

Tutoriales Síntesis de Redes Activas

Laboratorios SRA



Universidad Nacional de Córdoba
Facultad de Ciencias Exactas, Físicas y Naturales
Escuela de Ingeniería Electrónica

15 de diciembre de 2023

- 1 Importar un modelo de un amplificador operacional
- 2 Exportar datos de simulación para graficar en Python
- 3 Múltiples run de un circuito
- 4 Gráficos de circuitos con Python
- 5 Simulación de Montecarlo

En esta presentación se tratarán temas correspondientes al uso de LTSpice para aplicaciones en Síntesis de Redes Activas. Para ello, se mostrarán diferentes tutoriales que permiten comprender el funcionamiento del software y de algunas características que facilitan tanto el análisis como el diseño de circuitos.

Agregar un amplificador operacional rápidamente (I)

Si se desea utilizar un Opamp que no esta en la libreria de LTspice, se puede realizar el siguiente procedimiento para agregar el componente al Schematic (no lo agrega a la libreria): En primer lugar, se debe buscar el modelo spice del componente, por ejemplo el LM324. Guardar el archivo en formato .txt y pegarlo en la carpeta de LTspice, o donde sea que se guarde el Schematic en cuestión.

```
LM324: Bloc de notas
Archivo Edición Formato Ver Ayuda
* CREATED USING PARTS RELEASE 4.01 ON 09/08/89 AT 10:54
* (REV N/A) SUPPLY VOLTAGE: 5V
* CONNECTIONS: NON-INVERTING INPUT
*              | INVERTING INPUT
*              | | POSITIVE POWER SUPPLY
*              | | | NEGATIVE POWER SUPPLY
*              | | | | OUTPUT
*              | | | | |
.SUBCKT LM324 1 2 3 4 5
*
C1 11 12 5.544E-12
C2 6 7 20.00E-12
DC 5 53 DX
DE 54 5 DX
DLP 90 91 DX
DLN 92 90 DX
DP 4 3 DX
EGND 99 0 POLY(2) (3,0) (4,0) 0 .5 .5
FB 7 99 POLY(5) VB VC VE VLP VLN 0 15.91E6 -20E6 20E6 20E6 -20E6
GA 6 0 11 12 125.7E-6
GCH 0 6 10 99 7.067E-9
IEE 3 10 DC 10.04E-6
HLIM 90 0 VLIM 1K
Q1 11 2 13 QX
Q2 12 1 14 QX
R2 6 9 100.0E3
RC1 4 11 7.957E3
RC2 4 12 7.957E3
RE1 13 10 2.773E3
RE2 14 10 2.773E3
REE 10 99 19.92E6
RO1 8 5 50
RO2 7 99 50
RP 3 4 30.31E3
VB 9 0 DC 0
VC 3 53 DC 2.100
VE 54 4 DC .6
VLIM 7 8 DC 0
VLP 91 0 DC 40
VLN 0 92 DC 40
.MODEL DX D(TS=800.0E-18)
.MODEL QX PNP(TS=800.0E-18 BF=250)
```

Figura 1: Modelo SPICE de LM324

Agregar un amplificador operacional rápidamente (II)

Lo siguiente es, dentro del Schematic, tocar el boton Component(F2), buscar el componente opamp2.

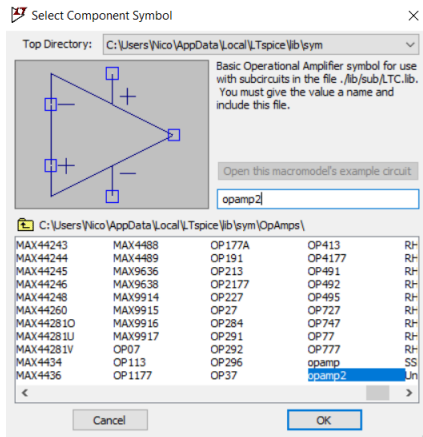


Figura 2: Selección del componente opamp2

Agregar un amplificador operacional rápidamente (III)

Seleccionado el componente, se lo agrega al Schematic y sobre el mismo, hacer click derecho. Se abrirá una pestaña con los atributos del componente.

