

ESPICE: UN NUEVO SIMULADOR DIDÁCTICO DE CIRCUITOS ELECTRÓNICOS. APLICACIONES.

A. Roldán, J.B. Roldán

Departamento de Electrónica y Tecnología de los Computadores.

Facultad de Ciencias. Universidad de Granada. España.

amroldan@ugr.es

Se presentan las características de un nuevo simulador didáctico de circuitos electrónicos multiplataforma, con interfaz gráfica avanzada GTK+, denominado ESPICE. El simulador basado en SPICE tiene nuevas funcionalidades que posibilitan el acceso a los datos internos de la simulación; el análisis e interpretación del fichero de simulación, las matrices de tensiones y corrientes en cada iteración; la inclusión de nuevas funciones de procesado de datos, tipos de análisis, modelos de dispositivos y obtención del ejecutable mediante la compilación en Windows y Linux previa descarga del código fuente del repositorio público. Así mismo, se presenta un conjunto de recursos multimedia que acompañan al simulador.

1. Introducción

Desde que el Profesor Donald O. Pederson de la Universidad de Berkeley, California (USA), presentara el 12 de Abril de 1973, el simulador SPICE [1] (“Simulation Program with Integrated Circuits Emphasis”), en el “*Sixteenth Midwest Symposium on Circuit Theory*“, nadie podría haber adivinado el impacto que iba a tener en la comunidad científica y en el conjunto de empresas del sector de la electrónica.

Una de las razones que ha hecho que *SPICE* haya sido un referente en el sector del análisis de circuitos es su gran expansión, producida básicamente por la decisión de dejar acceso público a las fuentes del programa. Este software fue una de las primeras distribuciones *Freeware*. Su origen está ligado al desarrollo de una herramienta didáctica concebida para introducir al alumnado en el estudio de los circuitos integrados, lo que permitió que *SPICE* se difundiera rápidamente por las escuelas de ingeniería de todo el mundo. La otra fase de auge en el uso del simulador comenzó cuando los alumnos, familiarizados con su utilización, iniciaron su vida profesional en las empresas del sector, donde introdujeron el simulador *SPICE* que, desde la Universidad de Berkeley, se facilitaba gratuitamente.

Muchos de los módulos principales de *SPICE* fueron extraídos de los trabajos de un grupo de alumnos en un proyecto dirigido por Ron Rohrer, entre ellos:

- § El procedimiento de resolución de sistemas de ecuaciones matriciales y dispersos, que permitían resolver circuitos de un orden de magnitud superior a los programas que hasta entonces existían.
- § El uso de algoritmos de integración implícita que proporcionaban soluciones más robustas en los análisis transitorios.
- § La inclusión de modelos de dispositivos semiconductores que posibilitaban al usuario su utilización con sólo proporcionar un conjunto reducido de parámetros, evitando el tedioso sistema usado hasta entonces con las versiones de FORTRAN¹.

¹ FORTRAN “**FOR**mula **TRAN**slation” desarrollado durante 1954-1957 por John Bachus para IBM

Este trabajo presenta una actualización didáctica del clásico simulador SPICE en la que se han mejorado algunos de los modelos de dispositivos existentes hasta la fecha, se han depurado e introducido nuevos algoritmos de análisis para mejorar la convergencia de gran variedad de circuitos electrónicos. También se ha permitido la salida por pantalla de los valores de parámetros internos del simulador, se ha incluido un editor de texto que permite la edición del fichero de descripción del circuito o *netlist* y compilación multiplataforma Windows-Linux por parte del usuario con posibilidad de modificación del código fuente.

Las ventajas de este simulador didáctico en el aprendizaje de las técnicas de análisis de circuitos en las titulaciones técnicas de electrónica son importantes ya que el núcleo de los simuladores profesionales existentes en el mercado (*PSPICE*, *HSPICE*, *ADS*, etc.) es similar al de *ESPICE*, y, el acceso a gran cantidad de información y procesos internos de la herramienta permite conocer y mostrar el procedimiento sistemático de análisis de un circuito.

La denominación elegida para el simulador didáctico *ESPICE*, representa los dos conceptos perseguidos y conseguidos por el equipo de desarrollo. *ESPICE*: *E* que indica que el simulador tiene todo su interfaz en castellano (aunque se pueden seleccionar otros idiomas: español, inglés, francés, alemán, valenciano y catalán) y *SPICE* porque contiene en su núcleo el esquema de análisis propuesto por el equipo de la Universidad de Berkeley con gran cantidad de actualizaciones.

2. Motivación

Existen en la actualidad bastantes programas de simulación de circuitos que se utilizan en la industria electrónica y en las instituciones académicas, todos son de pago. De hecho, las herramientas software que se utilizan en la industria electrónica actual son, en general, de las más caras del mercado debido al alto valor añadido de los productos electrónicos y su demanda en la sociedad actual. Estos altos precios dejan fuera del alcance de cualquier estudiante de electrónica estos programas.

A veces, algunas empresas crean una versión reducida de la herramienta, la denominada *versión estudiante*, con capacidades muy limitadas para que los alumnos se familiaricen con ellas. En este contexto, nuestro objetivo es el desarrollo de una herramienta de simulación de circuitos, que a pesar de tener una interfaz de usuario sencilla en comparación con los programas comerciales actuales, permite hacer las simulaciones que los alumnos deben realizar en su estudio de los circuitos como paso previo al montaje experimental de éstos en el laboratorio de electrónica sin limitación alguna en el número de componentes del circuito.

