `zeros`

La calculadora TI tiene un comando llamado zeros, el cual permite encontrar los ceros de un polinomio, para una determinada variable. Se utiliza así:

```
zeros(polinomio,variable)
```

La versión compleja de este comando es otro comando llamado `cZeros`, el cual se utiliza de la misma forma.

Podemos encontrar los ceros de una función de transferencia usando este comando, así: `zeros(función de transferencia)`. Podemos encontrar los polos de una función de transferencia usando este comando, así: `zeros(inverso de` `la función de transferencia)`. El inverso de la función de transferencia es 1 entre la función, o sino la función elevada a $^{-1}$.

Resolvamos la segunda pregunta del problema. Solicitamos los ceros así: `zeros(v3/v1,s)` y obtenemos `{-1/2}`. Aunque el comando no nos lo indica, este es un doble cero. Solicitamos los polos así: `zeros(v1/v3,s)` y obtenemos `{-1.31,-.191}`. Así, hemos terminado nuestro primer problema en dominio de la frecuencia.

<u>Problema N° 069</u>

Planteamiento. Encuentre **H**(s) = V_o/V_s para el circuito de la figura.

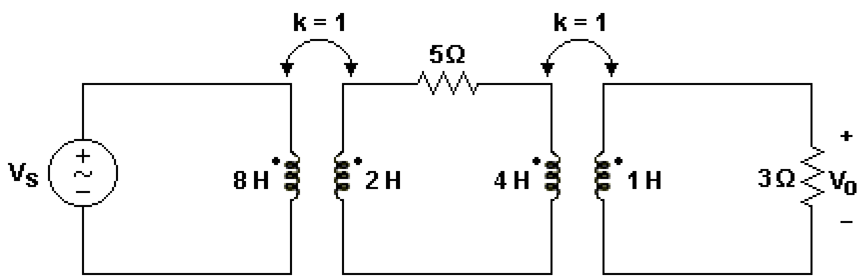

Figura 96. Circuito para el Problema N° 069.

Solución. No hay necesidad de agregar una fuente, pues el circuito la incluye. Ordenamos la simulación, usando cinco términos por elemento, y dando el valor de las inductancias mutuas en *M* en vez de *k*.

```
sq\fd("es,1,0,vs,0;l1,1,0,8,0;l2,2,0,2,0;r1,2,3,5,0;l3,3,0,4,0;l4,4,0,1,0;r2,4,0,3,0;m1,l1,l2,√(8*2),0;m2,l3,l4,√(4*1),0")
```

Con ¸ , se ejecuta la simulación. Solicitamos la función de transferencia `v4/v1` y obtenemos `3*s/(17*s+15)`. Veamos ahora un ejemplo de mayor grado de dificultad, pero antes aprendamos un nuevo concepto.

10.7 Exacto vs aproximado

En el punto 2.9 aprendimos que existen dos maneras en que la calculadora TI puede manejar un número: como exacto y como aproximado. Lo que determina cuál modo utilizará la calculadora para resolver un circuito es la manera en que se introducen los datos. Veamos cuáles son las ventajas y desventajas de cada uno de estos dos modos.

Modo Exacto:

- **Desventaja.** Generalmente, la solución de las ecuaciones toma más tiempo.
- **Ventaja.** Se pueden resolver de forma precisa algunos sistemas de ecuaciones que en modo aproximado podrían resultar irresolubles o resueltos de forma imprecisa.

Modo Aproximado:

- **Ventaja.** Generalmente, la solución de las ecuaciones toma menos tiempo.
- **Desventaja.** En algunos casos particulares, es posible que un sistema de ecuaciones no pueda ser resuelto en este modo, aunque en modo exacto si hubiese sido resuelto. En otros casos particulares, es posible que las respuestas pierdan precisión, especialmente en caso de números grandes.

En la gran mayoría de los casos, se puede usar cualquiera de los dos modos indistintamente. Sin embargo, cuando se trata de circuitos grandes, es preferible escoger el modo que mejor se ajuste a nuestra necesidad.

Problema N° 070

Planteamiento. Encuentre **H**(s) = V_{sal}/V_{ent} para la red de la figura. Los op-amps son ideales.

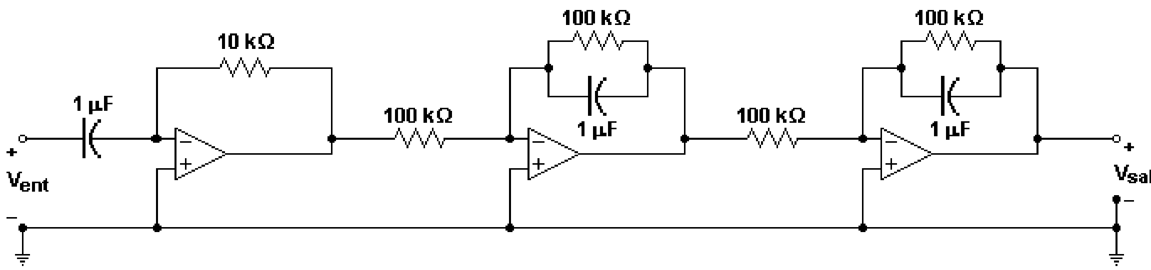

Figura 97. Circuito para el Problema N° 070.

Solución. Ordenamos la simulación, usando cinco términos por elemento. Preferimos emplear todos los valores en forma exacta, para garantizar la mejor probabilidad de obtener una solución, sin importar el tiempo de resolución.

```
sq\fd("e1,1,0,1,0;cc1,1,2,10^-6,0;r1,2,3,10^4,0;
o1,0,2,3,0;r2,3,4,10^5,0;r3,4,5,10^5,0;cc2,4,5,10^-6,0;
o2,0,4,5,0;r4,5,6,10^5,0;r5,6,7,10^5,0;cc3,6,7,10^-6,0;
o3,0,6,7,0")
```

Con ¸ , se ejecuta la simulación. Esta simulación toma más tiempo que cualquiera que hayamos visto hasta ahora. Este aumento en el tiempo no solamente se debe a que las ecuaciones generadas son enormes, sino a que se le aplica límite a todas las respuestas, pues existen op-amps en el circuito. El lector, al ordenar esta simulación, puede dejar la calculadora ejecutándola e irse a almorzar. Al regresar, encontrará que la simulación ha terminado. Solicitamos la función de transferencia `v7/v1` y obtenemos la expresión `-s/(s+10)^2`.

Aquí hay que hacer una anotación. Un simulador numérico, tal como SPICE, podría encontrar el valor numérico del voltaje de salida, si le damos un valor numérico para el voltaje de entrada y un valor numérico para la frecuencia. Pero para obtener una expresión simbólica para la función de transferencia, en términos de *s*, es necesario usar un simulador simbólico, como el *Symbulator*.

Veamos ahora cuatro ejemplos que ilustran que la herramienta `thevenin` también puede emplearse en el dominio de la frecuencia compleja.

Problema N° 071

Planteamiento. Encuentre: a) $\mathbf{Z}_{ent}$ para la red de la figura, en términos de *s*, b) $\mathbf{Z}_{ent}$(j8) en forma rectangular, c) $\mathbf{Z}_{ent}$(-2+j6) en forma polar, d) a qué valor debe cambiarse el resistor de 16Ω para que $\mathbf{Z}_{ent}$=0 en s= -5+j0, e) a qué valor debe cambiarse el resistor de 16Ω para que $\mathbf{Z}_{ent}$=: en s= -5+j0.

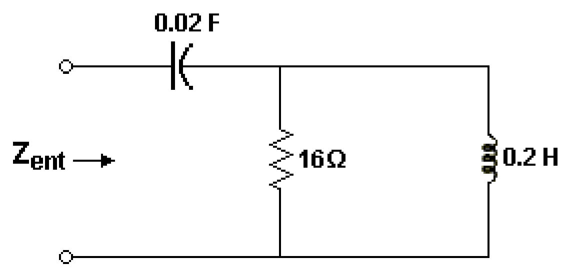

Figura 98. Circuito para el Problema N° 071.

Solución. Ejecutamos la herramienta, usando cinco términos por elemento.

```
sq\thevenin("c,1,2,.02,0;r,2,0,16,0;l,2,0,.2,0",1,0)
```

Presionamos ¸ . Escogemos `Passive` y `FD`. Presionamos ¸ . Respondemos la pregunta **a)** en este primer paso. Vemos en la pantalla el valor de zth. Salimos con ¸ . Podemos solicitar `zth` y obtenemos la expresión `16.*(s^2+3.125*s+250.)/(s*(s+80.))`.

Respondemos la pregunta **b)** así: `zth|s=8`***i*****ú**`Rect` y obtenemos `.158-4.67*`***i***.

Respondemos la pregunta **c)** así: `sq\absang(zth|s=-2+6`***i***`)` y obtenemos `6.85∠-114.°`.

Para responder las dos siguientes preguntas, volvemos a simular, usando un valor simbólico para el resistor.

```
sq\thevenin("c,1,2,.02,0;r,2,0,r,0;l,2,0,.2,0",1,0)
```

Con ¸ , se ejecuta la herramienta. Escogemos `Passive` y `FD`. Presionamos ¸ . Vemos en la pantalla el valor de zth. Salimos con ¸ . Respondemos la pregunta **d)** así: `cSolve(zth=0,r)|s=-5` y obtenemos `r=.909`.

Para responder la pregunta **e)**, requerimos dos pasos:

```
cSolve(zth=x,r)|s=-5
```

Obtenemos `r=(x+10.)/(x+11.)`. Ahora el segundo paso:

```
limit(ans(1),x,:)
```

Obtenemos `1`. Esta es la respuesta correcta.

<u>Problema N° 072</u>

Planteamiento. Este problema no tiene figura. Una red está compuesta por un capacitor de 1μF en paralelo con la combinación en serie de un inductor de 20mH y de un resistor de 500Ω. a) Encuentre $\mathbf{Z}_{ent}(s)$ para la red, en términos de *s*. b) Si s=σ-j0, encuentre el valor de σ, entre –20 y –10 kNp/s, en donde abs($\mathbf{Z}_{ent}$) es un mínimo. c) de nuevo con s=σ-j0, encuentre el valor de σ, menor que –30 kNp/s, en donde abs($\mathbf{Z}_{ent}$) es un máximo.

Solución. Ejecutamos la herramienta.

```
sq\thevenin("c,1,0,1E-6,0;l,1,2,.02,0;r,2,0,500,0",1,0)
```

Presionamos ¸ . Escogemos `Passive` y `FD`. Presionamos ¸ . Respondemos la pregunta **a)** en este primer paso. Vemos en la pantalla el valor de zth. Salimos con ¸ . De todas formas, lo solicitamos como `zth` y obtenemos la expresión:

```
1000000.*(s+25000.)/(s^2+25000.*s+50000000.)
```

Para responder las preguntas **b)** y **c)**, recordamos que tanto el mínimo como el máximo son puntos en los cuales la pendiente de la gráfica es 0. Usando la derivada, podemos resolver con `solve` para aquellas frecuencias en las cuales la derivada se hace 0.

```
solve(d(abs(zth),s)=0,s)
```

La *d* es la derivada, y se escribe con la combinación de teclas **2 =** (la **=** se encuentra arriba de la tecla del número 8. El comando significa: *"Encuentra los valores de s en los cuales la derivada del valor absoluto de zth con respecto a s, vale cero."* Obtenemos `s=-17929. or s=-32071.`. Según lo que nos dice el encabezado, el primer valor es un mínimo y el segundo es un máximo. Con esto, hemos respondido a las preguntas.

Problema N° 073

Planteamiento. Encuentre: a) $\mathbf{Z}_{ent}(\sigma)$ como una función de σ para la red de la figura, en términos de *s*, b) encuentre todos los ceros y polos, c) Grafique abs($\mathbf{Z}_{ent}(\sigma)$) vs σ, d) Grafique ang($\mathbf{Z}_{ent}(\sigma)$) vs σ.

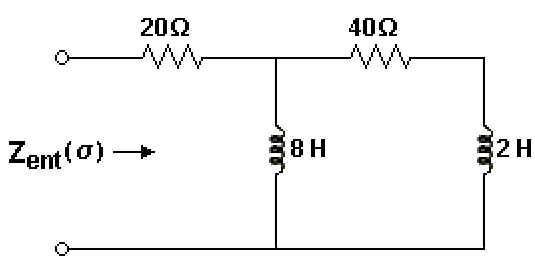

Figura 99. Circuito para el Problema N° 073.

