Síntesis de Redes Activas

Cálculo simbólico con Python

Britez, Fabio - Corvalán, Abel
Ing. Ferreyra, Pablo Alejandro

Facultad de Ciencas Exactas Físicas y Naturales - UNC

Introducción

El **cálculo simbólico** en **Python** se implementa para trabajar con expresiones algebraicas y funciones matemáticas en la programación.

Objetivos

- Aplicar el **cálculo simbólico** para la resolver ecuaciones de los circuitos propuestos en la materia Síntesis de Redes Activas.
- Implementar librerías necesarias para extender las capacidades del lenguaje con funciones y métodos predefinidos.
- Obtener expresiones matemáticas en formato LaTeX con Python.

Recomendación

Implementar el entorno de **Jupyter Notebook** para una estructura de cálculo en conjunto con representaciones y anotaciones.

Preparación previa - Instalar librerías

Para instalar librerías **Sympy**, **Numpy**, **Math** debemos:

1. Abrir un terminal **cmd**.
2. Colocar la siguiente línea de código:

cmd

```
pip install sympy numpy ipython
```

Preparación previa - Cargar librerías

Siempre debemos cargar las librerías que usaremos en nuestro código con la siguiente estructura:

Python

```
import "nombre de la librería" as "expresión"
```

Si queremos implementar **sympy** lo haremos de la siguiente forma:

```
1 import sympy as sym
```

De esta forma cuando usemos un método de librería escribiremos **sym** al principio de cada llamado.

Circuito de estudio

En base al circuito que se muestra en la figura implementaremos las diferentes funcionalidades de las librerías.

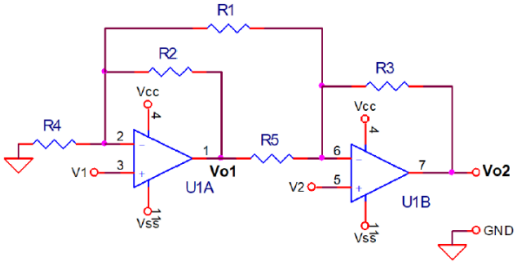

Figure: Circuito n° 1 - Laboratorio n° 1

Items a desarrollar

1. Cargar imágenes
2. Análisis del circuito
3. Agregar variables simbólicas
4. Escribir ecuaciones
5. Obtener miembro de la ecuación
6. Igualar expresiones
7. Despejar variable
8. Formatos de visualización de ecuaciones
9. Reemplazar variables
10. Resolución del circuito completo

Cargar imágenes

Para **cargar imágenes** en nuestro programa de Python se implementa la siguiente función:

Función

Image(filename= "img/nombre de la imagen")

Ejemplo:

```
1 Image(filename= "img/Circuito 1.png")
```

En el campo "filename" se carga la ruta de la imágen.

Expresión de salida del circuito

Se desea obtener la respuesta de circuito respecto a V_1 y V_2.

$$\boxed{V_o = V_1' + V_2'}$$

Análisis del circuito (I)

Se debe aplicar el teorema de superposición, por lo que se pasivan las fuentes V_1 y V_2 alternativamente.

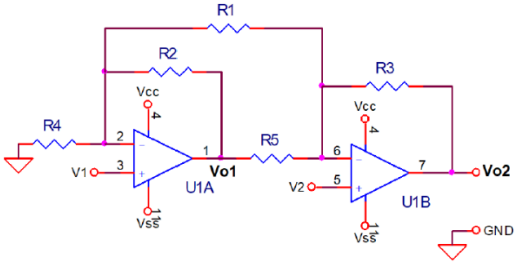

Figure: Circuito n° 1 - Laboratorio n° 1

Análisis del circuito (II)