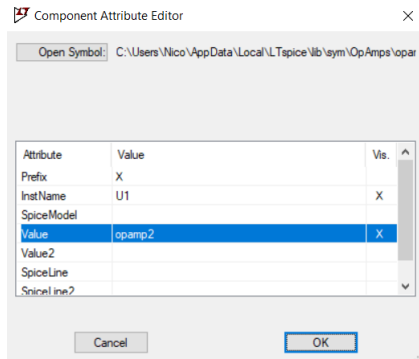


Figura 3: Atributos del componente

Agregar un amplificador operacional rápidamente (IV)

Hacer doble click en value, borrar el contenido y escribir el nombre del Opamp, en este caso, LM324. Dar click en OK y ya aparecera el componente con el nombre del Opamp deseado. Por ultimo lo que se debe hacer para una correcta simulación es presionar la tecla 'S' para que abra el menu de SPICE Directive, una vez dentro, escribir .include y el nombre del componente, para el caso del ejemplo seria.

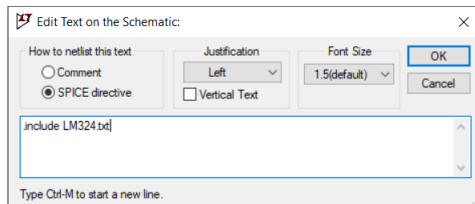


Figura 4: SPICE directive para incluir el componente al Schematic

Exportar datos a Python (I)

Luego de ejecutar un run, al hacer click derecho sobre el plot, se aprecia la siguiente ventana. Allí, se debe ir a File/Export data as text y se abrirá una ventana para guardar el archivo.

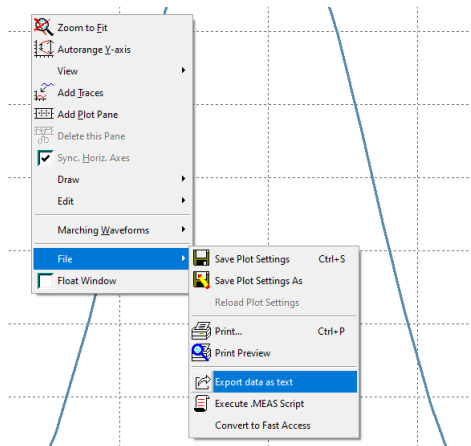


Figura 5: Cuadro de diálogo para exportar los datos de la simulación

Exportar datos a Python (II)

Al hacer click, aparece un cuadro de diálogo nuevo, en donde se puede seleccionar la ruta de destino del archivo. Se pide seleccionar un nombre, y una serie de datos para guardar.

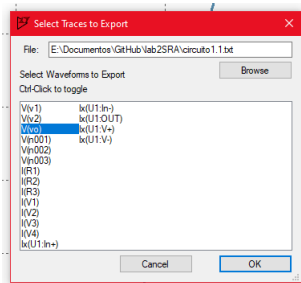
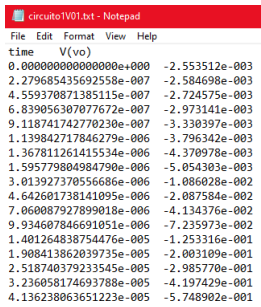


Figura 6: Selección de la ruta de destino de los datos

Exportar datos a Python (III)

Una vez seleccionada la ruta, el archivo de salida tiene un formato como el siguiente. Se puede observar que la primera columna corresponde al tiempo, y las siguientes a los valores de las variables que se hayan seleccionado para guardar.



```
File Edit Format View Help
time V(vo)
0.000000000000000e+000 -2.553512e-003
2.279685435692558e-007 -2.584698e-003
4.559370871385115e-007 -2.724575e-003
6.839056307077672e-007 -2.973141e-003
9.118741742770230e-007 -3.330397e-003
1.139842717846279e-006 -3.796342e-003
1.367811261415534e-006 -4.370978e-003
1.595779804984790e-006 -5.054303e-003
3.013927370556686e-006 -1.086028e-002
4.642601738141095e-006 -2.087584e-002
7.060087927899018e-006 -4.134376e-002
9.934607846691051e-006 -7.235973e-002
1.401264838754476e-005 -1.253316e-001
1.908413862039735e-005 -2.003109e-001
2.518740379233545e-005 -2.985770e-001
3.236058174693788e-005 -4.197429e-001
4.136238063651223e-005 -5.748902e-001
```

Figura 7: Formato de los datos de salida de LTSpice

Exportar datos a Python (IV)

Una vez realizados los pasos anteriores, para poder graficar datos con Python es necesario tener instaladas las librerías de Numpy, Pandas y Matplotlib. Para ello, con Python instalado, se instalan las dependencias con los comandos siguientes en la consola.

