name-project

lavelazquez

Apr 28, 2024

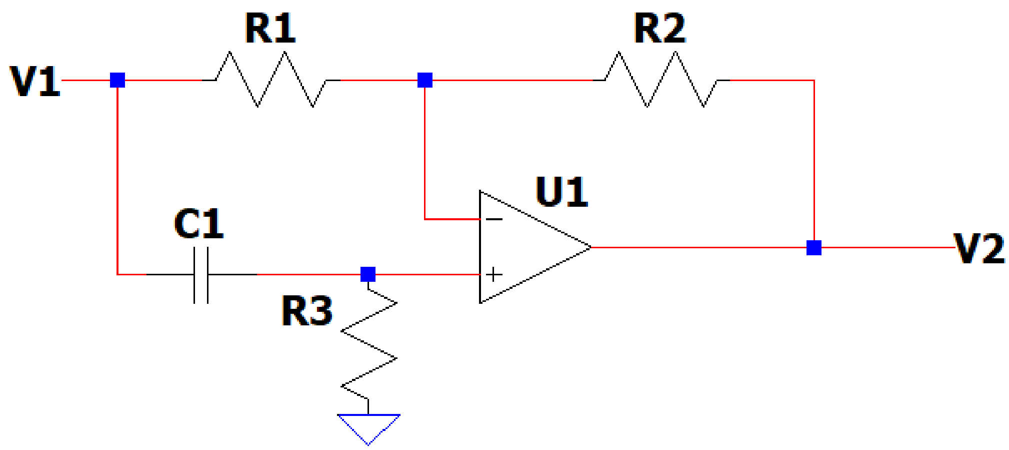
CONTENTS:

CHAPTER

ONE

TP-SEMANAL 1

Dado el siguiente circuito:



1. Obtener la función transferencia V2/V1 (módulo , fase y diagrama de polos y ceros).

Aplicando Norton:

$$\begin{aligned} V^{+}\left(G_3+SC_1\right)-V_1 SC_1=0 \\ V^{-}\left(G_2+G_1\right)-V_1 G_1-V_0 G_2=0 \end{aligned} \tag{1.1}$$

Asumiendo el OPAMP ideal y reordenando:

$$\begin{aligned} V^{+}=\frac{V_1 SC_1}{G_3+SC_1}=V^{-}=\frac{V_1 G_1+V_0 G_2}{G_2+G_1} \\ \frac{V_0}{V_i}=\frac{S-\frac{G_1}{G_2}\cdot\frac{G_3}{C}}{S+\frac{G_3}{C}}=\frac{S}{S+\frac{G_3}{C}}-\frac{G_1}{G_2}\cdot\frac{\frac{G_3}{C}}{S+\frac{G_3}{C}} \end{aligned} \tag{1.2}$$

Renombrando lo siguiente y proponiendolo como una suma de transferencias:

$$
\begin{aligned}
w_t &= \frac{G_3}{C} \\
K &= \frac{G_1}{G_2} \\
\frac{V_0}{V_i} &= \frac{S}{S + w_t} - K \cdot \frac{w_t}{S + w_t}
\end{aligned} \tag{1.3}
$$

2. Proponga una norma de impedancia y frecuencia de forma tal de llegar a una transferencia normalizada.

$$
\begin{aligned}
\$ &= \frac{S}{w_t} \\
\frac{V_0}{V_i} &= \frac{\$}{\$ + 1} - K \cdot \frac{1}{\$ + 1}
\end{aligned} \tag{1.4}
$$

Se obtiene la impedancia:

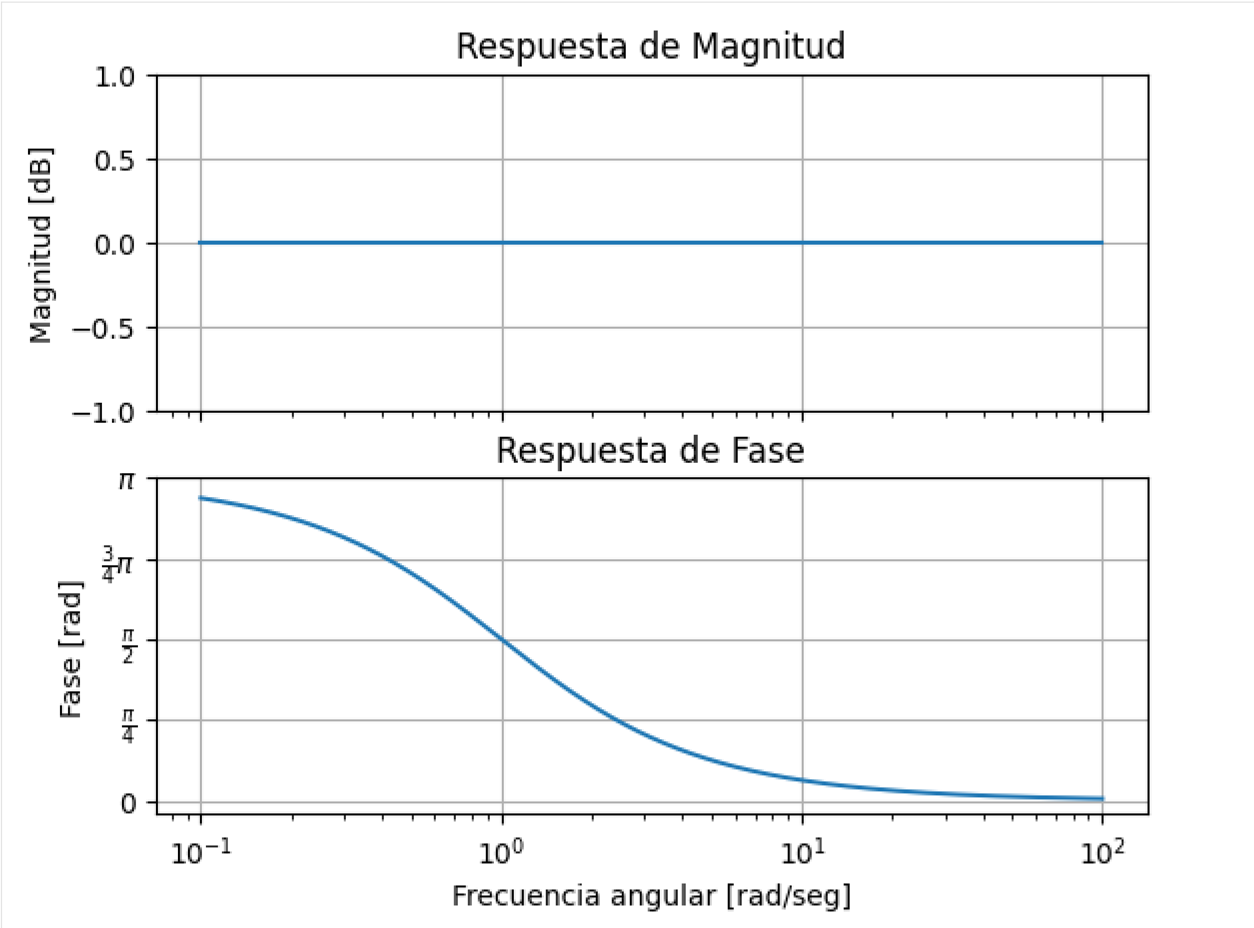
$$
\begin{aligned}
I_1 = I_a - I_b &= V^{+}\left(G_3 - G_1\right) + V_1 G_1 \\
V^{+} &= \frac{V_1 SC}{G_3 + SC} \\
Z &= \frac{1}{G_3} \frac{SC + G_3}{SC + G_1}
\end{aligned} \tag{1.5}
$$