Solución. Ejecutamos la herramienta.

```
sq\thevenin("r1,1,2,20,0;l1,2,0,8,0;r2,2,3,40,0;l2,3,0,
2,0",1,0)
```

Presionamos ¸ . Escogemos Passive y FD. Presionamos ¸ . Vemos en la pantalla el valor de zth, como función de *s*. Presionamos ¸ para salir. Solicitando: zth|s=σ!zth y obtenemos la expresión 4*(2*σ^2+65*σ+100)/(5*(σ+4)).

Respondamos la pregunta b). Pedimos los ceros con zeros(zth, σ) y obtenemos {5*(√(137)-13)/4,-5*(√(137)+13)/4}. Si preferimos ver estos ceros en forma aproximada, usamos ¥ ¸ y recibimos {-1.62,-30.9}. Pedimos los polos con zeros(1/zth, σ) y obtenemos {-4}.

Para hacer la gráfica que nos solicita la pregunta **c)**, seguimos los siguientes pasos.

abs(zth)|σ=x

ans(1)!y1(x)

Usamos ans(1) porque arriba tenemos la expresión. Presionamos ¥ „ . Aquí podemos especificar el rango que deseamos graficar. Usamos xmin=-50 y xmax=20. Presionamos ¥ …. La calculadora grafica la función, pero usando una escala vertical inapropiada. Presionamos „ j A en la TI-89 y „ A en la TI-92Plus, para aplicar el comando ZoomFit. Vemos la gráfica, en la escala correcta.

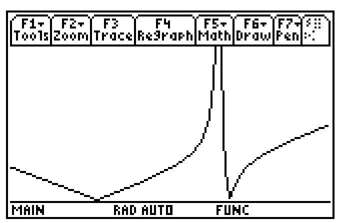

*Figura 100. Gráfica solicitada por la pregunta **c)** del Problema N° 073.*

Para hacer la gráfica que nos solicita la pregunta **d)**, seguimos los siguientes pasos.

angle(zth)| σ=x

`ans(1)!y1(x)`

Presionamos **¥ „** . Aquí podemos especificar el rango que deseamos graficar. Usamos xmin=0 y xmax=4.

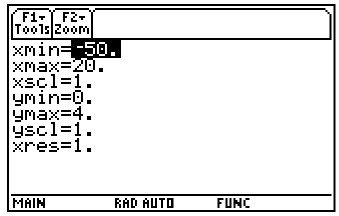

Figura 101. Especificando los límites de la gráfica.

Presionamos **¥ …**. Aparece la gráfica, con una escala inapropiada. Aplicamos el comando `ZoomFit`. Vemos la gráfica, en la escala correcta.

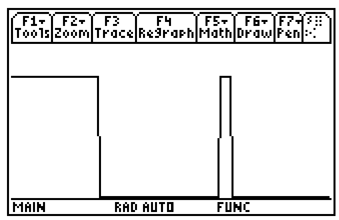

*Figura 102. Gráfica solicitada por la pregunta **d)** del Problema N° 073.*

Así, hemos respondido a todas las preguntas.

<u>Problema N° 074</u>

Planteamiento. Encuentre: a) $\mathbf{Z}_{ent}(s)$ para la red de la figura, b) encuentre todas las frecuencias críticas, c) Grafique abs($\mathbf{Z}_{ent}(s)$) vs s.

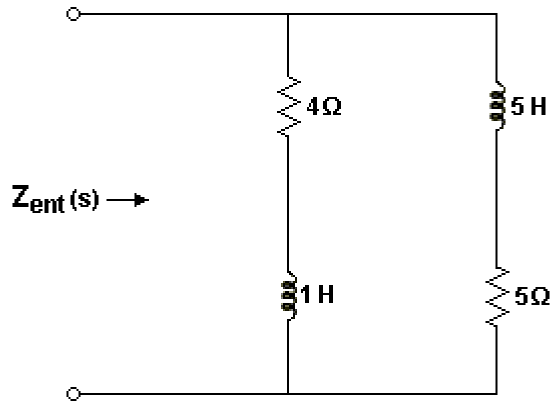

Figura 103. Circuito para el Problema N° 074.

Solución. Ejecutamos la herramienta.

```
sq\Thevenin("r1,1,2,4,0;l1,2,0,1,0;l2,1,3,5,0;r2,3,0,5,0",1,0):zth
```

Presionamos ¸ . Escogemos `Passive` y `FD`. Presionamos ¸ . Vemos en la pantalla el valor de zth, pero como función de *s*. Salimos con ¸ . En el *área de historia* vemos `5*(s+1)*(s+4)/(3*(2*s+3))`. Esta expresión apareció ahora, pues usamos `:zth` al final de la línea de comando.

Para la pregunta **b)**, pedimos los ceros con `cZeros(zth,s)` y obtenemos `{-4,-1}`. Pedimos los polos con `cZeros(1/zth,s)` y obtenemos `{-3/2,-:,:}`. Usamos el comando `cZeros` en vez del comando `zeros`, porque a diferencia del problema anterior, esta vez estamos buscando polos en s, los cuales pueden ser complejos. En el caso de los polos para σ, como la parte imaginaria es nula, los polos son enteros. Así, usamos el comando `cZeros` cuando vamos a encontrar frecuencias críticas complejas, y el comando `zeros` cuando vamos a encontrar frecuencias críticas reales.

Para hacer la gráfica que nos solicita la pregunta **c)**, seguimos los siguientes pasos.

```
abs(zth)|s=x
```

`ans(1)!y1(x)`

Usamos `ans(1)` porque arriba tenemos la expresión. Presionamos **¥ „** . Aquí podemos especificar el rango que deseamos graficar. Usamos xmin=-7 y xmax=2. Presionamos **¥ …**. La calculadora grafica la función, pero usando una escala vertical inapropiada. Presionamos **„ j** A en la TI-89 y **„** A en la TI-92Plus, para aplicar el comando `ZoomFit`. Vemos la gráfica, en la escala correcta.

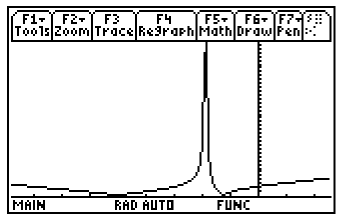

*Figura 104. Gráfica solicitada por la pregunta **c)** del Problema N° 074.*

Así, hemos respondido a todas las preguntas.

Capítulo

11

Simulación en el dominio del tiempo

11.1 Relación entre FD y TR

En este capítulo, veremos las técnicas de simulación para análisis en el dominio del tiempo, o sea análisis transitorio, en el *Symbulator*. Existe una íntima relación entre el análisis en el dominio de la frecuencia FD y el análisis en el dominio del tiempo TR. En TR, el *Symbulator* utiliza las técnicas de transformada de Laplace para obtener las respuestas completas en el dominio del tiempo, a partir de las respuestas en dominio de la frecuencia obtenidas con FD. La respuesta completa incluye la respuesta forzada y la transitoria. Como hemos aprendido en el salón de clases, la transformada de Laplace puede llevar una respuesta del dominio de la frecuencia al dominio del tiempo.

11.1.1 DiffEq

El *Symbulator* utiliza el poderoso programa *DiffEq*, escrito por Lars Frederiksen, quien es el mejor programador de calculadoras que he conocido, y mi amigo personal. *DiffEq* es un grupo de programas hechos para resolver ecuaciones diferenciales. La herramienta más avanzada para transformada de Laplace para calculadoras TI se llama *Advanced Laplace*, hecha también por Lars Frederiksen. Aunque menos poderosa, la transformada de Laplace del *DiffEq* es más rápida que la del *Advanced Laplace*. Por ello el *Symbulator* prefiere al *DiffEq*.

El *Symbulator* utiliza la función `laplace` del *DiffEq* al inicio de cualquier simulación FD o TR, y la función `ilaplace` del mismo al final de cualquier simulación TR. Por ello es necesario tener este programa instalado cuando se use el *Symbulator* para realizar estos dos tipos de análisis.

11.2 Dos maneras de obtener respuestas en el dominio del tiempo

Existen dos maneras para que el usuario del *Symbulator* obtenga respuestas en el dominio del tiempo:

1) La primera alternativa *(véase el punto 11.3)* es realizar un análisis en el dominio de la frecuencia, y luego convertir las respuestas que nos interesen al dominio del tiempo. Este análisis se realiza usando la puerta **sq\fd**. Este análisis resuelve el circuito en el dominio de la frecuencia, convirtiendo todos los valores de las fuentes que estén en el dominio del tiempo a su equivalente en el dominio de la frecuencia, pero no convierte las respuestas al dominio del tiempo. Nos entregará todas las respuestas en el dominio de la frecuencia. Como no se realiza una conversión automática de todas las respuestas al dominio del tiempo, se ahorra todo el tiempo que estas conversiones hubiesen consumido. Así, la simulación tardará el mínimo posible. Tras el análisis, el usuario puede llevar al dominio del tiempo aquellas respuestas que le interesen. Esta alternativa debe emplearse en dos casos: a) cuando deseamos respuestas tanto en el dominio del tiempo como en el dominio de la frecuencia, y b) cuando se desea una simulación lo más rápida posible, y deseamos solamente algunas respuestas en el dominio del tiempo, y no todo el conjunto.

2) La segunda alternativa *(véase del punto 11.5 en adelante)* es realizar un análisis transitorio. Este análisis se realiza usando la puerta **sq\tr**, cuyo uso explicaremos más adelante. Este análisis resuelve el circuito en el dominio de

la frecuencia, convirtiendo todos los valores de las fuentes que estén en el dominio del tiempo a su equivalente en el dominio de la frecuencia, y finalmente convierte todas las respuestas de vuelta al dominio del tiempo. Este análisis nos entregará todas las respuestas en el dominio del tiempo. El usuario nunca verá las respuestas en el dominio de la frecuencia. La conversión de todas las respuestas al dominio del tiempo puede consumir varios minutos, tanto más tiempo cuanto mayor sea el circuito. Por ello, esta alternativa debe emplearse únicamente en tres casos: a) cuando el circuito sea pequeño, b) cuando el usuario esté interesado en todas las respuestas en el dominio del tiempo y no sólo en algunas respuestas, y c) cuando al usuario no le importe que la simulación tarde algunos minutos.

11.3 Dos maneras de convertir respuestas al dominio del tiempo

Cuando usemos la primera alternativa, tendremos que convertir nosotros las respuestas que nos interesen al dominio del tiempo. Podemos hacerlo de dos maneras.

11.3.1 Función `ilaplace` *del DiffEq*

El *DiffEq* convierte una función del dominio de la frecuencia al dominio del tiempo, usando la función `ilaplace`, así:

```
dif\ilaplace(función de la frecuencia,s)
```

Vale la pena señalar que el *DiffEq* también hace la operación inversa, es decir, convertir una función del dominio del tiempo al dominio de la frecuencia, usando la función `laplace`, así:

```
dif\laplace(función del tiempo,t)
```

11.3.2 Herramienta `fd_to_tr` *del Symbulator*

La herramienta `fd_to_tr` del *Symbulator* es realmente una especie de atajo para el uso de la función `ilaplace` del *DiffEq*. La única función de esta herramienta es ahorrar al usuario la colocación de la `",s)"` al final de la línea de comando. Es la función `ilaplace` la que hace todo el trabajo. Esta herramienta se usa así:

```
sq\fd_to_tr(respuesta en dominio de la frecuencia)
```

Nos entrega la respuesta en el dominio del tiempo.

11.4 Menú del Symbulator

El *Symbulator* incluye un programa para crear un menú personalizado, llamado en inglés un *Custom Menu*. El programa se ejecuta así:

```
sq\makemenu()
```

Al ejecutarse, este programa creará un menú personalizado, en el cual pueden encontrarse todas las puertas, herramientas y elementos del *Symbulator*. Así, el usuario evitará tener que escribir los nombres de las herramientas y puertas, pues podrá seleccionarlas usando las flechas del cursor.