Código 1: Instalación de dependencias para Python

```
1 pip install numpy
2 pip install pandas
3 pip install matplotlib
```

Exportar datos a Python (V)

Dentro de un editor de texto, se debe crear un archivo con extensión .py, y se debe importar las librerías necesarias. Luego, se debe leer el archivo de datos, y se debe crear un DataFrame con los datos leídos. Por último, se debe graficar los datos con Matplotlib. Para ello, atendiendo al formato en que se exporta la información con LTSpice, se puede armar un script como el siguiente.

Código 2: Instalación de dependencias para Python

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 data = pd.read_csv('E:\Documentos\GitHub\lab2SRA\circuito1V01.txt',delimiter='\'
    ↪ t')
5 plt.plot(data['time'],data['V(vo)'])
6 plt.title('Salida con V1 activada y V2 pasivada')
7 plt.ylabel('Vo [V]')
8 plt.xlabel('Tiempo [seg]')
9 plt.grid(True)
10 plt.show()
```

Exportar datos a Python (VI)

Del código anterior, se obtiene una salida como la siguiente.

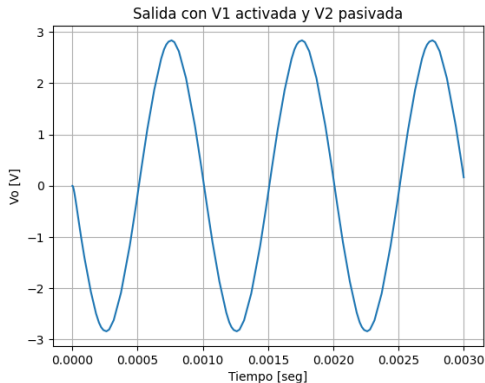


Figura 8: Ploteo de datos con Python

Múltiples run de un circuito (I)

Para simular repetidas veces un circuito en LTSpice, por ejemplo, para variar el valor de una fuente de tensión, a fin de analizar un rechazo al modo común. Se puede usar una directiva que permite ahorrar tiempo. Para esto, es necesario primero, determinar el valor a barrer, y en el campo del simulador en donde deba ser utilizado, hay que ingresar un nombre para la variable, llámese X , delimitado por llaves de apertura y cierre $\{ \}$.

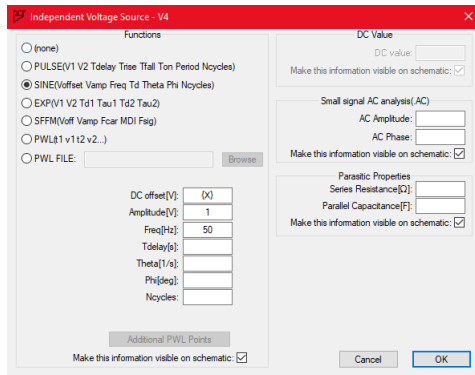


Figura 9: Uso de una variable en LTSpice

Múltiples run de un circuito (II)

Una vez hecho esto, se debe ir a la pestaña de SPICE Directive, y escribir `.step param X a b c`, donde **a** es el valor inicial, **b** el valor final, y **c** el paso. Luego, se debe hacer click en OK, y se debe ejecutar el run.

.step Statement Editor

.step is used to overlay simulation results while sweeping user-defined parameters.

Name of parameter to sweep:

Nature of sweep:

Start value:

Stop value:

Increment:

Syntax: `.step param <Name> <Start Value> <Stop Value> <Increment>`

Figura 10: SPICE directive para realizar múltiples run

Múltiples run de un circuito (III)

Del uso de la directiva anterior, puede obtenerse un plot como el siguiente. En el mismo, se puede observar que se realizaron varios runs, con valores de tensión de entrada de que van de 10 V a 30 V con un paso 0.75.

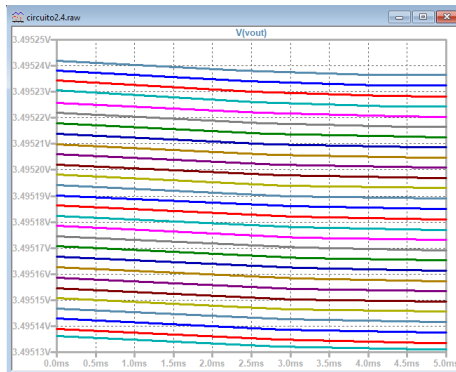
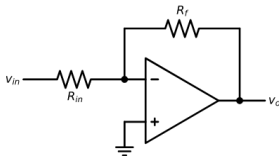


Figura 11: Ploteo de datos con directiva .step

Gráficos de circuitos con Python (I)

Con Python y la librería Schemdraw es posible realizar gráficos que tienen un look minimalista, así como también son cómodos de manipular. Para ello es necesario instalar la librería con el comando (`pip install Schemdraw`). A modo de ejemplo, se puede poner un código para un amplificador inversor.

Primero, se crea un nuevo dibujo y se añade un op-amp. Luego, se dibuja una línea hacia abajo desde la entrada inversora del op-amp y se añade un símbolo de tierra. Se añade una resistencia de entrada a la entrada no inversora del op-amp, y se dibuja una línea hacia arriba desde allí. Se añade una resistencia de retroalimentación desde la salida del op-amp, y se dibuja una línea hasta la salida del op-amp y se añade un punto. Finalmente, se dibuja una línea hacia la derecha desde la salida del op-amp y se añade una etiqueta. El resultado es un diagrama de un amplificador operacional con resistencias de entrada y retroalimentación, con etiquetas en los componentes y los puntos de entrada y salida.



Gráficos de circuitos con Python (II)

El circuito mostrado en la filmina anterior se logra con el uso del siguiente código.

Código 3: Instalación de dependencias para Python