De manera resumida las ventajas que presenta *ESPICE* para la docencia son:

- § La interfaz del simulador está en castellano aunque también implementa otros idiomas, lo que permite su utilización por toda la comunidad docente iberoamericana.
- § Se ha optimizado la interfaz de usuario de modo que integra un editor de texto para la creación/modificación de los ficheros de descripción del circuito o ficheros *netlist*.
- § Un gestor de librerías de componentes electrónicos que aglutina gran cantidad de librerías de elementos y subcircuitos ofrecidos por los fabricantes de componentes electrónicos.
- § Añade la posibilidad de modificar todas las opciones de control del simulador con una explicación de la función que realiza cada una.
- § Permite tener acceso a datos internos como las matrices dispersas que representan el sistema de ecuaciones de tensiones y corrientes del circuito en cada iteración y la actividad desarrollada por el analizador sintáctico que interpreta toda la información del fichero de simulación y la consola de entrada de datos del usuario.

- § Habilita el uso de los sistemas operativos habituales disponibles en el aula ya que el simulador se puede compilar tanto en Windows como en Linux.
- § Distribución gratuita del código fuente y de sus actualizaciones mediante acceso WEB al repositorio <http://espice.ugr.es/repos/espice>
- § Oferta de un conjunto de herramientas multimedia de ayuda y documentación de uso de ESPICE disponible para todos los usuarios con ejemplos, explicaciones, ejercicios resueltos, etc., accesible en <http://espice.ugr.es>.
- § Plataforma de desarrollo con acceso libre a los programadores para realizar análisis sobre el avance de versiones, insertar avisos de errores detectados y mejoras, accesible en <http://espice.ugr.es/trac>
- § Permite la inclusión y modificación de los modelos de los dispositivos existentes lo que lo hace muy interesante en actividades de I+D en el campo del modelado compacto de dispositivos electrónicos.

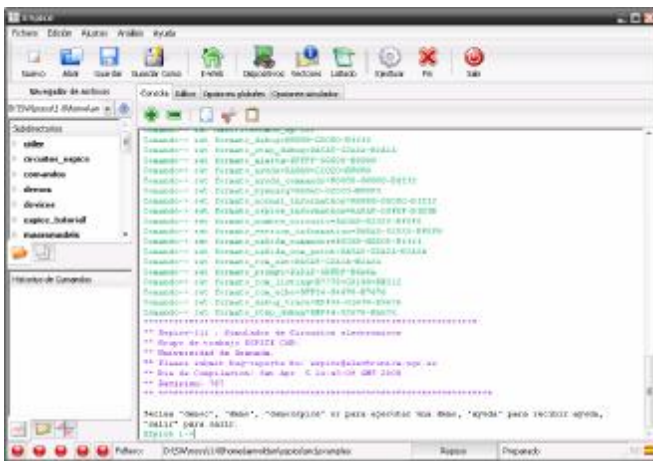


Figura 1. Captura consola entrada-salida.

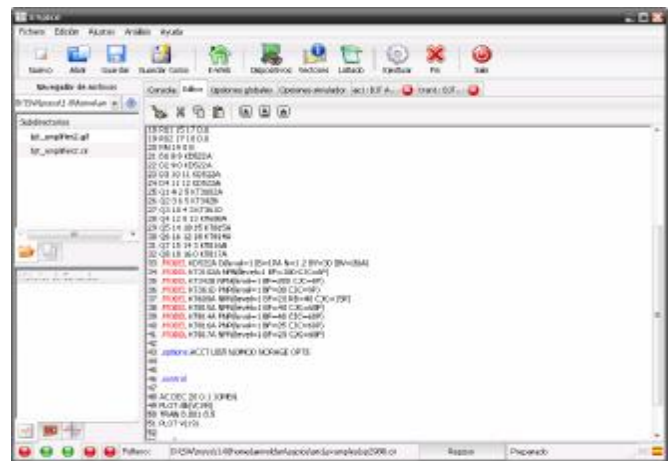


Figura 2. Captura consola editor de texto

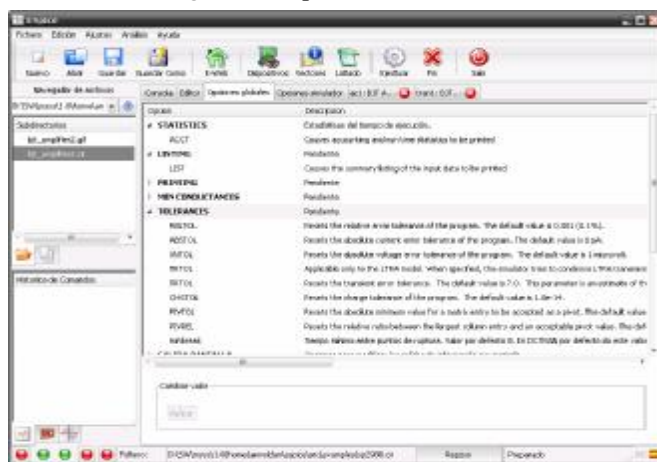


Figura 3. Captura consola opciones internas.

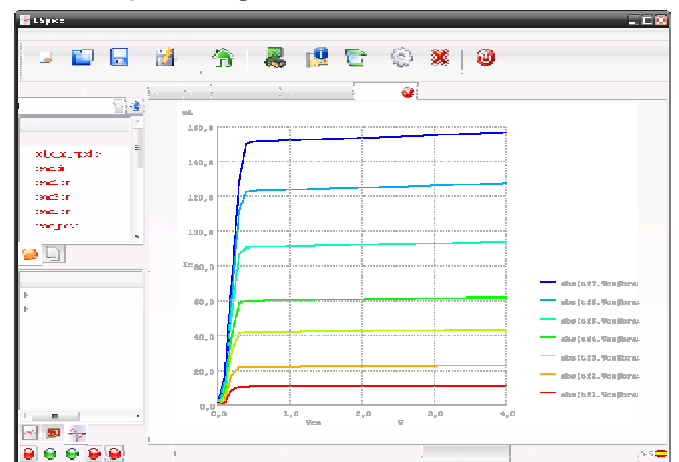


Figura 4. Captura consola gráfica.