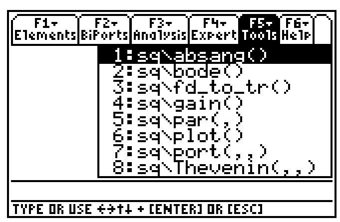

Figura 105. Menú personalizado del Symbulator.

Para información sobre cómo restaurar el menú original de la calculadora, léase el *Manual de Usuario*.

11.5 Puerta para el dominio del tiempo: `tr`

Si escogemos la segunda alternativa, es decir realizar una simulación en el dominio del tiempo *t*, usaremos la *puerta* llamada **sq\tr**.

11.6 Dato de entrada

El único dato de entrada que hay que proporcionar a esta puerta es la descripción del circuito, usando la misma notación que se usa para una simulación en el dominio de la frecuencia. La simulación en dominio del tiempo se ordena así: `sq\tr(circuito)`.

11.7 Condiciones iniciales

Al igual que la simulación en el dominio de la frecuencia, el quinto término en la descripción de un inductor o un capacitor es su condición inicial.

11.8 Respuestas

En el caso de la simulación en dominio del tiempo, las respuestas son: los voltajes en los nodos, las caídas de voltaje en los elementos y las corrientes en los elementos. No se entregan las potencias consumidas por los elementos. Como ya explicamos en el punto 11.2, todas estas respuestas serán entregadas en el dominio del tiempo. Por lo tanto, serán expresiones simbólicas en función del tiempo *t*.

Veamos ahora varios problemas que ilustran cómo se resuelven en el *Symbulator* los problemas de análisis transitorio.

Problema N° 075

Planteamiento. a) Según la figura, encuentre la razón $I_L(s)/V_s(s)$. b) Haga $v_s(t)$=100u(t) V y encuentre $i_L(t)$.

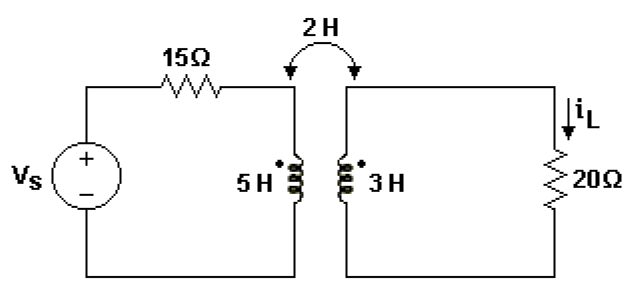

Figura 106. Circuito para el Problema N° 075.

Solución. Como el problema no nos especifica condiciones iniciales, ni tampoco nos da la manera de calcularlas, asumimos que son nulas. Este problema tiene dos partes. La primera nos solicita una respuesta en dominio de la frecuencia, mientras que la segunda nos solicita una respuesta en el dominio del tiempo. Existen dos buenas razones para resolver este problema utilizando la primera alternativa. (Como explicamos en el punto 11.2, la primera alternativa consiste en simular con `sq\fd` y luego transformar con `sq\ft_to_tr`.) La primera razón es que sólo nos interesa una respuesta en el dominio del tiempo. La segunda razón es que además nos interesa una respuesta en el dominio de la frecuencia. Como explicamos en el punto 11.2, ambas razones son causa suficiente para usar la primera alternativa en vez de la segunda. A continuación, el procedimiento para resolver este problema.

Para efectos de responder la primera pregunta, no importa qué valor asignemos a la fuente de voltaje. Podríamos definir su valor como `vs`, o como `100u(t)`, y la respuesta sería la misma. Esto se debe a que la respuesta es una relación, y la fuente influirá tanto en el numerador como en el denominador, cancelándose su efecto. Ahora bien, si al principio utilizamos un valor como `vs`, cuando vayamos a responder la segunda pregunta, tendremos que volver a simular utilizando el nuevo valor de `100u(t)`. Sin embargo, si al principio utilizamos el valor de `100u(t)`, esta única simulación nos servirá para responder ambas preguntas. Así, aprovechamos el hecho de que la primera pregunta nos lo permite, y usamos el valor de la fuente como `100u(t)`. Es necesario decir que la función `ilaplace` del programa *DiffEq* entiende lo que significan `u(t)` y `δ(t)`. De hecho, esta función asume que cualquier valor numérico introducido está multiplicado por `u(t)`. Por ello, no es necesario introducir el valor `100u(t)`, pues para `ilaplace` decir `100` y decir `100u(t)` es lo mismo. A continuación, la descripción:

```
sq\fd("e1,1,0,100,0;r1,1,2,15,0;r2,3,0,20,0;l1,2,0,5,0;
l2,3,0,3,0;m1,l1,l2,2,0")
```

Con ¸, se ejecuta la simulación. Para responder la primera pregunta, solicitamos la relación `ir2/v1` y obtenemos `2*s/(11*s^2+145*s+300)`. Para responder la segunda pregunta, usamos `sq\fd_to_tr(ir2)` y obtenemos `40*√(313)*`*e*`^((5*√(313)/22-145/22)*t)/313-40*√(313)*`*e*`^((-5*√(313)/22-145/22)*t)/313`. Estas son las respuestas correctas. ¿Desea el lector ver las respuestas en forma aproximada? Si solicitamos `approx(ans(1))`, obtendremos `2.26*(.0765)^t-2.26*(2.46E-5)^t` (redondeando). Si solicitamos `expand(ans(1))` , obtendremos `2.26/(`*e*`^t)^(2.57)-2.26/(`*e*`^t)^(10.6)` (redondeando). Aprendamos más sobre este comportamiento de las respuestas aproximadas.

11.9 Presentación aproximada con exponencial

Cuando el modo Exact/Approx está en AUTO, la calculadora TI mantendrá una expresión en la forma *e*`^(`*n*`t)` sólo si *n* es un número exacto. Por ejemplo, si introducimos *e*`^(2t/3)` en la *línea de entrada*, veremos que en el *área de historia* la calculadora nos presenta también *e*`^(2t/3)`. Esto se debe a que el número que acompaña a t, es decir `2/3`, es un número exacto.

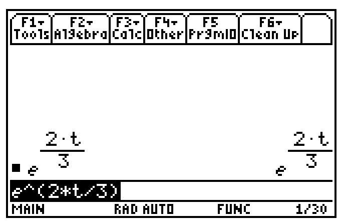

Figura 107. La exponencial se mantiene como tal.

La desventaja de esta manera exacta de presentar la respuesta reside en que en algunas ocasiones, como en el caso del problema que acabamos de resolver, la expresión exacta es tan voluminosa, que resulta incómodo su uso. Además, algunos libros de texto y profesores favorecen las respuestas en presentación aproximada.

Si *n* es un número aproximado, la calculadora convertirá la expresión a un valor totalmente aproximado, al punto de eliminar las exponenciales. Por ejemplo, usemos la expresión en forma aproximada, ya sea introduciendo *e*`^(2t/3.)` o `approx(`*e*`^(2t/3))` en la *línea de entrada*, (pues tanto el punto como el comando `approx` tienen el efecto de convertir la expresión en aproximada). Veremos que en el *área de historia* la calculadora nos presenta ahora `(1.94773404)^t`. Esto se debe a que el número que acompaña a t, es decir `2/3.`, es un número aproximado.

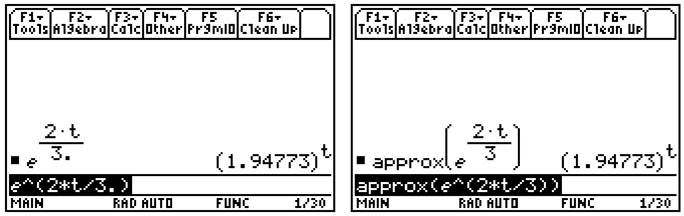

Figura 108. La exponencial se convierte a un aproximado.

La desventaja de esta manera totalmente aproximada de presentar la respuesta reside en que los exponenciales han sido reemplazados, diluídos en el número base aproximado. En los años que llevo estudiando ingeniería eléctrica, jamás he visto un libro de texto o un profesor que favorezca el uso de las respuestas en esta presentación aproximada sin exponenciales.

Por lo tanto, es deseable obtener una combinación de las virtudes de ambas maneras: una expresión aproximada que sea más pequeña que la exacta, pero que conserve las exponenciales en su lugar. En la calculadora TI, es posible obtener este tipo de respuesta, utilizando el comando `expand`. Este comando se utiliza así: `expand(expresión,variable)`, y su resultado es que la expresión será mostrada de manera expandida, con respecto a la variable. En nuestro ejemplo, si introducimos `expand(`*e*`^(2t/3.),t)` o `expand(approx(`*e*`^(2t/3)),t)` en la *línea de entrada*, veremos que en el *área de historia* la calculadora nos presenta `(`*e*`^t)^.666666667`.

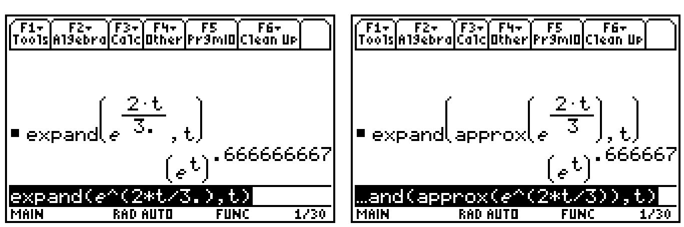

Figura 109. La expresión se expande.

También en este caso, nosostros preferiríamos una presentación algo diferente, así como la siguiente: $e^{\cdot 666666667t}$. Aún así, esa presentación es mucho mejor que las dos anteriores.

Si expandimos una expresión exponencial cuyo número *n* es negativo, el comando `expand` mostrará un comportamiento interesante. Por ejemplo, con `expand(e^(-2t/3.),t)` en la *línea de entrada*, veremos que en el *área de historia* la calculadora nos presenta `1/(e^t)^.666666667`.

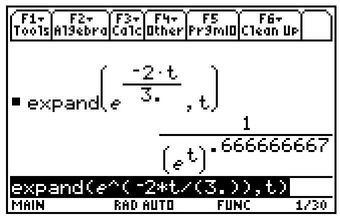

Figura 110. La exponencial aparece en el denominador.

Ciertamente, nosostros preferiríamos una presentación diferente, así como la siguiente: e^$^{-.666666667t}$. Pero teniendo la primera expresión, podemos obtener la segunda fácilmente.

<u>Problema N° 076</u>

Planteamiento. a) Para el circuito que se muestra en la figura, encuentre $H(s)=v_{C2}/v_S$. b) Sea $v_{C1}(0^+)=0$ y $v_{C2}(0^+)=0$. Encuentre $v_{C2}(t)$ si $v_S(t)=u(t)$.

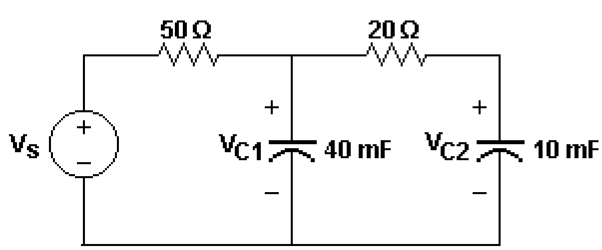

Figura 111. Circuito para el Problema N° 076.

Solución. El problema nos especifica claramente que las condiciones iniciales son nulas. Como este problema es idéntico al anterior, nos ahorramos las explicaciones, pues resultarían redundantes. Sólo diremos que, para el *DiffEq*, y por ende para el *Symbulator*, `u(t)` es lo mismo que 1.

```
sq\fd("es,1,0,1,0;r1,1,2,50,0;r2,2,3,20,0;cc1,2,0,.04,0;cc2,3,0,.01,0")
```

Con ¸ , se ejecuta la simulación. Para responder la primera pregunta, solicitamos la relación `v3/v1` y obtenemos `2.5/(s^2+6.75*s+2.5)`. Para responder la segunda pregunta, usamos `expand(sq\fd_to_tr(v3),t)` y obtenemos `-1.066/(e^t)^(.393)+.0659/(e^t)^(6.36)+1.` (redondeando).

En vez de haber usado `expand(sq\fd_to_tr(v3),t)`, pudimos haber usado primero `sq\fd_to_tr(v3)` y luego `expand(ans(1),t)`, y el resultado sería el mismo.

11.10 Intervalos y simulaciones

En el *Symbulator*, **cada intervalo de tiempo requiere una simulación**. Para simular el comportamiento de un circuito que tiene un interruptor, el cual conmuta en un tiempo dado, digamos t=0, debemos hacer dos simulaciones: una para t<0 y otra para t>0.