Se estudia el caso con las siguientes condiciones:

$$Caso\,1: V_1 \neq 0V \quad V_2 = 0V$$

$$Caso\,2: V_1 = 0V \quad V_2 \neq 0V$$

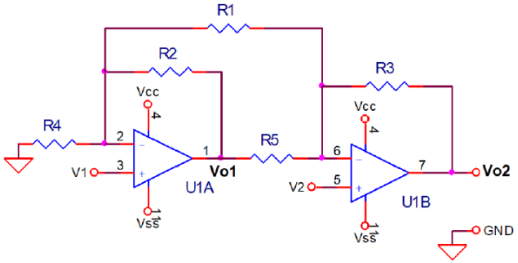

Figure: Circuito n° 1 - Laboratorio n° 1

Análisis del circuito - Caso 1 (I)

Se estudia el circuito con las siguientes condiciones:

$$V_1 \neq 0V$$

$$V_2 = 0V$$

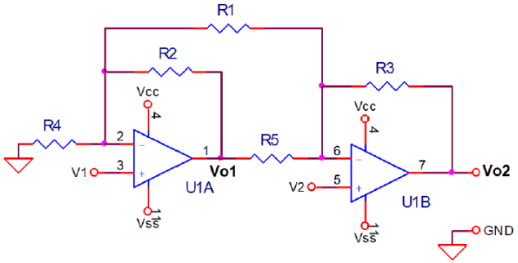

Figure: Circuito n° 1 - Laboratorio n° 1

Agregar variables simbólicas (I)

Se inspecciona el circuito y se detectan las variables V_1, V_2 y R_1, R_2, R_3, R_4, R_5. Para **agregar variables simbólicas** se escriben líneas de código con el siguiente formato:

Agregar variables simbólicas (II)

Función

"variable en Python" = sym.symbols('"símbolo"')

```
1 V1,V2= sym.symbols('V_{1},V_{2}')
2 R1,R2,R3,R4,R5= sym.symbols('R_{1},R_{2},R_{3},R_{4},R_{5}')
```

En el lado izquierdo se tiene el nombre de cada variable dentro del programa.
En el lado derecho se tiene el símbolo con el que se va a mostrar en la interfaz. Es importante escribir los símbolos del lado derecho en formato LaTeX.

Análisis del circuito - Caso 1 (II)

Las ecuaciones que modelan el amplificador para el caso 1 son las siguientes:

$$I_{R2} = \frac{V_{o1} - V_1}{R2}$$

$$I_{Rp} = \frac{V_1}{R_p}$$

Se tiene además que estas corrientes son iguales $I_{R2} = I_{Rp}$

Análisis del circuito - Caso 1 (III)

Se agregan variables que servirán para escribir las ecuaciones de corriente mencionadas anteriormente:

```
1 Rp,Ir2,Irp,Vo1= sym.symbols('R_{p},I_{R2},I_{Rp},V_{o1}')
2 s_Vo1,sVo1= sym.symbols('s_Vo1,sVo1')
```

Escribir ecuaciones (I)

Para **escribir ecuaciones simbólicas** se debe escribir con el siguiente formato de código:

Escribir ecuaciones (II)

Función

sym.Eq('"1er miembro", "2do miembro" ')

```
1 equ_Ir2= sym.Eq(Ir2,(Vo1-V1)/R2)
2 equ_Irp= sym.Eq(Irp,(V1/Rp))
```

En el lado izquierdo se tiene el nombre del objeto de tipo **ecuación**.
En el lado derecho se tienen los términos de nuesta ecuación.

Igualación de ecuaciones

Como se había menciondo I_{R2} y I_{Rp} son equivalentes, por lo que se debe obtener el segundo miembro de cada ecuación.

Obtener miembro de la ecuación (I)

Para **obtener un miembro de una ecuación** se programa con la siguiente estructura:

Obtener miembro de la ecuación (II)

Con **.lhs** se obtiene el término de la izquierda de la igualación, mientras que con **.rhs** se obtiene el término de la derecha.