En el pasado, se han realizado intentos de desarrollo de simuladores basados en SPICE parecidos al nuestro (NGSpice, WinSPICE, OPUSSpice, etc.), pero, en la mayoría de las ocasiones: no se han terminado o no se ha alcanzado el nivel obtenido con ESPICE. Como se puede comprobar, haciendo uso de la herramienta, el grado de depuración y usabilidad logrado mediante la interfaz gráfica permite afirmar

que estamos ante una de las mejores herramientas no comerciales de simulación de circuitos didáctica de la actualidad.

Para mejorar sus prestaciones se han incluido los siguientes módulos:

- § Núcleo principal de *SPICE*, que no necesita presentación ya que es el simulador de circuitos más popular en el sector de la electrónica. Tras sus 30 años de vida, Spice3 se ha convertido un un standar de-facto en la simulación de circuitos electrónicos. Origen: Univ. de Berkeley
- § *CIDER* está enlazado con Espice para realizar simulaciones de dispositivos electrónicos mediante el simulador *DSIM*. Esta herramienta permite un mayor grado de precisión en la simulación de dispositivos críticos y nuevos prototipos. Los dispositivos *DSIM* se describen en términos de su estructura geométrica y materiales que lo forman. Origen: Univ. de Berkeley.
- § *XSPICE* es una extensión de *SPICE* que añade el soporte de modelado compacto y simulación de componentes digitales al incluir un algoritmo controlado por eventos digitales. Origen: Georgia Institute of Technology

Durante varios cursos, *ESPICE* ha sido utilizado por los alumnos de la Universidad de Granada de los primeros cursos de las ingenierías de Telecomunicación, Informática y Electrónica. Se han obtenido unos resultados inmejorables. A parte, se han incorporado las mejoras y ampliaciones sugeridas por los usuarios.

3. Arquitectura del simulador

Para el desarrollo del nuevo diseño del simulador didáctico se ha tenido en cuenta las características particulares de la actividad docente que se imparte actualmente en las aulas. En particular, la configuración variada de ordenadores y de sistemas operativos, donde una sencilla instalación multiplataforma permite gran flexibilidad. Estos requerimientos han influenciado la elección de un gestor gráfico de ventanas de usuario, GUI, multiplataforma y un sistema unificado de ficheros válido en sistemas Windows-Linux; codificado, todo ello, en lenguaje *C*. El tamaño del ejecutable se ha intentado minimizar al máximo, de modo que en 4.5 Mb se han incluido las funciones necesarias con la potencia de las ventanas gráficas de *GTK+*, lo que permite una rápida instalación que evita alguno de los problemas de tiempo y espacio de las versiones limitadas de estudiante.

En la figura 5, se representa el diagrama de bloques correspondiente a la arquitectura elegida para *ESPICE*. En la parte inferior se detalla el módulo de modelos de dispositivos (fuentes tensión, corriente independientes y dependientes, elementos pasivos, semiconductores, etc.) que han sido implementados como librerías dinámicas cargadas bajo demanda por el núcleo. A su vez, éste se comunica con una consola bidireccional virtual con el GUI (Graphical User Interface, interfaz gráfica de usuario) *GTK+*, a través de la cual, el usuario recibe la entrada y la salida. Mediante los sistemas *OpenGL*, *X11* y *GTK+* se generan las ventanas correspondientes a las gráficas y la interfaz de ventanas.

4. Procedimiento de actualización del sistema operativo

Para poder ejecutar el simulador de circuitos en el sistema operativo elegido (Linux, Windows) es necesaria realizar previamente la instalación de algunos paquetes software relacionados con el Gestor de Ventanas multiplataforma *GTK*. Del mismo modo y en el caso de que el usuario prefiera realizar la compilación del código fuente para introducir alguna modificación o conocer el procedimiento de obtención del ejecutable, será necesaria la instalación adicional de algunos paquetes más orientados al desarrollo de aplicaciones basadas en *GTK*.

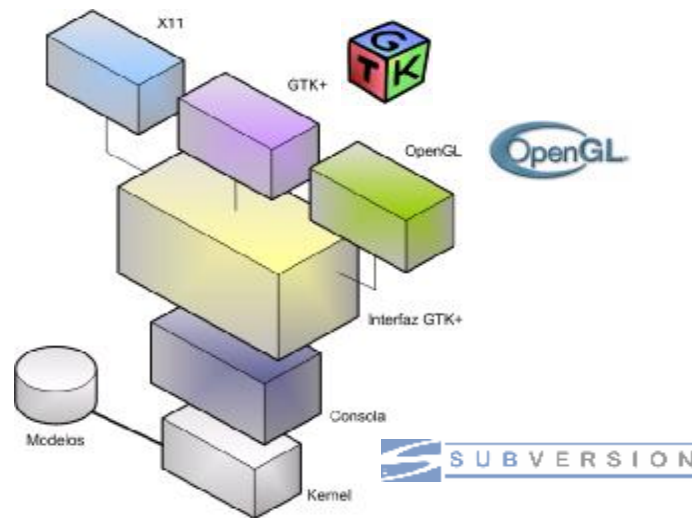


Figura 5. Arquitectura de ESPICE.

Con el fin de evitar la descripción en detalle de las librerías para ejecución y compilación, se presenta a continuación una única lista con los paquetes necesarios:

- WINDOWS XP-W2003-VISTA:
 - § GTK-win32: gtk-2.10.11-win32-1.exe [o superior]
 - § GTK-sourceview: libgtksourceview-1.6.0-devel-win32.zip
 - § MinGW: <http://www.mingw.org/>
[Utilidad: instalación del paquete necesario para descarga del código fuente desde el repositorio]
 - § Msys: <http://www.mingw.org/msys.shtml>
[Utilidad: plataforma de compilación de aplicaciones Linux en Windows]
- Distribuciones LINUX:
 - § *DEBIAN*: automake, libtool, autoconf, libgtk2.0-dev, libgtkglext1-dev, libxaw7-dev, libgtksourceview1.0-0-dev
 - § *UBUNTU*: automake, libtool, libgtk2.0-dev, libxaw7-dev, libgtksourceview-dev, texinfo, tex-common
 - § *SUSE*: dbus-1-glib-devel, esound-devel, gconf2-devel, gnome-vfs2-devel, gtksourceview-devel, hal-devel, libbonobo-devel, libgnome-devel, libgnomeprint-devel, libidl-devel, orbit2, orbit2-devel, popt-devel