El mismo principio rige para la función u(t). Matemáticamente, *n**u(t) indica que para tiempos <t la función vale 0, y para tiempos >t la función vale *n*. De la misma forma, matemáticamente, *n**u(-t) indica que para tiempos <t la función vale *n*, y para tiempos >t la función vale 0. Para efectos de la simulación, la u(t) no tiene ese significado. Debemos realizar una simulación para cada intervalo de tiempo.

Veamos varios ejemplos.

Problema N° 077

Planteamiento. Después de permanecer cerrado por mucho tiempo, el interruptor en el circuito de la figura se abre en t=0. a) Encuentre $i_L(t)$ para t>0. b) Evalúe $i_L(10ms)$. c) Encuentre t_1 si $i_L(t_1)= 0.5i_L(0)$.

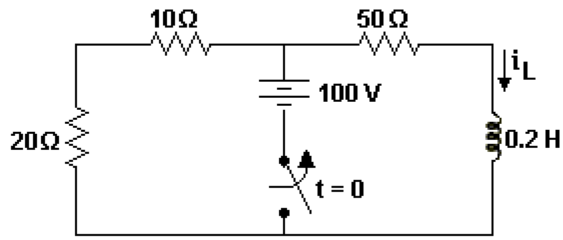

Figura 112. Circuito para el Problema N° 077.

Solución. El problema nos indica que hay un interruptor que conmuta en t=0. Debemos realizar primero una simulación, en corriente directa, para t<0, la cual nos permitirá encontrar la condición inicial del inductor. Luego realizaremos una segunda simulación, en dominio de la frecuencia, para t>0, para responder las preguntas. El procedimiento es el siguiente:

```
sq\dc("r1,1,0,20;r2,2,1,10;r3,2,3,50;l1,3,0,.2;e1,2,0,100"):il1
```

Solicitamos la simulación en corriente directa, y seguidamente pedimos la corriente en el inductor. Con ¸ , se ejecuta la simulación. Obtenemos `2` como corriente en el inductor. Esta será su condición inicial para la simulación del siguiente intervalo.

```
sq\fd("r1,1,0,20,0;r2,2,1,10,0;r3,2,3,50,0;l1,3,0,.2,2"):sq\fd_to_tr(il1)
```

Solicitamos la simulación en dominio de la frecuencia, y seguidamente pedimos la corriente en el inductor, en dominio del tiempo. Con ¸ , se ejecuta la simulación. Obtenemos `2*`*e*`^(-400*t)` como expresión para la corriente en el inductor. Esto responde la pregunta **a)**.

Para responder la pregunta **b)**, usamos `ans(1)|t=10E-3` y obtenemos `.0366` (redondeando).

Para responder la pregunta **c)**, usamos `solve(ans(2)=.5*2,t)`. Nótese que la orden quiere decir: *"Dime cuánto vale el tiempo cuando se cumple que la expresión*

`2*e^(-400*t),` *invocada por ans(2), es igual a 0.5 la corriente inicial, la cual es 2."* Obtenemos `t=.00173` (redondeando).

En resumen, este problema no nos entregó las condiciones iniciales, pero nos entregó una manera para calcularlas mediante una simulación en corriente directa para antes de que conmutara el interruptor. Este problema requirió dos simulaciones, una para cada intervalo de tiempo.

Problema N° 078

Planteamiento. Si $i_L(0)$=10 A en el circuito de la figura, encuentre $i_L(t)$ para t>0.

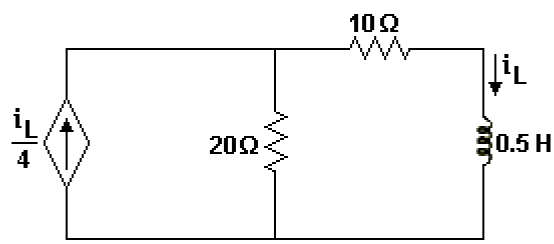

Figura 113. Circuito para el Problema N° 078.

Solución. El problema nos entrega la condición inicial del inductor. Por lo tanto, sólo necesitamos realizar una simulación:

```
sq\fd("j1,0,1,il/4,0;r1,1,0,20,0;r2,1,2,10,0;l,2,0,.5,10"):sq\fd_to_tr(il)
```

Solicitamos la simulación en dominio de la frecuencia, y seguidamente pedimos la corriente en el inductor en dominio del tiempo. Con ¸ , se ejecuta la simulación. Obtenemos `10*e^(-50*t)` como expresión para la corriente en el inductor.

Este problema nos entregó las condiciones iniciales. Sólo requirió una simulación.

Problema N° 079

Planteamiento. a) Encuentre $v_C(t)$ para t>0. b) ¿En qué tiempo v_C= 0.1$v_C(0)$?

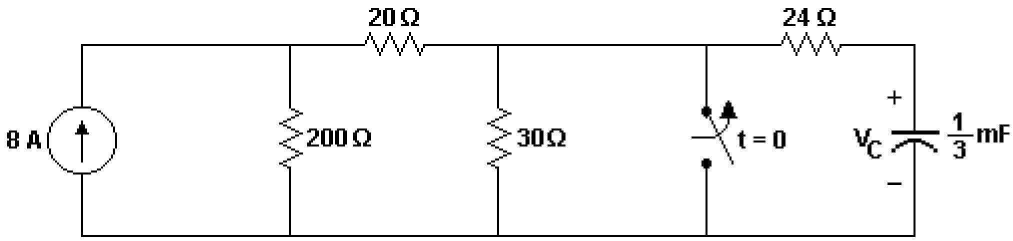

Figura 114. Circuito para el Problema N° 079.

Solución. Este problema es similar al Problema N° 076. Preferimos usar valores exactos.

```
sq\dc("j1,0,1,8;r1,1,0,200;r2,1,2,20;r3,2,0,30;r4,2,3,24;c,3,0,(1/3)/1000"):vc
```

Simulamos en corriente directa, y pedimos el voltaje en el capacitor. Presionamos ¸ . Obtenemos `192` como voltaje en el capacitor. Esta será su condición inicial para la simulación del siguiente intervalo. Simulamos en dominio de la frecuencia, y pedimos el voltaje en el capacitor, en dominio del tiempo.

```
sq\fd("j1,0,1,8,0;r1,1,0,200,0;r2,1,2,20,0;r3,2,0,30,0;r4,2,3,24,0;c,3,0,(1/3)/1000,192;s1,2,0,0,0"):sq\fd_to_tr(vc)
```

Nótese que para simular el interruptor cerrado, usamos un cortocircuito. Podríamos haber eliminado la parte del circuito que está a la izquierda del interruptor, pues no nos interesa. Así ahorraríamos trabajo al simulador. Pero seremos rigurosos, y la dejaremos. Presionamos ¸ . Obtenemos `192*`*e*`^(-125*t)`. Esto responde la pregunta **a)**.

Para responder la pregunta **b)**, usamos `solve(ans(1)=.1*ans(2),t)`. La orden quiere decir: *"Dime cuánto vale el tiempo cuando se cumple que la expresión `192*`e`^(-125*t)`, invocada por `ans(1)`, es igual a 0.1 el voltaje inicial, invocado por `ans(2)`"*. Obtenemos `t=.01842` (redondeando).

Este problema no nos entregó las condiciones iniciales, pero nos entregó una manera para calcularlas, mediante una simulación en corriente directa del intervalo anterior a la conmutación del interruptor. Este problema requirió dos simulaciones, una para cada intervalo de tiempo.

Problema N° 080

Planteamiento. Suponga que el circuito que se muestra en la figura se encuentra en esa forma desde hace mucho tiempo. Encuentre v_C(t) para t>0.

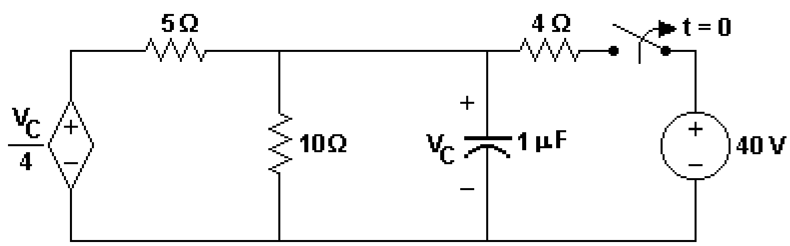

Figura 115. Circuito para el Problema N° 080.

Solución. Veamos.

```
sq\dc("e1,1,0,vc/4;r1,1,2,5;r2,2,0,10;c,2,0,10^-6;
r3,2,3,4;e2,3,0,40"):vc
```

Preferimos usar valores exactos. Recuérdese que el signo de `10^-6` es el de negativo, no el de resta. Presionamos ¸ . Obtenemos `20` como voltaje en el capacitor. Esta será su condición inicial para la simulación del siguiente intervalo.

```
sq\fd("e1,1,0,vc/4,0;r1,1,2,5,0;r2,2,0,10,0;c,2,0,10^-
6,20"):sq\fd_to_tr(vc)
```

Simulamos en dominio de la frecuencia, y pedimos el voltaje en el capacitor, en dominio del tiempo. Presionamos ¸ . Obtenemos `20*`*e*`^(-250000*t)`.

Este problema no nos entregó las condiciones iniciales, pero nos entregó una manera para calcularlas. Este problema requirió dos simulaciones, una para cada intervalo de tiempo.

<u>Problema N° 081</u>

Planteamiento. Suponga que el op-amp es ideal. Encuentre $v_o(t)$.

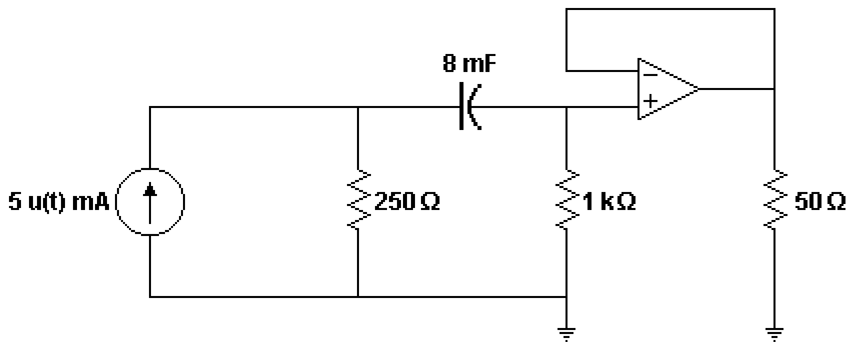

Figura 116. Circuito para el Problema N° 081.

Solución. El problema nos da a entender que no hay condiciones iniciales. Por ello, realizaremos sólo una simulación. Simulamos en dominio de la frecuencia, y pedimos el voltaje en el capacitor, en dominio del tiempo. Preferimos usar valores exactos.

```
sq\fd("j1,0,1,5/1000,0;r1,1,0,250,0;ca,1,2,8/1000,0;r2,
2,0,1000,0;o1,2,3,3,0;r3,3,0,50,0"):sq\fd_to_tr(v3)
```

Presionamos ¸ . Obtenemos *e*`^(-t/10)`.

Este problema usó condiciones iniciales nulas. Sólo requirió una simulación.

<u>Problema N° 082</u>

Planteamiento. Suponga que el op-amp es ideal. Encuentre $v_o(t)$.

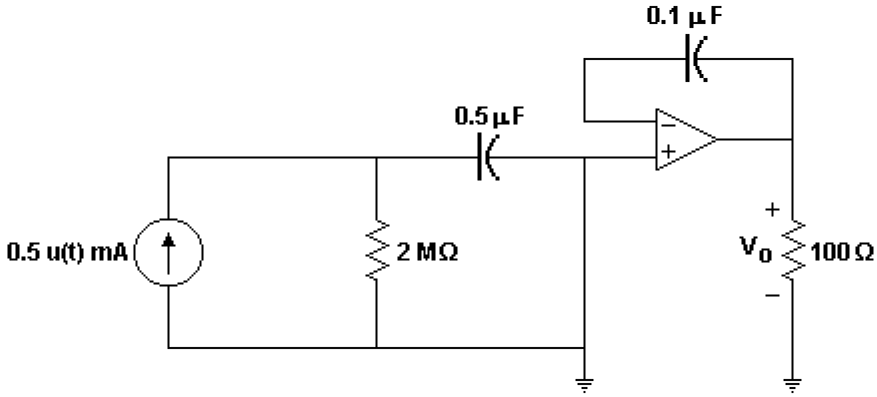

Figura 117. Circuito para el Problema N° 082.