Función

"nombre de la ecuación".lhs
"nombre de la ecuación".rhs

Igualar expresiones

Como se mencionó anteriormente se deben igualar las expresiones de I_{R2} e I_{Rp}.

```
1 equ1= sym.Eq(equ_Ir2.rhs, equ_Irp.rhs)
```

Esto es:

$$\frac{V_{o1} - V_1}{R_2} = \frac{V_1}{R_P}$$

Análisis del circuito - Caso 1 (IV)

Ahora se quiere despejar la variable V_{o1} de la ecuación **equ1**, para obtener la ganancia V_{o1}/V_1.

Despejar variable

Para despejar una variable se escribe la siguiente estructura:

Función

sym.solve("nombre de ecuación", "variable")

```
1 s_Vo1= sym.solve(equ1, Vo1)
```

Esta función devuelve la solución en un arreglo de tamaño n, siendo n la cantidad de soluciones.

Nueva ecuación

Se arma una nueva ecuación para la ganancia parcial V_{o1}/V_1 a partir de la solución obtenida anteriormente.

```
1 s_Vo1= sym.Eq(Vo1/V1, s_Vo1[0]/V1)
```

Formatos de visualización de ecuaciones (I)

Para **visualizar** las ecuaciones se pueden implementar varias alternativas. Estas son:

- Formato matemático en texto plano *sym.pprint*()
- Formato LaTeX *sym.print_latex*()
- Formato simbólico

Formatos de visualización de ecuaciones (II)

Se visualiza en **formato matemático en texto plano** con la función **pprint()**.

Función

sym.pprint("nombre de ecuación")

Formatos de visualización de ecuaciones (III)

Se visualiza en **formato LaTeX** con la función **print_latex()**.

Función

sym.print_latex("nombre de ecuación")

Formatos de visualización de ecuaciones (IIV)

Se visualiza en **formato simbólico** con la función **print_latex()**.

Función

"nombre de ecuación"

Para imprimir con este formato, debe escribirse el nombre de la ecuación al final del cuadro de código.

Reemplazar variables (I)

Se puede **reemplazar variables** simbólicas por valores numéricos o por otras expresiones simbólicas. Se llama a esta función de la siguiente forma:

Reemplazar variables (II)

Se hace el llamado al método **.subs()**.

Función

"nombre de ecuacion".subs("variable", "expresion o número")

Tenemos en nuestras ecuaciones que:

$$R_p = \frac{R_1 R_4}{R_1 + R_4}$$

Reeemplazar variables (III)

Para nuestro caso, queremos reemplazar la variable R_p por su equivalente $(R1//R4)$.

```
1 s_Vo1=sVo1.subs(Rp,(R1*R4)/(R1+R4))
```

Se obtiene la siguiente expresión para la ganancia:

$$\boxed{\frac{V_{o1}}{V_1} = \frac{R_1 + R_4}{R_1 R_4}\left(\frac{R_1 R_4}{R_1 R_4} + R_2\right)}$$

Análisis del circuito - Caso 1 (V)

Para obtener la expresión de la ganancia V_{o2}/V_{o1} se aplica LCK.

$$I_{R3} = I_{R5}$$

$$\frac{V_{o2}}{R_3} = \frac{-V_{o1}}{R_5}$$

Resolución ganancia V_{o2}/V_{o1} - Caso 1 (I)

Con el siguiente fragmento de código se obtiene la ganancia V_{o2}/V_{o1}.

```
# Cargo las variables particulares del caso de estudio.
Ir3, Ir5= sym.symbols('I_{R3}, I_{R5}')
Vo2= sym.symbols('V_{o2}')
s_Vo2, sVo2= sym.symbols('s_Vo2, sVo2')
# Escribo las ecuaciones
equ_Ir3= sym.Eq(Ir3, Vo2/R3)
equ_Ir5= sym.Eq(Ir5, -Vo1/R5)
# Igualo las expresiones Ir3 e Ir5.
equ2= sym.Eq(equ_Ir3.rhs, equ_Ir5.rhs)
# Resuelvo sistema, se obtiene solucion en formato matriz.
s_Vo2= sym.solve(equ2, Vo2)
# Obtengo solucion
sVo2= sym.Eq(Vo2/Vo1, s_Vo2[0]/Vo1)
```