El procedimiento de instalación en Windows se simplifica porque el usuario únicamente tendrá que ejecutar unos programas que están acompañados de su asistente de instalación para lo cual será necesario tener privilegios de administrador. Por el contrario en Linux, el procedimiento conlleva la instalación desde los repositorios de la distribución, de los paquetes enumerados anteriormente. El procedimiento no es único. Sin pérdida de generalidad y aprovechando su sencillez, se presentará en este trabajo, el utilizado en la distribución DEBIAN.

```
user2> sudo apt-get install automake, libtool, autoconf, libgtk2.0-dev, libgtkglext1-dev, libxaw7-dev, libgtksourceview1.0-0-dev
```

² La cuenta de usuario, en adelante “*user*” deberá disponer de privilegios suficientes para ejecutar el comando “*sudo*” que permite suplantar al administrador o “*root*” del sistema, de modo que, se puedan instalar nuevos paquetes software en el sistema.

5. Procedimiento de compilación del código fuente

Vamos a detallar en este apartado el proceso de compilación necesario para poder hacer uso del simulador en las plataformas más usuales que podemos encontrar en los laboratorios de nuestros centros educativos. Antes de comenzar, es necesario verificar que se tiene conexión a Internet.

- LINUX:

```
user> sudo apt-get install subversion
```

[ACCIÓN: instalación del paquete necesario para conexión y descarga del código fuente desde el repositorio]

```
user> svn co http://espice.ugr.es/repos/espice [ACCIÓN: descarga del código fuente y salvado en el directorio /espice]
```

```
user> cd espice
```

```
user> ./compila
```

[ACCIÓN: ejecución del script que inicia la generación de ficheros, compilación y enlazado.]

```
user> cd src
```

```
user> ./espice
```

[ACCIÓN: ejecución del simulador.]

- Windows:

```
user> sudo apt-get install subversion
```

```
user> svn co http://espice.ugr.es/repos/espice
```

```
user> cd espice
```

```
user> ./compila_win
```

```
user> cd src
```

```
user> ./espice
```



```
andres@PORTATIL_A ~$ cd espice
andres@PORTATIL_A ~$ ls
ANALYSES      Makefile.in      config.sub      patches
AUTHORS       NEWS             configure       sh_ews_stackdua
BUGS          README          contrib         src
COPIES        Stuart_Poly_Notes contrib         stop-h.in
ChangeLog     autogen.sh      copisDetos.sh  stop-hl
DEVICES       automática cache depccap         svn-commit.2.ts
Dextyfile     bienvenida      doc             svn-commit.3.ts
Espice_kdevelop compila_win     examples       svn-commit.tap
Espice_kdevelop_pos compila_win     install-sh     svn_subir
Espice_kdevelopms config_guess    lib            tests
FAQ           config.h        libtool        xgraph
INSTALL       config.h.in     libwin.sh     yltop
Makefile      config.log      man
Makefile.am   config.status   missing
```

```
andres@PORTATIL_A ~$ cd espice
andres@PORTATIL_A ~$ svn update
U      src\lib\language\espice_sp.wln
G      compila_win
Actualizado a la revision 766.
andres@PORTATIL_A ~$
```

Figura 6. Estructura de directorios en MSYS.



Figura 7. Control de cambios efectuados.

En la figura 6, se muestra la estructura de directorios creados por subversión [3] sobre una ventana de consola *Msys* [4] ejecutada sobre Windows. En la parte inferior de ésta, se ha realizado una actualización del código fuente del repositorio del que se ha descargado el fichero “*espice_sp.wln*” y el script de compilación automática para Windows “*compila_win*” donde se informa sobre la versión descargada del repositorio, que en este caso es la número 766. En la figura 7, se muestra una captura de la información WEB proporcionada por el gestor de cambios *TRAC* [5] donde se detalla el listado de versiones disponibles en el repositorio así como una breve explicación de los cambios realizados en cada actualización.

4. Aplicaciones

Para poder mostrar el potencial del simulador didáctico se presentan dos aplicaciones diferentes. En la primera, se hace hincapié en la capacidad de análisis del simulador y en la segunda (apartado b) se muestra la potencia del análisis del lenguaje de *scripts* que incorpora *ESPICE*:

a.- Un circuito formado por 4 secciones de filtro pasivo L-C de segundo orden, similar al presentado en la figura 8 [6], con un valor del parámetro Q de cada filtro de 0.5, 1, 2 y 4, y una configuración que nos permite estudiar la respuesta del circuito con diferentes impedancias de entrada en una única simulación:

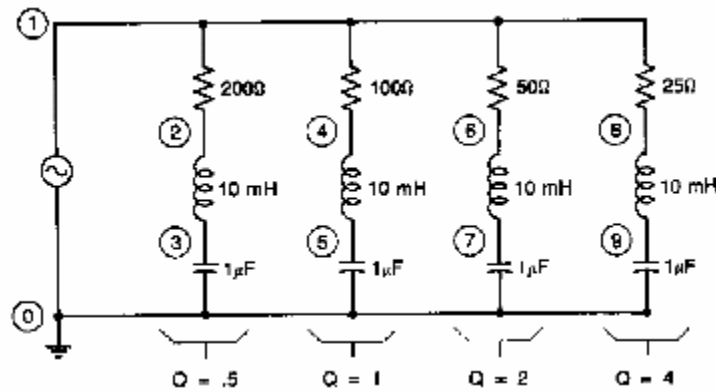


Figura 8. Filtro pasivo L-C de segundo orden

La descripción del circuito se presenta en el listado 9. Se ha elegido un generador de excitación con una amplitud de 1V AC de manera que se simplifica la obtención de la función de transferencia en pequeña señal al no ser necesario dividir por la amplitud del generador de entrada.