Solución. Este problema es igual al anterior. Preferimos usar valores aproximados.

```
sq\fd("j1,0,1,.5E-6,0;r1,1,0,2E6,0;ca,1,2,.5E-6,0;
o1,0,2,3,0;cb,2,3,.1E-6,0;r2,3,0,100,0"):sq\fd_to_tr(v3)
```

Presionamos ¸ . Obtenemos `5.*`*e*`^(-t)-5.`

Este problema usó condiciones iniciales nulas. Sólo requirió una simulación.

Problema N° 083

Planteamiento. En el circuito mostrado, sea L=5H, R=8Ω, C=12.5mF y $v(0^+)$=40V. Encuentre a) *v(t)* si $i(0^+)$=8 A, b) *i(t)* si $i_C(0^+)$=8 A.

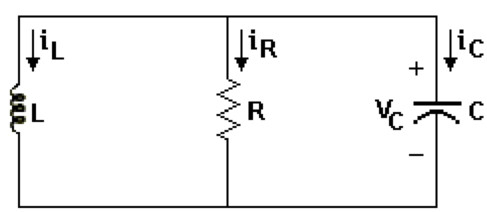

Figura 118. Circuito para el Problema N° 083.

Solución. El problema nos da la condición inicial del capacitor. Luego nos hace dos preguntas. Para la primera pregunta, nos indica también la condición inicial del inductor. Simulamos en dominio de la frecuencia, y pedimos el voltaje en dominio del tiempo.

```
sq\fd("l,1,0,5,8;r,1,0,8,0;c,1,0,12.5E-3,40"):sq\fd_to_tr(v1)
```

Presionamos ¸ . Obtenemos 160.*e^(-8*t)-120.*e^(-2*t). Esto responde la pregunta **a)**.

Para la segunda pregunta, el problema no nos indica la condición inicial del inductor, pero nos da la manera de encontrarla. Simulamos en dominio de la frecuencia, y pedimos el voltaje en dominio del tiempo. Usamos un valor simbólico io como condición inicial para el inductor, ya que no conocemos su valor numérico. (La condición inicial del inductor que nos entregó el problema anteriormente era válida sólo para la pregunta **a**).

```
sq\fd("l,1,0,5,io;r,1,0,8,0;c,1,0,12.5E-3,40")
```

Presionamos ¸ . Sabemos que la corriente en el capacitor vale 8 en t=0. Pedimos sq\fd_to_tr(ic) y obtenemos una expresión simbólica en términos de io. Debemos resolver esta expresión para obtener el valor numérico de io. La resolvemos así: solve(ans(1)=8,io)|t=0. Lo que estamos pidiendo es: *"Dime cuánto vale* io *si la corriente en el capacitor, cuya expresión se encuentra en* ans(1)*, en el tiempo cero vale 8."* Esto es una manera de preguntar: *¿Cuánto tiene que ser la condición inicial del inductor para que la corriente en el capacitor en tiempo cero sea 8?* Obtenemos: io=-13.. Esa es la condición inicial del capacitor. Ahora debemos encontrar la expresión para la corriente en el inductor. Podríamos simular nuevamente, usando la condición inicial numérica. Pero lo más sencillo es simplemente reemplazar en la expresión el valor que acabamos de encontrar con solve. Pedimos sq\fd_to_tr(il)|ans(1) y obtenemos 3.*e^(-8*t)-16.*e^(-2*t). Esto responde la pregunta **b)**.

Problema N° 084

Planteamiento. Encuentre $i_L(t)$ para t>0 en el circuito que se muestra en la figura.

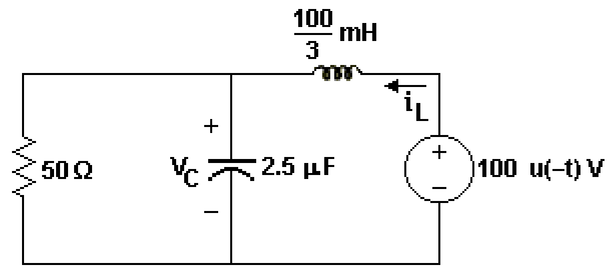

Figura 119. Circuito para el Problema N° 084.

Solución. Debemos entender la u(-t) como un interruptor: la fuente vale 100 para t<0, y vale 0 para t>0. Por lo tanto, usaremos el concepto de dos intervalos.

```
sq\dc("r,1,0,50;c,1,0,2.5E-6;l,1,2,(100/3)E-3;
e,2,0,100")
```

Recuérdese que el signo que aparece en `E-6` es el de negativo, no el de resta. Presionamos ¸ . Obtengamos las condiciones iniciales. Para `vc` obtenemos `100`. Para `il` obtenemos `-2`. Simulamos en dominio de la frecuencia, y pedimos la corriente en el inductor, en dominio del tiempo.

```
sq\fd("r,1,0,50,0;c,1,0,2.5E-6,100;l,1,2,(100/3)E-3,-2;
e,2,0,0,0"):expand(sq\fd_to_tr(il))
```

Nótese que a la fuente se le dió un valor de 0. Había otras dos alternativas: usar un cortocircuito, o eliminar la fuente y conectar el inductor al nodo de referencia. Presionamos ¸ . Obtenemos `.25/(`*e*`^t)^6000-2.25/(`*e*`^t)^(2000.)`. Esto es lo mismo que decir $.25e^{-6000\,t} - 2.25e^{-2000\,t}$

<u>Problema N° 085</u>

Planteamiento. Encuentre $i_s(t)$ para t>0 en el circuito de la figura si $v_s(t)$ es a) 10*u*(-t) V, b) 10*u*(t) V.

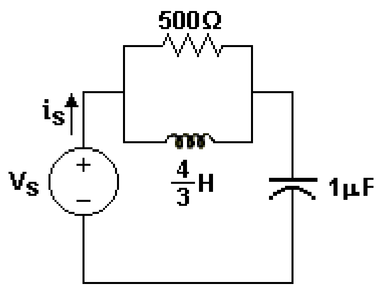

Figura 120. Circuito para el Problema N° 085.

Solución. Respondamos la pregunta **a)**. Debemos entender la u(-t) como un interruptor que hace que la fuente valga 10 para t<0, y 0 para t>0. Por lo tanto, usaremos el concepto de dos intervalos.

```
sq\dc("e,1,0,10;r,1,2,500;l,1,2,4/3;c,2,0,1/10^6")
```

Presionamos ¸ . Obtengamos las condiciones iniciales. Para `vc` obtenemos `10`. Para `il` obtenemos `0`.

```
sq\fd("e,1,0,0,0;r,1,2,500,0;l,1,2,4/3,0;c,2,0,1/10^6,1
0"):sq\fd_to_tr(-ie)
```

Usamos el negativo en *–ie* porque *is* está definida en sentido contrario a *ie*. Presionamos ¸ . Obtenemos `e^(-500*t)/400-9*e^(-1500*t)/400`. Si queremos ver la respuesta en forma aproximada, en vez de `sq\fd_to_tr(-ie)`, usamos `expand(approx(sq\fd_to_tr(-ie)))`. Obtenemos la expresión `.0025/(e^t)^500-.0225/(e^t)^1500`. Esto es lo mismo que decir $.0025e^{-500t}-.0225e^{-1500t}$. Esto responde la pregunta **a)**.

Respondamos la pregunta **b)**. Debemos entender la u(t) como un interruptor que hace que la fuente valga 0 para t<0, y 10 para t>0. Por lo tanto, sólo necesitaremos un intervalo, pues la fuente está muerta antes de t. Usamos condiciones iniciales nulas.

```
sq\fd("e,1,0,10,0;r,1,2,500,0;l,1,2,4/3,0;c,2,0,1/10^6,
0"):sq\fd_to_tr(-ie)
```

Presionamos ¸ . Obtenemos `9*e^(-1500*t)/400-e^(-500*t)/400`. Si queremos ver la respuesta en forma aproximada, usamos el mismo procedimiento explicado arriba. Obtendríamos $.0225e^{-1500\,t}$ - $.0025e^{-500\,t}$. Esto responde la pregunta **b)**.

11.11 El Modo (Impala)

Al igual que el *Modo (Experto)*, el *Modo (Impala)* es una variante en la manera en que el *Symbulator* resuelve un problema. Pero a diferencia del *(Experto)* que debe ser activado por el usuario a voluntad, el *(Impala)* se activa automáticamente cuando es necesario.

El *Modo (Impala)* se activa cuando el usuario introduce alguna fuente cuyo valor está en función del tiempo, es decir que contiene exponenciales, funciones trigonométricas, etc. Ejemplos de estas fuentes serían: $100e^{-80t}$, $10e^{-t}sen(2t+30°)$ y otras por el estilo.

Lo que hace este modo es escribir y resolver las ecuaciones utilizando el valor de la fuente como una variable simbólica, llamada ϕ*nombre*, donde *nombre* es el de la fuente. Una vez se han resuelto las ecuaciones, antes de almacenar las respuestas, el *Impala* reemplaza la variable ϕ*nombre* por el valor de la fuente en dominio de la frecuencia. Esta simple sustitución ahorra una considerable cantidad de tiempo. En el siguiente ejemplo, y en un ejemplo del siguiente capítulo, veremos el *Modo (Impala)* en acción. Sabremos que este modo se activa cuando en la pantalla aparezcan los mensajes que así lo indican.

Este modo recibe su nombre en honor a un antílope africano, el cual al verse amenazado por sus depredadores, escapa dando prodigiosos saltos. Así también el *Symbulator*, al encontrarse con fuentes muy complejas, *salta*, evitando el uso de su valor complicado y resuelve las ecuaciones usando un valor simbólico simple; luego *aterriza*, reemplazando la fuente por su valor complicado en dominio de la frecuencia.

Problema N° 086

Planteamiento. Sean v_s=*100e*$^{-80t}$ y $v_1(0)$=20V en el circuito de la figura. a) Encuentre *i(t)*. b) Encuentre $v_1(t)$. c) Encuentre $v_2(t)$.

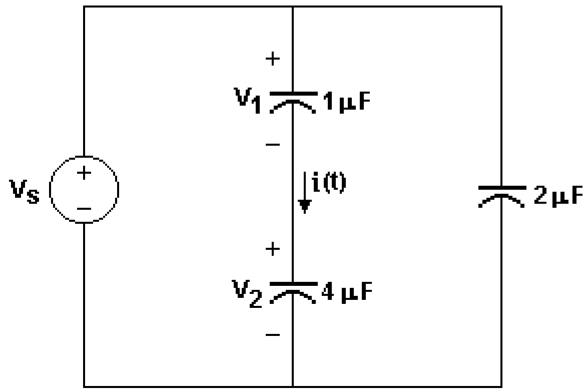

Figura 121. Circuito para el Problema N° 086.

Solución. El problema nos da la condición inicial del capacitor de 1µF, sin embargo no nos indica las de los capacitores de 4µF y de 2µF. Ejecutar una simulación en DC de este circuito no nos ayudará a encontrar estas condiciones iniciales. Pero podemos deducirlas fácilmente, usando la Ley de los Voltajes de Kirchhoff. Así, deducimos que la condición inicial del capacitor de 4µF es 80V, y la del capacitor de 2µF es 100V.

Como el problema nos pide tres respuestas en el dominio del tiempo, y el circuito es pequeño, usaremos TR en vez de FD. Usaremos valores exactos.

```
sq\tr("e,1,0,100e^(-80t),0;cc1,1,2,1*10^-6,20;
cc2,2,0,4*10^-6,80;cc3,1,0,2*10^-6,100")
```

Nótese que el orden en que se han definido los nodos de los capacitores está de acuerdo con la polaridad del voltaje inicial declarado. Presionamos ¸. Vemos que el *Modo (Impala)* toma parte en la simulación. Pedimos `icc1` y obtenemos `-4*e^(-80*t)/625`. Pedimos `vcc1` y obtenemos `80*e^(-80*t)-60`. Pedimos `vcc2` y obtenemos `20*e^(-80*t)+60`. Estas son las respuestas correctas.