Resolución ganancia V_{o2}/V_{o1} - Caso 1 (II)

```
#Formula en formato de texto plano.
sym.pprint(sVo2)
#Formula en formato latex
print('Formula en formato LaTex: ')
sym.print_latex(sVo2)
#Formula en formato simbolico
sVo2
```

Resolución ganancia V_{o2}/V_{o1} - Caso 1 (III)

Se obtiene la siguiente expresión para la ganancia V_{o2}/V_{o1}.

$$\boxed{\frac{V_{o2}}{V_{o1}} = -\frac{R_3}{R_5}}$$

Resolución - Caso 1 (I)

Se obtiene la ganancia del circuito para el caso 1 ($V_1 \neq 0V \;\; V_2 = 0V$)

$$\boxed{A_{V1} = -\frac{R_3\left(R_1R_4 + R_2\left(R_1 + R_4\right)\right)}{R_1R_4R_5}}$$

$$\boxed{V_o = -3V_1}$$

Análisis del circuito - Caso 2 (I)

Se estudia el circuito con las siguientes condiciones:

$$V_1 = 0V$$

$$V_2 \neq 0V$$

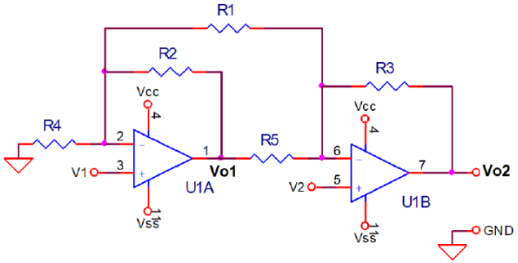

Figure: Circuito n° 1 - Laboratorio n° 1

Análisis del circuito - Caso 2 (II)

Las ecuaciones que modelan el amplificador para el caso 2 son las siguientes:

$$I_{R1} = -\frac{V_2}{R_1}$$

$$I_{R2} = \frac{V_{o1}}{R_2}$$

Se tiene además que estas corrientes son iguales $I_{R1} = I_{R2}$

Script - Caso 2 (I)

Se realiza el siguiente script:

```
# Cargo las variables particulares del caso de estudio.
Ir1, R= sym.symbols('I_{R1}, R')
# Escribo las ecuaciones de Ir1 e Ir2.
equ_Ir3= sym.Eq(Ir2, Vo1/R2)
equ_Ir5= sym.Eq(Ir1, -V2/R1)
# Igualo las expresiones Ir3 e Ir5.
equ2= sym.Eq(equ_Ir3.rhs, equ_Ir5.rhs)
sym.pprint(equ2)
# Resuelvo sistema, se obtiene solucion en formato matriz.
s_Vo1= sym.solve(equ2, Vo1)
# Obtengo solucion
sVo1= sym.Eq(Vo1, s_Vo1[0])
```

Script - Caso 2 (II)

Para la segunda etapa se tiene:

```
# Cargo las variables particulares.
Vx, Vo= sym.symbols('V_{x}, V_{o}')
# Nodo Vx es igual a la salida Vo1
Vx= sVo1.rhs
# Se aplica LCK
Vop= sym.Eq(((Vo2-V2)/R3), ((V2-Vx)/R5)+(V2/R1))
# Se normaliza el valor de las resistencias
s_Vo= Vop.subs({R1: R, R2: R, R3: R, R4: R, R5: R})
# Se despeja Vo.
sVo2= sym.solve(s_Vo, Vo2)
# Se arma la ecuaci n de Vo
s= sym.Eq(Vo, sVo2[0])
```

Resolución Caso 2 (I)

Se obtiene la siguiente expresión del circuito para el caso 2 ($V_1 = 0V \;\; V_2 \neq 0V$)

$$\boxed{V_o = 4V_2}$$

Resolución del circuito completo

Finalmente sumando las respuestas del circuito aplicando el teorema de superposición:

$$\boxed{V_o = -3V_1 + 4V_2}$$