Los valores de los elementos del circuito se han seleccionado para que la frecuencia de resonancia de los filtros pertenezca a la banda de audio y presenten un valor diferente del factor de calidad Q.

```

Four Double Pole, low pass, LC-Filter
Vin 1 0 AC=1 DC=0
*Q=0.5
R1 1 2 200
L1 2 3 10mH
C1 3 0 1uF
*Q=1
R2 1 4 100
L2 4 5 10mH
C2 5 0 1uF
*Q=2
R3 1 6 50
L3 6 7 10mH
C3 7 0 1uF
*Q=4
R4 1 8 25
L4 8 9 10mH
C4 9 0 1uF
    
```

Listado 9. Netlist del filtro pasivo L-C

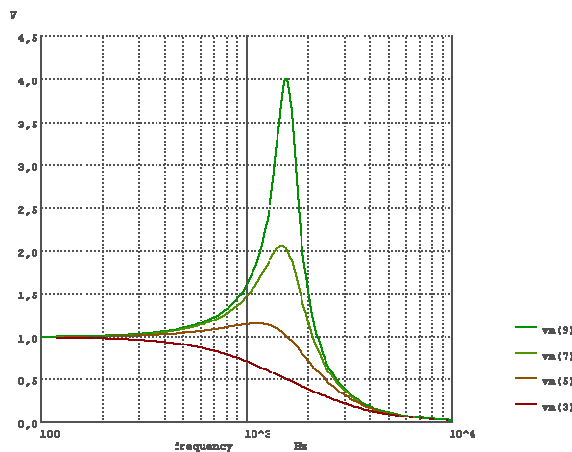


Figura 10. Respuesta en magnitud.

En la figura 10 se muestra la respuesta en magnitud calculada por el simulador para las diferentes secciones L-C. El significado de las curvas corresponde a:

- vm(9) → respuesta en magnitud para filtro L-C de Q=0.5
- vm(7) → respuesta en magnitud para filtro L-C de Q=1
- vm(5) → respuesta en magnitud para filtro L-C de Q=2

- $vm(3) \rightarrow$ respuesta en magnitud para filtro L-C de $Q=4$

En la figura 11 se muestra la respuesta en fase de las secciones, medida en grados, en los terminales de salida de los condensadores, que nos informa sobre la posible distorsión de fase que sufre la envolvente de señal al atravesar dicho filtro. La respuesta en fase presenta una relación entre el análisis en frecuencia y el temporal. Para mantener la envolvente de la señal que atraviesa el filtro es necesario asegurar la linealidad con la frecuencia de la respuesta en fase. Esto es equivalente a decir que el filtro introduce un retraso temporal uniforme sobre la señal. Así pues, para poder evaluar esta propiedad del filtro, conocida como retraso de grupo, es necesario calcular la derivada de la respuesta en fase con respecto a la frecuencia (1).

$$vg = -\frac{dV_{PHASE}(nodo)}{dw} = -\frac{dV_{PHASE}(nodo)}{2p \cdot df} = -\frac{dvp(nodo)}{2p \cdot df} \quad (1)$$

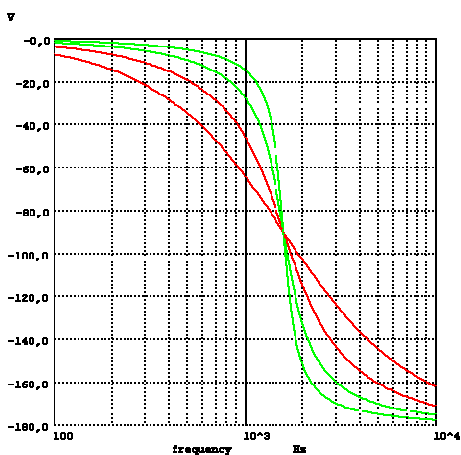


Figura 11. Respuesta en fase.

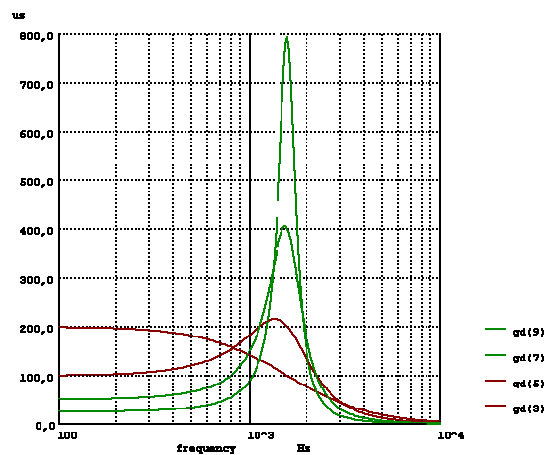


Figura 12. Retraso de grupo.

El retraso de grupo para las 4 secciones L-C se expone en la figura 12 donde hay que tener en cuenta que el error obtenido al realizar el cálculo de la derivada numérica sobre un conjunto de puntos finitos será tolerable solo en el caso de que las diferencias en fase entre valores contiguos sean pequeñas.

La impedancia de entrada es un parámetro de importancia y normalmente sufre variaciones con la frecuencia y para su cálculo vamos a utilizar un generador de tensión fija y nula en serie con la rama de cada sección L-C que nos permitirá obtener la corriente de entrada. Utilizando un comando de representación gráfica (2) sobre la consola de entrada-salida del simulador, podremos generar la figura 13 donde se muestra el módulo de la impedancia de entrada de las secciones.

ESpice 1-> plot v(1)/mag(i(VAUX1)) v(1)/mag(i(VAUX2)) v(1)/mag(i(VAUX3)) v(1)/mag(i(VAUX4)) xlimit 100 10k ylimit 10 1000 loglog (2)

Del mismo modo, se puede obtener la representación de la fase de la impedancia de entrada de las secciones.