```
1 import matplotlib.pyplot as plt
2 import schemdraw
3 import schemdraw.elements as elm
4 with schemdraw.Drawing() as d:
5     op = elm.Opamp(leads=True)
6     elm.Line().down(d.unit/4).at(op.in2)
7     elm.Ground(lead=False)
8     Rin = elm.Resistor().at(op.in1).left().idot().label('$R_{in}$', loc='bot').label('$v_{\{
9         \hookrightarrow in\}$', loc='left')
10    elm.Line().up(d.unit/2).at(op.in1)
11    elm.Resistor().tox(op.out).label('$R_f$')
12    elm.Line().toy(op.out).dot()
13    elm.Line().right(d.unit/4).at(op.out).label('$v_{o}$', loc='right')
```

Método de Montecarlo (I)

Para realizar una simulación de Montecarlo en LTSpice XVII es necesario tener la librería estándar que viene en con la instalación del programa. Los pasos a seguir son los siguientes.

Primero hay que preparar el esquemático. Habrá que determinar que tipo de simulación se quiere realizar (temporal o respuesta en frecuencia).

```
.tran 100m  
;ac dec 100m 0.1m 100Meg
```

Figura 13: SPICE directive para Método de Montecarlo

Método de Montecarlo (II)

Ademas con click derecho sobre las fuentes, las configuraremos para determinar si son de alimentación o es una señal.

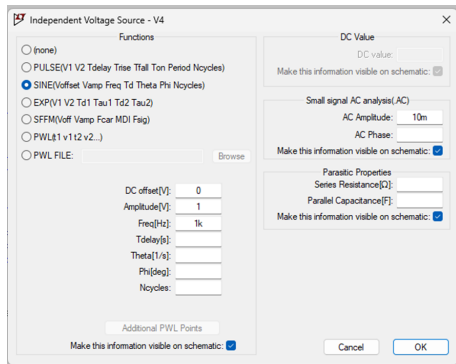


Figura 14: Configuración de una fuente de tensión

Método de Montecarlo (III)

El esquemático de ejemplo es el siguiente.

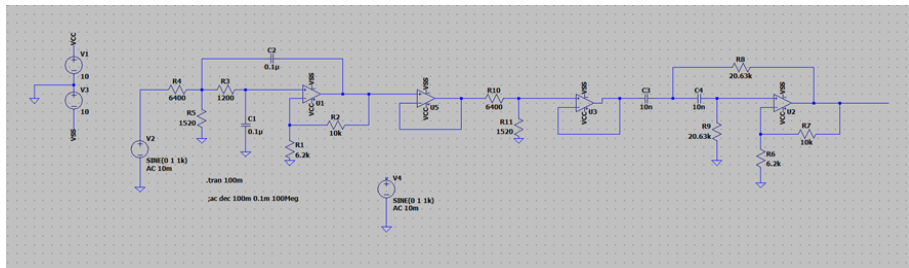