Capítulo 12 Herramientas para graficar

12.1 Introducción

La calculadora TI posee poderosas capacidades para graficar. A su vez, el *Symbulator* incluye dos herramientas útiles para graficar diagramas de Bode y funciones del tiempo *t*.

12.2 La herramienta `bode`

La herramienta `bode` permite hacer diagramas de Bode. Se ejecuta así:

```
sq\bode()
```

Es muy fácil de usar. Aprendamos a usar esta herramienta resolviendo un problema real. Utilicemos uno que el Profesor Edilberto Yee, de la Universidad Tecnológica de Panamá, nos presentó como tarea final del curso de Electrónica III.

Problema N° 087

Planteamiento. Haga el diagrama de Bode de ganancia para Vo/Vg, desde 10^4 hasta 10^{13} rad/seg, para el siguiente circuito.

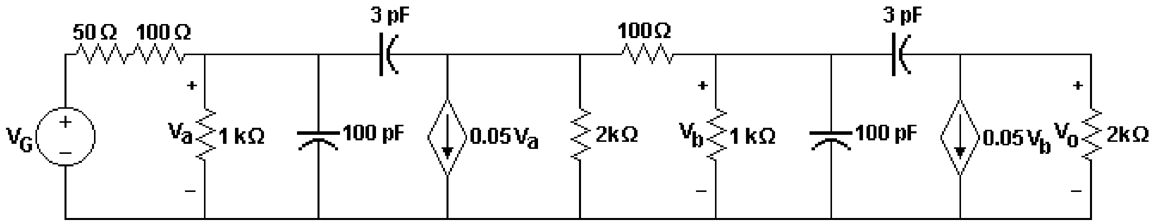

Figura 122. Circuito para el Problema N° 087.

Solución. Como hemos explicado ya, cuando se trata de circuitos grandes, una simulación con valores exactos nos dará respuestas precisas, mientras que una simulación con valores aproximados nos dará respuestas rápidas. Como este problema fue una tarea para resolver en casa, el tiempo no era crítico, y preferimos emplear valores exactos, para asegurar la precisión de las respuestas.

Damos la orden de ejecutar la simulación en dominio de la frecuencia, y ordenamos seguidamente la ejecución de la herramienta `bode`.

```
sq\fd("e1,5,0,vg,0;r1,5,1,150,0;r2,1,0,1000,0;cc1,1,0,
10^-10,0;cc2,1,2,3*10^-12,0;jd1,2,0,5/100*v1,0;r3,2,0,2000
,0;r4,2,3,100,0;r5,3,0,1000,0;cc3,3,0,10^-10,0;cc4,3,4,
3*10^-12,0;jd2,4,0,5/100*v3,0;r6,4,0,2000,0"):sq\bode()
```

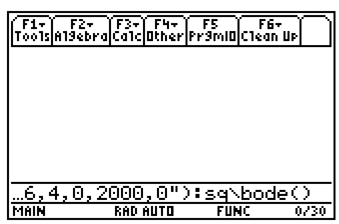

Figura 123. Orden del Problema N° 087, en la TI-89.

Presionamos ¸ . La simulación tarda alrededor de 3 minutos. Al final, se ejecuta la herramienta `bode`. La herramienta nos presenta un pequeño menú, el cual nos pide escoger entre hacer un gráfico de Bode o limpiar las variables creadas.

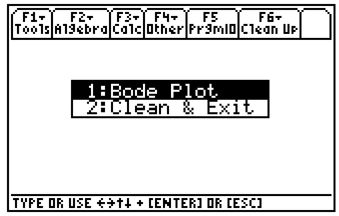

Figura 124. Primera selección de la herramienta bode, en la TI-89.

Escogemos la primera opción de este menú. La segunda se usa cuando, después de haber realizado algunas gráficas, deseamos terminar el trabajo y limpiar las variables que la herramienta haya creado. Por lo tanto, seleccionamos Bode Plot y presionamos ¸ . Aparece un menú.

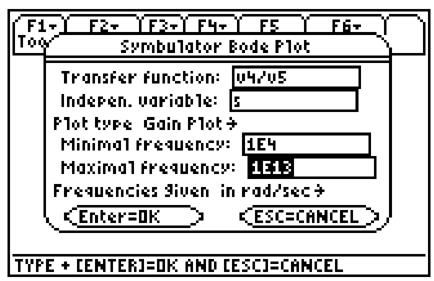

Figura 125. Menú de la herramienta bode, en la TI-89.

En este menú debemos introducir lo siguiente:

- *Función de transferencia.* Aquí colocamos la función de transferencia cuya gráfica deseamos hacer. Podemos escribir la función misma, o colocar el nombre de las variables que la definen. En nuestro caso, colocamos `v4/v5`, o sea Vo/Vg.
- *Variable independiente.* Es la variable que representa a la frecuencia. Casi siempre es *s*. Colocamos `s`.
- *Tipo de diagrama.* Hay tres tipos: ganancia (Gain Plot), fase (Phase Plot) y ambos (Both Plots). Si se escoge ganancia o fase, la pantalla completa mostrará la gráfica. Si se escoge ambos, la pantalla se dividirá en dos para mostrar ganancia arriba y fase abajo. En nuestro caso, seleccionamos `Gain Plot`. Para regresar la pantalla a su forma original, vamos a **3** y ponemos el modo Split Screen en FULL.
- *Frecuencia mínima.* Debe ser mayor que cero. En nuestro caso, colocamos `1E4`.

- *Frecuencia máxima.* Debe ser mayor que la mínima. En nuestro caso, colocamos `1E13`.
- *Unidades de la frecuencia.* Sirve para especificar en qué unidades introdujimos las frecuencias arriba. En nuestro caso, seleccionamos `rad/sec`.

Presionamos ¸. Aparece un mensaje, anunciándonos que las unidades del eje vertical son decibeles (dB), y las del eje horizontal son el logaritmo de la frecuencia en radianes/segundos.

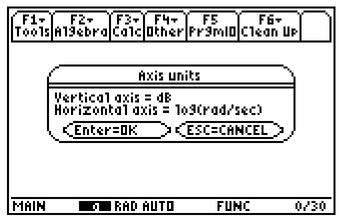

Figura 126. Anuncio de las unidades de los ejes, en la TI-89.

Presionamos ¸. Tras un minuto aproximadamente, aparece el diagrama.

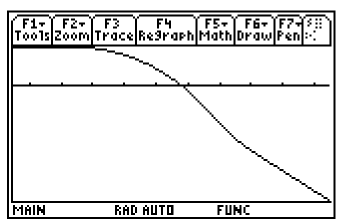

Figura 127. Gráfica de ganancia, en la TI-89.

Esta gráfica es la correcta. Obtenerla me permitió recibir todos los puntos en la tarea asignada. Haberla obtenido con un simulador hecho por mí mismo, me hizo además ser merecedor del respeto del Profesor Yee, a quien tengo en muy alta estimación.

Aquí termina la tarea. Sin embargo, supongamos que se nos hubiese pedido también el diagrama de fase. No hay necesidad de ejecutar nuevamente la simulación, pues las variables v4 y v5 siguen estando en la memoria. Basta con ejecutar nuevamente la herramienta `bode`.

```
sq\bode()
```

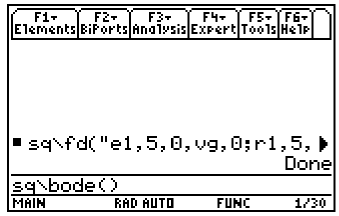

Figura 128. Ejecutando nuevamente la herramienta bode, en la TI-89.

Los valores previamente introducidos aún están ahí. Esto es gracias a ciertas variables que genera la herramienta, las cuales pueden ser borradas al final. Dejamos todo igual, pero seleccionamos `Phase Plot`.

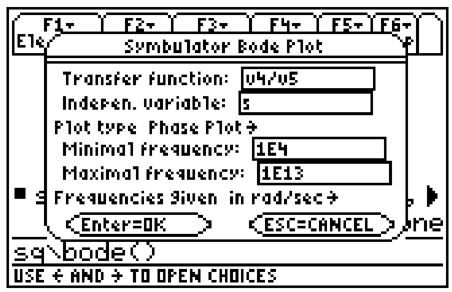

Figura 129. Menú en la herramienta bode, en la TI-89.

Presionamos ¸ . Aparece un mensaje, anunciándonos que las unidades del eje vertical son radianes, y las del eje horizontal son el logaritmo de la frecuencia en radianes/segundos.

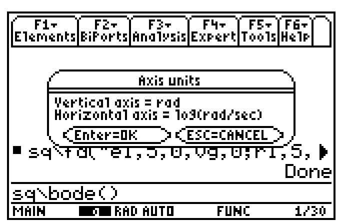

Figura 130. Herramienta bode, en la TI-89.

Presionamos ¸ . Tras un minuto aproximadamente, aparece el diagrama.

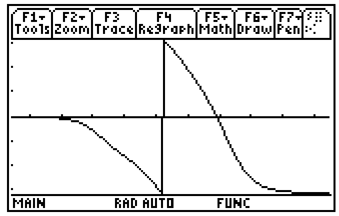

Figura 131. Herramienta bode, en la TI-89.

Habiendo terminado, ejecutamos nuevamente la herramienta y escogemos Clean & Exit para limpiar las variables creadas.

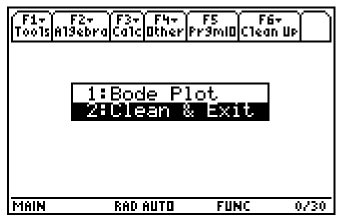

Figura 132. Limpiando las variables, en la TI-89.

12.3 La herramienta `plot`

La herramienta `plot` permite graficar funciones del tiempo t. Se ejecuta así:

```
sq\plot()
```

También es muy fácil de usar. Aprendamos a usar esta herramienta resolviendo un problema real. Utilicemos un problema que la Profesora Eliane Boulet de Cabrera, de la Universidad Tecnológica de Panamá, presentó a sus estudiantes en un examen. En este examen no se solicitaba la gráfica de las respuestas, pero nosotros podemos emplearlo para ilustrar el uso de la herramienta.

Problema N° 088

Planteamiento. Dado el siguiente circuito, encuentre la respuesta completa para v_2(t) y v_1(t) para t>0. Luego, grafique el valor de v_2(t) y v_1(t) *vs* tiempo, de 0 a 7 segundos.

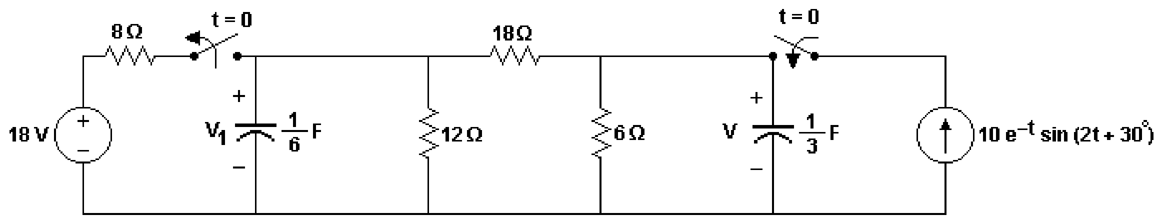

Figura 133. Circuito para el Problema N° 088.

Solución. Usaremos valores exactos para la precisión, pero ahorraremos tiempo usando FD en vez de TR, pues en medio de un exámen el tiempo es muy valioso. Damos la orden de ejecutar la simulación en corriente directa, para encontrar las condiciones iniciales.

```
sq\dc("e1,3,0,18;r1,3,1,8;cc1,1,0,1/6;r2,1,0,12;r3,1,2,
18;r4,2,0,6;cc2,2,0,1/3")
```

Para `vcc1` obtenemos `9`. Para `vcc2` obtenemos `9/4`. Estas serán nuestras condiciones iniciales para el siguiente intervalo. Damos la orden de ejecutar la simulación en dominio de la frecuencia.

```
sq\fd("cc1,1,0,1/6,9;r2,1,0,12,0;r3,1,2,18,0;r4,2,0,6,0
;cc2,2,0,1/3,9/4;js,0,2,10*e^(-t)*sin(2t+30°),0")
```

Presionamos ¸ . La simulación se ejecuta, con la participación del *Modo (Impala).* Ahora vamos a obtener las respuestas, y al mismo tiempo las almacenaremos en sus variables, para poder invocarlas luego desde la herramienta `plot`. Utilizamos `sq\fd_to_tr(v1)`. Obtenemos una expresión simbólica muy voluminosa. Por ello, utilizamos, `expand(approx(ans(1)),t)!v1t` obtenemos la respuesta completa en forma aproximada para el voltaje en el capacitor de la izquierda: `-.667*e^(-t)*cos(2.*t)-2.33*e^(-t)*sin(2.*t)+13.8*e^(-t/2)-4.16*e^(-t)`. Con `expand(approx(sq\fd_to_tr(v2)),t)!v2t` obtenemos la respuesta completa aproximada para el voltaje en el capacitor de la derecha: `-13.7*e^(-t)*cos(2.*t)+5.17*e^(-t)*sin(2.*t)+13.8*e^(-t/2)+2.08*e^(-t)`.