ESpice 2-> plot (-ph(i(VAUX1))) (-ph(i(VAUX2))) (-ph(i(VAUX3))) (-ph(i(VAUX4))) xlog title "Arg(Zin)" (3)

Finalmente y para poner de manifiesto la potencia de la representación gráfica del simulador se presenta la tensión de salida de cada sección en coordenadas polares (4) y la impedancia de entrada en representación polar en la carta de Smith (5) [7]:

ESpice 4-> plot v(3) v(5) v(7) v(9) polar pointplot (4)

ESpice 5-> plot norm(v(1)/i(VAUX1)) norm(v(1)/i(VAUX2)) norm(v(1)/i(VAUX3)) norm(v(1)/i(VAUX4)) smith (5)

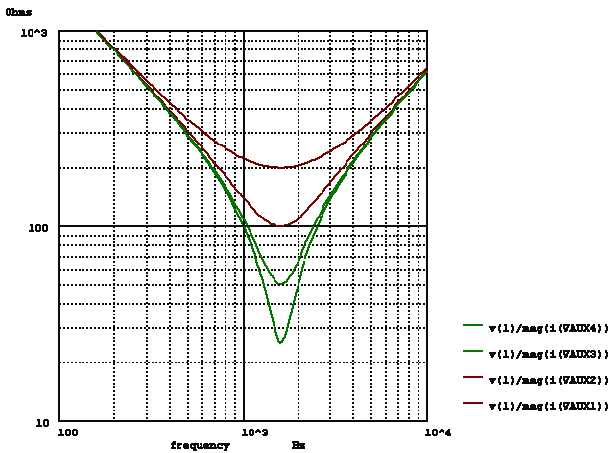


Figura 13. Representación $|Z_{in}|$.

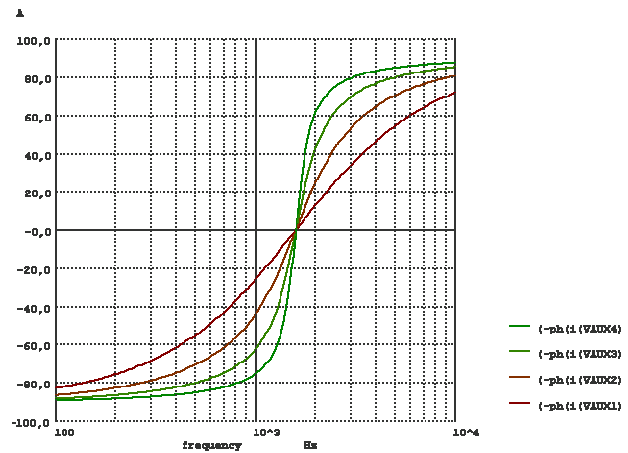


Figura 14. Representación $Arg(Z_{in})$.

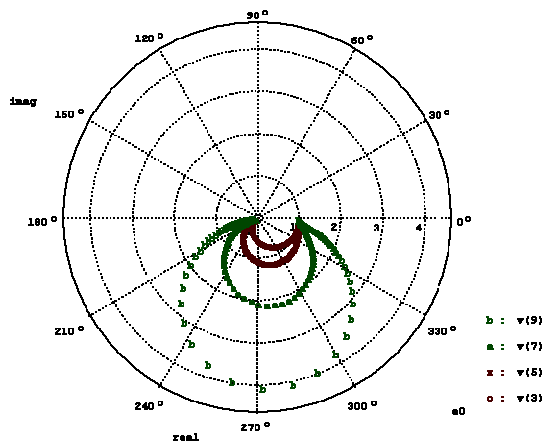


Figura 15. Tensión salida en condensadores.

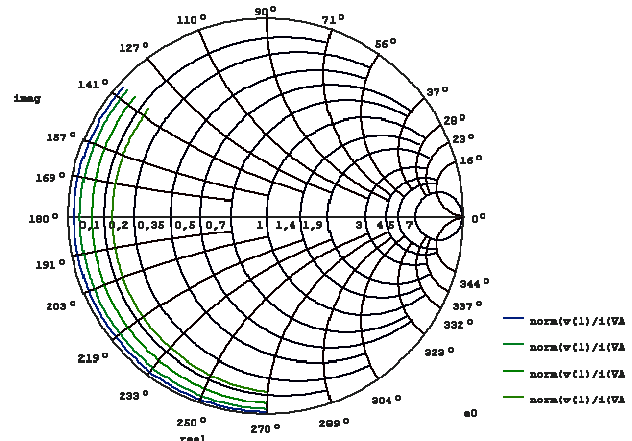


Figura 16. Impedancia de entrada normalizada.

b.- Amplificador de dos etapas realimentado con BJT; ver figura 17, en el que se plantea la necesidad de obtener la ganancia en bucle cerrado [8]. Para ello, es necesario romper el lazo de realimentación sin perturbar las condiciones estáticas en el punto donde éste sea roto. El punto elegido ha sido la unión entre el colector Q_1 y la base de Q_2 (señalado con el símbolo X). Para poder realizar la simulación, introduciremos un inductor en serie L_t de 1GH en serie con los terminales de ruptura del bucle. Dicha inductancia presenta una gran impedancia para las señales en AC, manteniéndose sin alteración el punto de polarización. En la figura 18 se puede ver la nueva configuración del circuito.

Para calcular la impedancia equivalente que permitirá abrir el bucle, debemos introducir una tensión de 1V AC en el generador V_t en la base de Q_2 o nodo 21 a través de un condensador de desacoplo $C_{ii}=1GF$ de gran valor y de este modo obtener la impedancia equivalente (6) que lógicamente dependerá de la frecuencia:

$$Z_{in}(f) = \frac{V_t}{I_t} \quad (6)$$

El rango de frecuencias del estudio debería estar dentro del ancho de banda del amplificador, en nuestro caso y dado que la impedancia es puramente resistiva trabajaremos únicamente a la frecuencia de 1Hz. La obtención de la corriente y tensión de interés sobre la fuente V_t y el nodo 22 respectivamente se puede realizar mediante el uso de la siguiente orden en la consola del simulador.