Figura 15: Esquemático de ejemplo

Método de Montecarlo (IV)

Para verificar las simulaciones, se tiene la siguiente respuesta temporal.

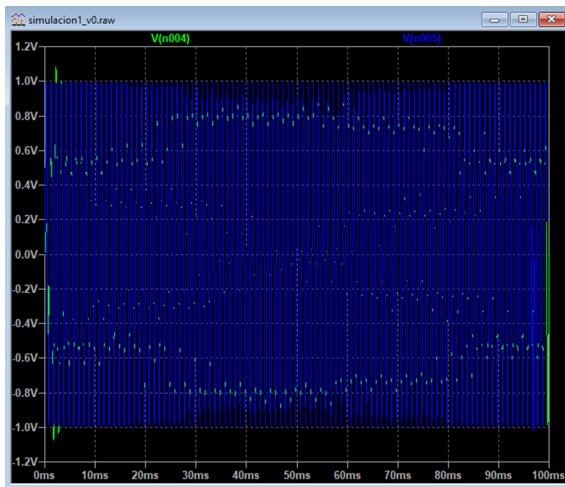


Figura 16: Respuesta temporal

Método de Montecarlo (V)

Una vez realizado el esquemático, se utiliza la función mc (Monte Carlo). Además se define un parámetro que será la tolerancia de la resistencia.

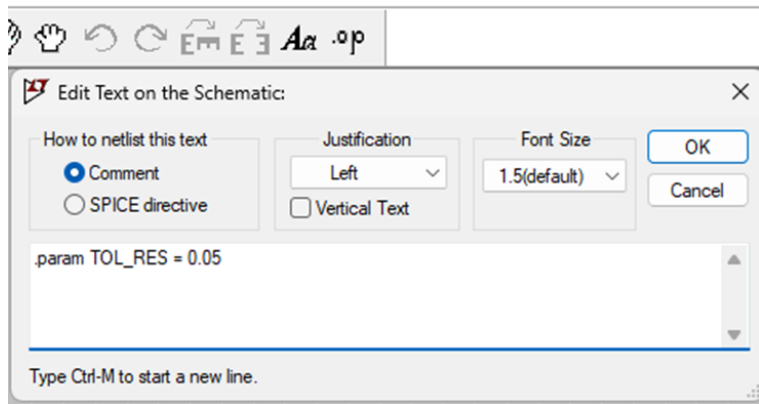


Figura 17: Configuración de la directiva

Método de Montecarlo (VI)

Además, se define la tolerancia de los capacitores, en este caso del 10 %. Con estos parámetros definidos, se realiza la siguiente asignación a los valores de la resistencia mediante la función mc. En donde la función es: {mc(valor del componente,tolerancia)}

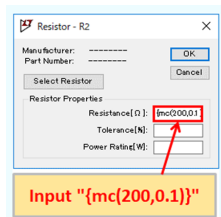


Figura 18: Uso de la directiva .mc

Método de Montecarlo (VII)

Con el comando SPICE directive, se escribe lo siguiente.

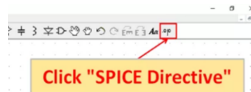


Figura 19: Configuración de la directiva .mc

Método de Montecarlo (VIII)

Tenemos que tener en cuenta que stop value es la cantidad de simulaciones que realizamos. Finalmente realizando el análisis de Montecarlo para el análisis temporal obtenemos.

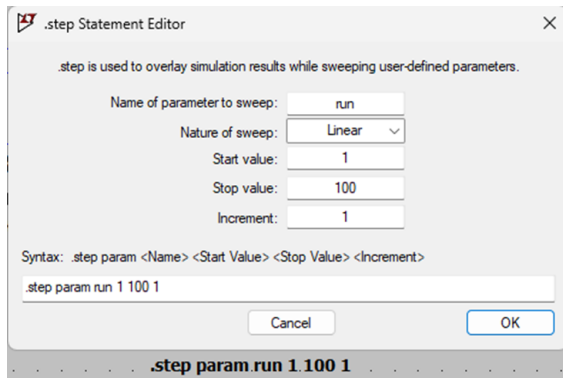


Figura 20: Simulaciones con distintos pasos (Monte Carlo)

Gracias por su atención