Ahora podemos ejecutar la herramienta `plot`.

```
sq\plot()
```

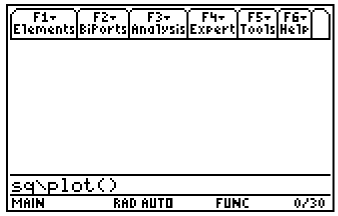

Figura 134. Ejecutando la herramienta plot, en la TI-89.

Aparece un menú, en el cual debemos insertar tres datos:

- *Función.* Es la función del tiempo *t* que deseamos graficar. En nuestro caso, `v2t`. Este es el nombre que dimos a la variable en la cual almacenamos el voltaje en dominio del tiempo.
- *Tiempo mínimo.* Es el tiempo mínimo de la gráfica, o sea el extremo izquiero del eje horizontal. En nuestro caso, `0`.
- *Tiempo máximo.* Es el tiempo máximo de la gráfica, o sea el extremo derecho del eje horizontal. Debe ser mayor que el mínimo. En nuestro caso, `7`.

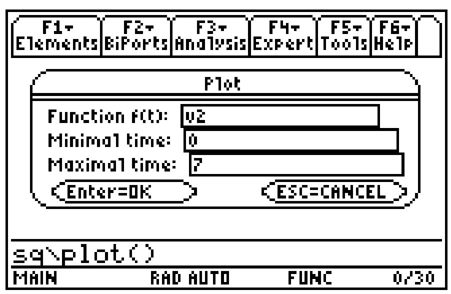

Figura 135. Menú de la herramienta plot, en la TI-89.

Presionamos ¸ . Aparece la gráfica.

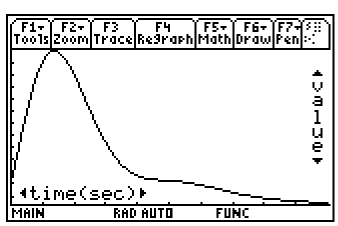

Figura 136. Gráfica del voltaje v2, en la TI-89.

Una vez tenemos esta gráfica lista en la pantalla, podemos aplicar sobre ella todos los comandos gráficos que posee la calculadora TI. Por ejemplo, el de valor máximo. Para mayor información sobre estos comandos, consúltese el *Manual de Usuario*.

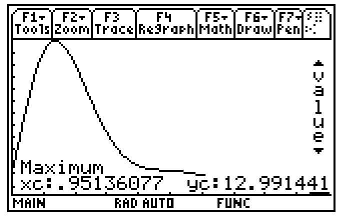

Figura 137. Comando de máximo, en la TI-89.

Para obtener la gráfica del voltaje `v1`, se repite el mismo procedimiento, pero introduciendo `v1` como la función que se desea graficar.

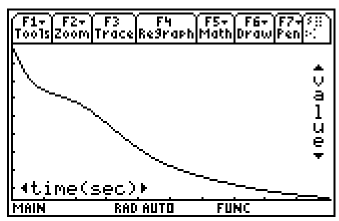

Figura 138. Gráfica del voltaje v1, en la TI-89.

Con esto, hemos terminado el problema. Como era de esperarse, la herramienta `plot` también puede graficar funciones que no sean respuestas del *Symbulator*. Por ejemplo, si introducimos como función la siguiente: $e^{-t}sen(10t)$, y graficamos entre 0 y 3 segundos, obtendremos la siguiente gráfica.

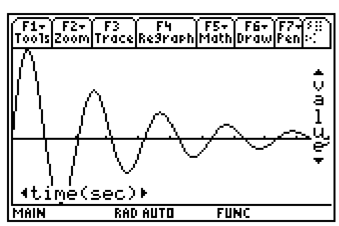

Figura 139. Gráfica de una función, en la TI-89.

De esta forma, hemos terminado este capítulo sobre el *Symbulator*.

Conclusiones

El *Symbulator* es una herramienta útil para el análisis de circuitos lineales. Permite el uso de valores numéricos y simbólicos. Puede simular en corriente directa, corriente alterna, dominio de la frecuencia y análisis transitorio. Entrega como respuestas todos los valores que interesan al usuario: voltajes en nodos, corrientes, caídas de voltajes y potencias en elementos. Facilita encontrar los equivalentes Thévenin, Norton y de bipuerto de una red.

Debido a sus capacidades, el *Symbulator* puede ser de gran ayuda en la enseñanza, aprendizaje y ejercicio de la Ingeniería Eléctrica, en el área de análisis de circuitos lineales, especialmente en aquellos problemas de circuitos que requieran de respuestas simbólicas.

Es de esperarse que la tecnología desarrollada en el *Symbulator*, y madurada durante estos dos años de pulimiento, sirva de base para desarrollar más y mejores productos en el futuro, incluyendo programas para nuevas plataformas tales como computadoras personales y computadoras portátiles.

Recomendaciones

No escribo estas recomendaciones como autor, sino basándome en mi propia experiencia como usuario, y en los testimonios que he recibido por escrito de los cientos de estudiantes y profesionales que han empleado el *Symbulator* como una herramienta de aprendizaje y verificación, durante los dos años de desarrollo.

Recomiendo que los profesores empleen el *Symbulator* como un instrumento de enseñanza en sus cursos de teoría de circuitos. Creo que este simulador tiene cabida en el salón de clases, debido a sus características y portabilidad. Los profesores pueden utilizarlo para verificar las claves de sus exámenes antes de calificarlos, y como parte de sus clases.

Considero que el estudiante debe aprender en primer lugar las técnicas manuales de resolución de circuitos, y que en las evaluaciones los profesores deben exigirle presentar todos los cálculos y pasos manualmente. Al mismo tiempo, recomiendo que se le permita al estudiante emplear instrumentos como el *Symbulator*, a manera de herramienta de verificación en clases, tareas y exámenes. Pienso que el tener una manera de verificar sus respuestas, favorecerá el aprendizaje y la confianza del estudiante en sí mismo. Especialmente porque los libros de textos de teoría de circuitos no incluyen respuestas a todos los problemas que presentan, y porque algunas de las respuestas que presentan están erradas.

Referencias

Análisis de circuitos en ingeniería

Hyat & Kemmerly

Quinta edición (Tercera edición en español)

Editora McGraw-Hill

1993

Manual de Usuario - Calculadora TI-89

Texas Instruments

Banta Book Group

1998

Manual de Usuario - Calculadora TI-92Plus

Texas Instruments

Banta Book Group

1999

Apéndice B
El *Symbulator* y la simulación simbólica

Los conocimientos que tiene el hombre sobre la electricidad han venido desarrollándose durante siglos, desde las ranas de Volta hasta los de transistores cuánticos. También se han desarrollado las herramientas de análisis de la ingeniería eléctrica. La simulación simbólica, por ser un invento muy reciente, está entre las más nuevas de estas herramientas.

En 1998, me topé con el primer simulador simbólico de circuitos que había visto en mi vida (y de hecho el único que conozco, aparte del *Symbulator*). Su nombre es *Syrup*, y está diseñado para ejecutarse en el entorno del programa matemático *MapleV* para computadora personal. Aunque nunca llegué a dominar este simulador, gracias a él comprendí que la simulación simbólica es una herramienta muy poderosa para el aprendizaje y el diseño. En aquel tiempo yo era estudiante del curso de Teoría de Circuitos II en el Centro Regional de la Universidad Tecnológica en Azuero, y sentí que sería muy útil tener un simulador simbólico disponible en el salón de clases. Entablé buena amistad con Joe Riel, el autor de *Syrup*, y él me explicó el concepto subyacente de este programa, un concepto extremadamente simple en verdad. A diferencia de los simuladores numéricos, incluso algunos que yo mismo había programado hasta la fecha, el *Syrup* no utilizaba matrices y análisis nodal modificado. Por el contrario, generaba ecuaciones nodales, las resolvía y las presentaba simbólicamente. Con esta idea en mente, el 30 de marzo de 1999 emprendí la programación de un simulador semejante en mi calculadora TI-89, al cual llamé inicialmente *SCS* (por *Symbolic Circuit Simulator*) y luego *Symbulator*. El *Syrup* fue una inspiración inicial, mas no un modelo a seguir, pues mi idea del simulador perfecto para el salón de clases discrepaba en algunos aspectos con el *Syrup*. Debido a que, durante los dos años que tardé en perfeccionarlo, no tuve contacto con ningún otro simulador simbólico, me ví forzado a inventar y desarrollar por mis propios medios las técnicas de la simulación simbólica que hoy presento, y que han rendido excelentes resultados a los miles de usuarios que el *Symbulator* tiene en el mundo entero. He hecho esta aclaración porque considero que el trabajo presentado en esta tesis debe tomarse como lo que es en verdad: una aportación pionera a un campo vírgen de la ingeniería eléctrica. Tal vez por eso mereció el primer lugar en el Concurso de Proyectos Estudiantiles IEEE 2000 para Latinoamérica (Región 9).

Apéndice C

¿Cuándo es preferible no usar el *Symbulator* ?

Una importante habilidad que debe desarrollar un usuario del *Symbulator* es la de reconcer cuándo la simulación le representa un ahorro de tiempo y cuándo no. Esta habilidad se manifiesta como una especie de olfato, de intuición, que le dice si, debido a la naturaleza de un problema, le resultará más complicado resolverlo con una simulación simbólica que con técnicas manuales. Se adquiere tras mucha práctica. Sólo la propia experiencia nos proporcionará esta intuición especial.

A continuación, presento un ejemplo que ilustra un caso en el cual es recomendable declinar el uso del simulador. Se trata de un circuito que, a primera vista, parece ser un candidato perfecto para la simulación simbólica en el *Symbulator*. Sin embargo, resulta mucho más sencillo usar una simple línea con el comando `solve` de la calculadora para resolverlo.

<u>Problema N° 089</u>

Planteamiento. En el circuito mostrado, se sabe que abs(I_1)=5 A y abs(I_2)=7 A. Encuentre I_1 e I_2.

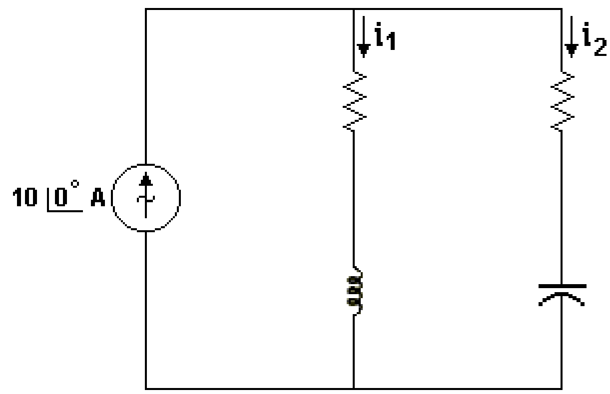

Figura 140. Circuito para el problema N° 089.