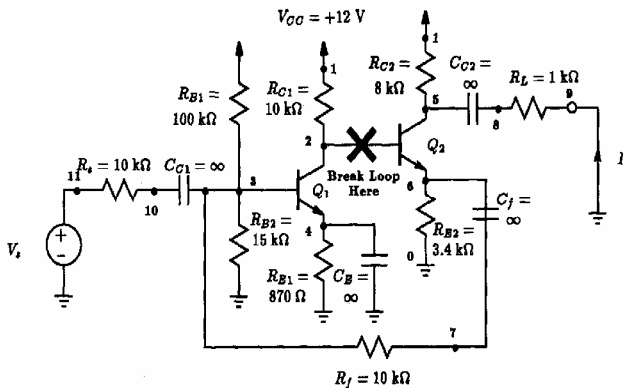


Figura 17. Amplificador realimentado.

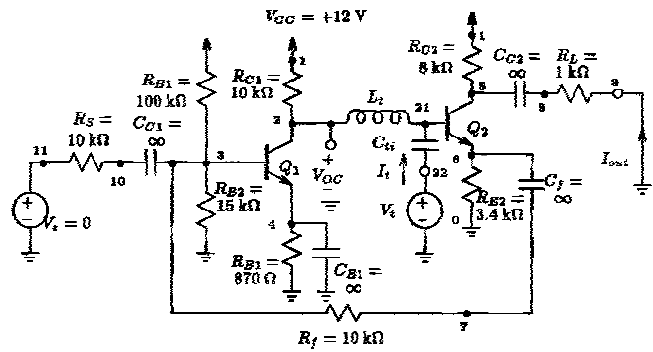


Figura 18. Apertura del Bucle de realimentación.

ESpice 6->print vm(22) vp(22) mag(i(Vt)) ph(i(Vt)) (7)

Index frequency vm(22) vp(22) vm(i(Vt)) vp(i(Vt)) vm(22)/mag(i(Vt)) (8)

| Index | frequency | vm(22) | vp(22) | vm(i(Vt)) | vp(i(Vt)) | vm(22)/mag(i(Vt)) |
|-------|-----------|---------|---------|-----------|-----------|-------------------|
| 0 | 1,0e+00 | 1,0e+00 | 0,0e-00 | 2,4e-06 | 1,8e+02 | 4,0e+05 |
| 1 | 2,0e+00 | 1,0e+00 | 0,0e-00 | 2,4e-06 | 1,8e+02 | 4,0e+05 |
| 2 | 3,0e+00 | 1,0e+00 | 0,0e-00 | 2,4e-06 | 1,8e+02 | 4,0e+05 |

La resistencia equivalente se puede obtener a través de (9) cuyo resultado se ha incluido en la última columna de la tabla (8):

ESpice 7->print vm(22)/mag(i(Vt)) (9)

Una vez calculada la resistencia equivalente $R_f = 402.8 \text{ K}\Omega$ se puede insertar en el circuito a través de un condensador de desacoplo C_{to} , que no modifique el punto de operación del circuito, ver figura 19, donde podremos obtener la ganancia en bucle abierto mediante (10) ejecutando en el simulador (11):

$$A \cdot b = -\frac{V_o}{V_i} = 59,54 \quad (10)$$

ESpice 8->print vm(22) vp(22) vm(23) vp(22) v(23)/v(22) (11)

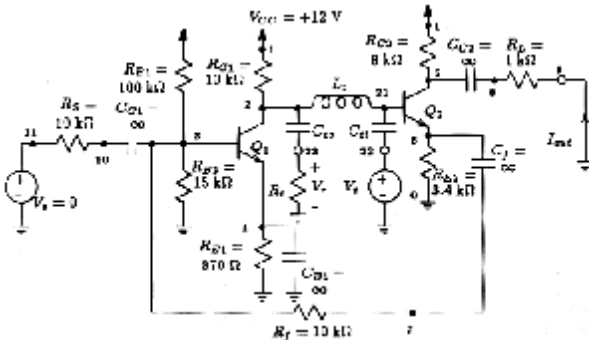


Figura 19. Amplificador con bucle abierto.

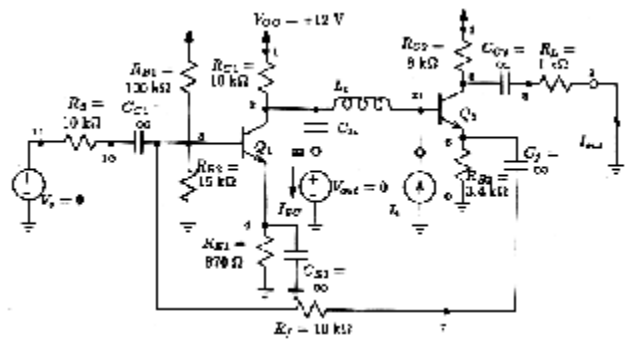


Figura 20. Método teórico de Rosenstark a $T_{S.C.}$

Un procedimiento alternativo y mucho más rápido para estudiar este mismo circuito sería utilizar el método de método de Rosenstark [8,9]. Este método permite obtener la ganancia del bucle de realimentación únicamente realizando dos ensayos al circuito del bucle abierto. El primero consiste en medir la ganancia en tensión de salida en circuito abierto $T_{O.C.}$ y el segundo, en medir la ganancia en corriente en corto circuito $T_{S.C.}$, utilizando en cada caso un generador de tensión y corriente como excitación. La expresión utilizada para calcular $A \cdot b$ se presenta en (12):