Solución. Simplemente, usaremos el comando `solve`:

`cSolve(abs(i1)=7 and abs(i2)=5 and i1+i2=10, {i1=1+1`***i***`,i2=1-1`***i***`})`

Nótense dos aspectos de esta línea:

1. Hemos escrito tres ecuaciones para encontrar dos incógnitas. Este podría parecer un error, pues se debe tener igual número de ecuaciones que de incógnitas. Sin embargo, cuando recordamos que cada incógnita compleja es en verdad dos incógnitas, comprendemos que deseamos resolver cuatro incógnitas y no dos. La tercera ecuación puede ser considerada, desde esta perspectiva, como dos ecuaciones en lugar de una, pues en ella se expresan implícitamente las siguientes dos ecuaciones: `real(i1)+real(i2)=10 and imag(i1)+imag(i2)=0`. Así, en la línea de comando mostrada arriba tenemos virtualmente cuatro ecuaciones con cuatro incógnitas.
2. Hemos agregado un signo de igualdad y un valor seguido al nombre de cada variable, al listar las incógnitas. En vez de decir `{i1,i2}`, hemos dicho `{i1=1+1`***i***`,i2=1-1`***i***`}`. Estos valores se conocen como *condiciones iniciales*, y su función es darle a la calculadora un punto de partida para la resolución numérica de las ecuaciones. Mientras más cercano al valor real sea el punto de partida, más rápido se encontrará la respuesta. Obsérvese que hemos dado un valor imaginario positivo a I_1, y un valor imaginario negativo a I_2. La razón es simple: sabemos de antemano que, como la

rama de I_2 tiene un capacitor, su parte imaginaria será negativa, mientras que la parte imaginaria de I_1 será positiva, pues hay una inductancia en esa rama.

Presionamos ¸ . Aparecen las respuestas. Dependiendo de los modos de la calculadora, las respuestas se mostrarán en forma rectangular o polar. Como deseamos verlas en forma polar, sin importar los modos, usamos la herramienta `absang`, así:

```
sq\absang(i1)|ans(1)
```

ans(1) se refiere a las respuestas obtenidas por el comando anterior. El número que está entre paréntesis indica a qué respuesta anterior nos referimos. Presionamos ¸ . Obtenemos `7. ∠27.66°`.

```
sq\absang(i2)|ans(2)
```

Presionamos ¸ . Obtenemos `5. ∠-40.54°`.

Nótese cómo hemos obtenido estas respuestas fácilmente, usando `solve`. Usando el *Symbulator*, el esfuerzo hubiese sido mucho mayor. Por eso, en este caso y en muchos otros casos similares a éste, es preferible declinar el uso del simulador, y tomar otro camino más sencillo.

Apéndice
D
Origen de los problemas usados

Los problemas de esta tesis fueron, en su mayoría, extraídos del siguiente libro: **Análisis de circuitos en ingeniería,** *Hyat & Kemmerly* **-** Quinta edición (Tercera edición en español). Editora McGraw-Hill. 1993. Las excepciones han sido señaladas. A continuación un listado de las páginas en las cuales se pueden encontrar los problemas usados en la tesis.

Capítulo	Problema	Fuente	Comentario
2	001	Hyat & Kemmerly	Página 56 Circuito a)
2	002	Hyat & Kemmerly	Página 57 Circuito b)
2	003	Hyat & Kemmerly	Página 34 Circuito a)
2	004	Hyat & Kemmerly	Página 34 Circuito b)
2	005	Hyat & Kemmerly	Página 33
2	006	Hyat & Kemmerly	Página 49 Problema 1-25
2	007	Hyat & Kemmerly	Página 30

2	008	Hyat & Kemmerly	Página 32 Figura 1-28b)
2	009	Hyat & Kemmerly	Página 48 Problema 1-23
2	010	Hyat & Kemmerly	Página 74 Figura 2-17
2	011	Hyat & Kemmerly	Página 47 Problema 1-15
3	012	Hyat & Kemmerly	Página 22
3 y 4	013	Hyat & Kemmerly	Página 27 Figura 1-24
3 y 4	014	Hyat & Kemmerly	Página 28 Figura 1-24b)
3 y 4	015	Hyat & Kemmerly	Página 28 Figura 1-24c)
3 y 4	016	Hyat & Kemmerly	Página 101 Problema 2-8
5	017	Hyat & Kemmerly	Página 81
5	018	Hyat & Kemmerly	Página 84 Figura 2-26a)
5	019	Hyat & Kemmerly	Página 85 Figura 2-27a)
5	020	Hyat & Kemmerly	Página 50 Problema 1-31
5	021	Hyat & Kemmerly	Página 493 Problema 15-17
5	022	Hyat & Kemmerly	Página 87 Figura 2-29a)
5	023	Hyat & Kemmerly	Página 106 Problema 2-29
5	024	Hyat & Kemmerly	Página 490 Problema 15-5
5	025	Hyat & Kemmerly	Página 106 Problema 2-31
6	026	Hyat & Kemmerly	Página 260 Figura 8-10a)
6	027	Hyat & Kemmerly	Página 261 Figura 8-11a)
6	028	Hyat & Kemmerly	Página 264 Problema 8-15
6	029	Hyat & Kemmerly	Página 261 Figura 8-11b)
6	030	Hyat & Kemmerly	Página 266 Problema 8-29
6	031	Hyat & Kemmerly	Página 432

6	032	Hyat & Kemmerly	Página 451 Problema 14-3
6	033	Hyat & Kemmerly	Página 453 Problema 14-15
6	034	Hyat & Kemmerly	Página 452 Problema 14-11
6	035	Hyat & Kemmerly	Página 456 Problema 14-31
6	036	Hyat & Kemmerly	Página 457 Problema 14-33
6	037	Hyat & Kemmerly	Página 457 Problema 14-35
6	038	Hyat & Kemmerly	Página 456 Problema 14-29
7	039	Hyat & Kemmerly	Página 494 Problema 15-19
7	040	Hyat & Kemmerly	Página 483 Figura 15-19d)
7	041	Hyat & Kemmerly	Página 466 Figura 15-6a)
7	042	Hyat & Kemmerly	Página 469
7	043	Hyat & Kemmerly	Página 470
7	044	Hyat & Kemmerly	Página 473 Figura 15-12
7	045	Hyat & Kemmerly	Página 483 Figura 15-19a)
7	046	Hyat & Kemmerly	Página 483 Figura 15-19b)
7	047	Hyat & Kemmerly	Página 483 Figura 15-19c)
7	048	Hyat & Kemmerly	Página 495 Problema 15-25
7	049	Hyat & Kemmerly	Página 484 Figura 15-20
7	050	Hyat & Kemmerly	Página 495 Problema 15-27
7	051	Hyat & Kemmerly	Página 495 Problema 15-23
7	052	Hyat & Kemmerly	Página 486 Figura 15-23
7	053	Hyat & Kemmerly	Página 496 Problema 15-33
7	054	Hyat & Kemmerly	Página 487 Figura 15-24a
7	055	Hyat & Kemmerly	Página 488 Figura 15-26

7	056	Hyat & Kemmerly	Página 497 Problema 15-37
8	057	Nevin McChesney (US)	-
8	058	Hyat & Kemmerly	Página 263 Problema 8-9
8	059	Hyat & Kemmerly	Página 264 Problema 8-11
8	060	Hyat & Kemmerly	Página 265 Problema 8-19
8	061	Hyat & Kemmerly	Página 266 Problema 8-25
8	062	Hyat & Kemmerly	Página 418 Problema 13-7
8	063	Hyat & Kemmerly	Página 418 Problema 13-3
9	064	Hyat & Kemmerly	Página 286 Problema 9-19
9	065	Hyat & Kemmerly	Página 492 Problema 15-15
9	066	Ing. Jake Adams (US)	-
9	067	Hyat & Kemmerly	Página 491 Problema 15-7
10	068	Hyat & Kemmerly	Página 375 Problema 12-19
10	069	Hyat & Kemmerly	Página 455 Problema 14-23
10	070	Hyat & Kemmerly	Página 423 Problema 13-41
10	071	Hyat & Kemmerly	Página 373 Problema 12-7
10	072	Hyat & Kemmerly	Página 373 Problema 12-11
10	073	Hyat & Kemmerly	Página 374 Problema 12-13
10	074	Hyat & Kemmerly	Página 374 Problema 12-15
11	075	Hyat & Kemmerly	Página 455 Problema 14-21
11	076	Hyat & Kemmerly	Página 376 Problema 12-27
11	077	Hyat & Kemmerly	Página 161 Problema 4-1
11	078	Hyat & Kemmerly	Página 162 Problema 4-11
11	079	Hyat & Kemmerly	Página 163 Problema 4-15

11	080	Hyat & Kemmerly	Página 165 Problema 4-23
11	081	Hyat & Kemmerly	Página 191 Problema 5-31
11	082	Hyat & Kemmerly	Página 191 Problema 5-33
11	083	Hyat & Kemmerly	Página 223 Problema 6-3
11	084	Hyat & Kemmerly	Página 223 Problema 6-5
11	085	Hyat & Kemmerly	Página 227 Problema 6-29
11	086	Hyat & Kemmerly	Página 144 Problema 3-19
12	087	Dr. Eliane Boulet de Cabrera (UTP)	-
12	088	Prof. Edilberto Yee (UTP)	-
Ap. C	089	Hyat & Kemmerly	Página 286 Problema 9-17

Apéndice A
Breve historia del *Symbulator*

El autor del *Symbulator* es Roberto Pérez-Franco, estudiante de Ingeniería Electromecánica en la Universidad Tecnológica de Panamá, egresado del Centro Regional de Azuero. Inició la programación del *Symbulator* bajo el nombre *SCS* el 30 de marzo de 1999. La primera versión en aparecer públicamente fue la 0.92, colocada en Internet el 2 de abril de 1999. La versión 1 incluyó inductancias mutuas. La versión 2, presentada el 19 de agosto de 1999 en el Centro Regional de Azuero, incluyó bipuertos. La versión 3 incorporó el *Modo (Experto)*. La versión 4, presentada en Internet el 9 de enero de 2001, incorporó el *Modo (Impala)*. La versión más reciente, llamada Q, fue presentada el 30 de marzo de 2001 en la Internet; en ella se introdujeron grandes mejoras, modificando alrededor del 40% del código.

El simulador se ha ofrecido permanentemente de forma gratuita en la Internet, y la retroalimentación de los usuarios ha permitido pulir el código, eliminar errores y agregar características novedosas.

Hoy en día, a dos años de su aparición pública, el simulador es usado por miles de estudiantes y profesionales de la ingeniería eléctrica alrededor del mundo, y ha recibido varios honores, incluyendo el **1er lugar en el Concurso de Proyectos Estudiantiles del**

IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), para el año 2000 en Latinoamérica (Región 9).

Idioma

El autor del *Symbulator* tiene el español como lengua nativa. Ha publicado múltiples libros en este idioma y se cuenta entre los escritores jóvenes más promisorios de su patria, Panamá. Aparte del español, maneja otros idiomas tales como el inglés y el Esperanto. Igualmente, ha cursado estudios de francés, y desea seguir ampliando y profundizando su espectro lingüístico en el futuro.

Cuando se inició el proyecto *Symbulator* en 1999, las calculadoras TI eran prácticamente inexistentes en Panamá. En el desarrollo de un programa, los usuarios juegan un papel vital, para ayudar con su retroalimentación a purgar el código de errores, y para dar ánimos al programador con sus comentarios y agradecimientos. Debe mencionarse que este programa no ha representado ninguna regalía para su autor, pues siempre ha sido de distribución gratuita.

Dado que la mayoría de los usuarios de esta calculadora en el mundo hablan el inglés como primera o segunda lengua, se hizo forzoso utilizar esta lengua como la lengua de trabajo para el programa y su documentación. Así, usuarios del programa en los cinco continentes pudieron enviar sus comentarios y reportes al autor. Como las capacidades del programa han ido ampliándose vertiginosamente en estos dos años, el rápido cambio de las versiones no ha dejado tiempo disponible para traducirlas. Hasta el día de hoy el *Symbulator* no ha dejado de crecer, y por ello tanto el programa como la documentación siguen estando en inglés.

Le expreso mis disculpas a cualquiera que se pueda sentir ofendido por esta situación. El deseo del autor siempre ha sido desarrollar una herramienta útil para todos los estudiantes de ingeniería del mundo. De hecho, el interés del autor en el Esperanto tiene su origen en su inconformidad con la falta de una verdadera lengua internacional neutra, común a todos los humanos.

Es necesario notar que a todos los usuarios que han planteado preguntas en español, se les ha respondido en ese idioma. Igualmente, este escrito – el cual es la más completa documentación del Symbulator – ha sido originalmente redactada en español. Se planea su traducción al inglés y al Esperanto, entre otros idiomas, a corto plazo.