$$A \cdot b = -T_{O.C.} \| T_{S.C.} = -\frac{1}{\frac{1}{T_{O.C.}} + \frac{1}{T_{S.C.}}} \quad (12)$$

```

Dual Common Emitter Stages amplifier
* input source set to zero for output impedance calculation
Vs 11 0 DC 0 AC 0
Rs 11 10 10k
* short-circuit output port
Vout 0 9 0
* power supply
Vcc 1 0 DC 12V
* amplifier circuit
* 1st stage
Rc1 1 2 10k
Q1 2 3 4 Q2N3904
Re1 4 0 870
Ce1 4 0 1GF
Rb1 1 3 100k
Rb2 3 0 15k
* 2nd stage
Rc2 1 5 8k
Q2 5 21 6 Q2N3904
* decoupling capacitors
Cc1 10 3 1GF
Cc2 5 8 1GF
* load
Rl 8 9 1k
* feedback circuit
Re2 6 0 3.4k
Rf 3 7 10k
Cf 6 7 1GF
* inject signal into feedback loop without disturbing DC bias
Lt 2 21 1GH
Lti 21 25 0GH
Cti 25 22 1GF
Vt 22 0 AC 1V DC 0C
It 0 21 AC 0A DC 0A
Lto 2 24 1GH
Cto 23 24 1GF
Vsc 23 0 DC 0 AC 0
.model Q2N3904 NPN (level=1 Is=6.734f Xti=3 Eg=1.11
+ Vaf=74.03 Bf=416.4 Ne=1.259 Ise=6.734f Ikf=66.78m
+ Xtb=1.5 Br=.7371 Nc=2 Isc=0 Ikf=0 Rc=1
+ Cjc=3.638p Mjc=.3085 Vjc=.75 Fc=.5 Cje=4.493p
+ Mje=.2593 Vje=.75 Tr=239.5n Tf=301.2p ltf=.4
+ Vtf=4 Xtf=2 Rb=10)
.control
let T_open_cir = (0, 0)
let T_short_cir = (0, 0)
ac lin 3 1Hz 3Hz
echo "Medida Tension salida en Circuito Abierto Voc"
echo "Lti[A] = $&@Lti[inductance]"
echo "Lto[H] = $&@Lto[inductance]"
echo "It[A] = $&@It[dc]"
print vm(22) vp(22) mag(i(Vt)) ph(i(Vt)) vm(22)/mag(i(Vt))
let T_open_cir = v(2)[0]
echo "T_open_circuit=$&T_open_cir"
echo "Medida Corriente salida en corto Circuito Isc"
alter @It[ac] = 1A
alter @Lti[inductance] = 1GH
alter @Lto[inductance] = 0GH
echo "Lti[H] = $&@Lti[inductance]"
echo "Lto[H] = $&@Lto[inductance]"
echo "It[A] = $&@It[acmag]"
ac lin 3 1Hz 3Hz
print vm(i(Vsc)) vp(i(Vsc))
let T_short_cir = i(Vsc)[0]
echo "T_short_circuit=$&T_short_cir"
echo "Medida Corriente salida en corto Circuito Isc"
print mag(-1/(1/T_open_cir+1/T_short_cir))
unlet let T_short_cir let T_open_cir
.endc
.end

```

La ventaja de este método es que $T_{O.C}$ y $T_{S.C}$ pueden ser calculados para un rango de frecuencias sin necesidad cambiar el circuito o la resistencia equivalente R_t del procedimiento anterior. Haciendo uso del circuito de la figura 20 y cambiando dinámicamente mediante el lenguaje de *script* del simulador podemos añadir los elementos V_{SC} , L_{to} , C_{to} al nodo 2 y V_t , C_{ti} , L_{ti} al nodo 21 y obtener mediante una única simulación las dos ganancias en tensión y corriente ($T_{O.C} = -60,86$ y $T_{S.C} = -2753$) que a través de (12) nos permite obtener el mismo valor $A \cdot b = -T_{O.C} \parallel T_{S.C} = 59,54$.

ESpice 9->print vm(22) vp(22) vm(2) vp(2) v(23)/v(22)(13) (14)

El uso de este procedimiento orientado al análisis en un gran ancho de banda y la capacidad de ESPICE de interpretar comandos que permiten variar los valores de los componentes que conforman la topología de un circuito permiten simplificar los cálculos y reducir la manipulación necesaria de los archivos de descripción de circuitos o esquemáticos.

5. Conclusiones

El simulador didáctico presentado, ESPICE, reúne una serie de características que lo hacen ideal para su utilización en entornos docentes y permite, además, acercar al alumno a los procesos involucrados en la simulación y diseño de circuitos electrónicos con herramientas *TCAD*. El acceso a variables internas del simulador facilita la comprensión de los cálculos involucrados en una simulación. La gran cantidad de ejemplos resueltos incluidos en la distribución de la herramienta, el acceso al código fuente actualizado para poder compilarlo, el sistema de control de versiones y errores, y el sitio WEB de ayuda y de introducción al uso hacen de ESPICE una aplicación de referencia, en castellano, entre otros muchos idiomas, que puede ayudar a la comunidad educativa iberoamericana en sus tareas docentes.

Referencias

- [1] Laurence W. Angel, *The Life of SPICE. Bipolar and Circuits and Technology Meeting*, 1996, Mineapolis
- [2] MinGW - Minimalist GNU for Windows. URL: <http://www.mingw.org/>
- [3] SUBVERSION - Open Source Software Engineering Tools. URL: <http://subversion.tigris.org/>
- [4] MSYS - POSIX-like command line interpreter system. URL: <http://www.mingw.org/msys.shtml>
- [5] TRAC- Trac is an enhanced wiki and issue tracking system for software development project. URL: <http://trac.edgewall.org/>
- [6] Paul W. Tuinenga. *Spice A guide to Circuit simulation and analysis using PSPICE*. Prentice Hall. (1991)
- [7] D. Pozar. *Microwave Engineering 3rd ed.* John Wiley & Sons. New York (2005)
- [8] G. Roberts, A. Sedra. *Spice 2nd ed.* Oxford University Press, (1996)
- [9] S. Rosenstark. *Feedback Amplifiers Principles*. Macmillan Publishing. New York (1986)