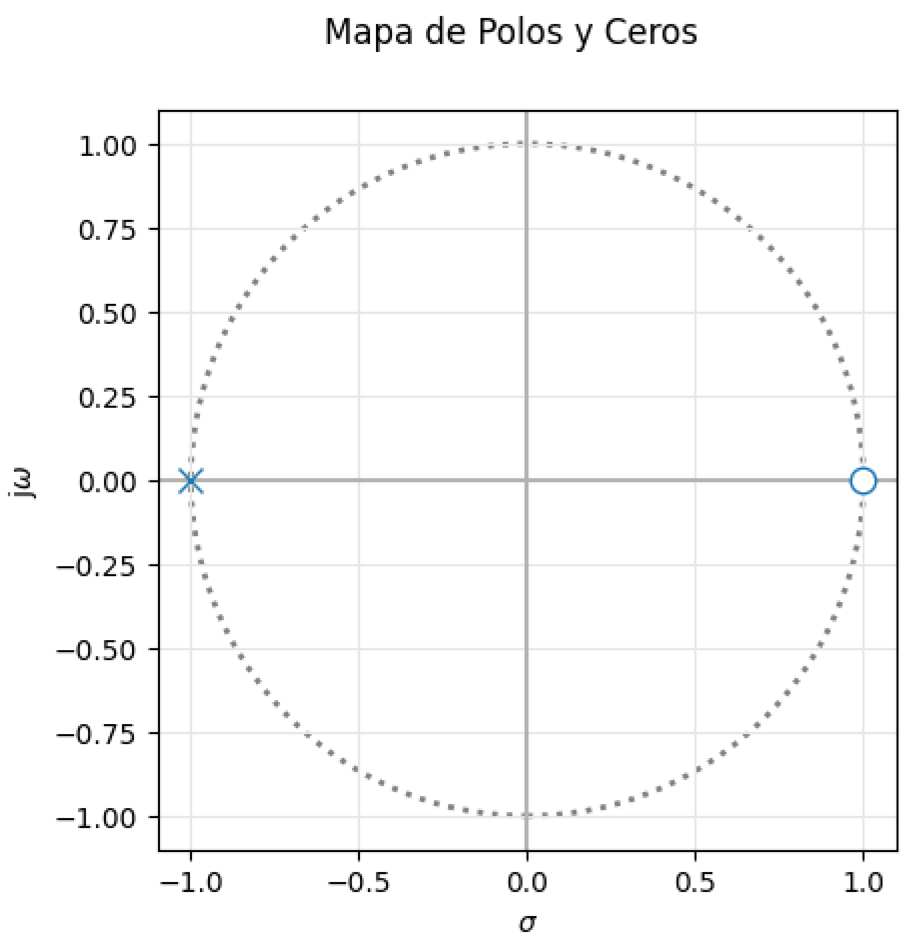
3. Simule la función transferencia normalizada en Python.

Se simulara con valores de K = [0, .5 , -.5]

[4]:
```python
import numpy as np
from scipy import signal as sig
from matplotlib import pyplot as plt

from pytc2.sistemas_lineales import bodePlot, pzmap, GroupDelay, analyze_sys

plt.figure(1)
plt.close(1)

num = np.array([ 1, -1])
den = np.array([ 1, 1])

H1 = sig.TransferFunction( num, den )

fig1, axs = bodePlot(H1)  # Obtener la figura y las subtramas

axs[0].set_ylim(-1, 1)  # Establecer límites del eje y en la subtrama 0

pzmap(H1)
```

[4]:
```
(2, <Axes: xlabel='$\\sigma$', ylabel='j$\\omega$'>)
```


Respuesta de Magnitud
Magnitud [dB]
1.0
0.5
0.0
−0.5
−1.0
Respuesta de Fase
Fase [rad]
π
3/4π
π/2
π/4
0
10^−1
10^0
10^1
10^2
Frecuencia angular [rad/seg]



[8]:
```
import sympy as sp
from IPython.display import display, Math
from pytc2.general import print_subtitle

V, Vi, Vo, G1, G2, G3, S, C1 = sp.symbols('V Vi Vo G1 G2 G3 S C1')

eq1 = V*(G3 + S*C1) - Vi*S*C1
eq2 = V*(G2 + G1) - Vi*G1 - Vo*G2

def get_Transfer(equations, V):

    sol = sp.solve(equations, V)
    T = sol[Vo]/sol[Vi]
    num, den = sp.fraction(T)
    coeficiente_mayor_grado = sp.Poly(den, S).all_coeffs()[0]

    num = sp.simplify(num/coeficiente_mayor_grado)
    den = sp.simplify(den/coeficiente_mayor_grado)

    return num/den
```

(continues on next page)

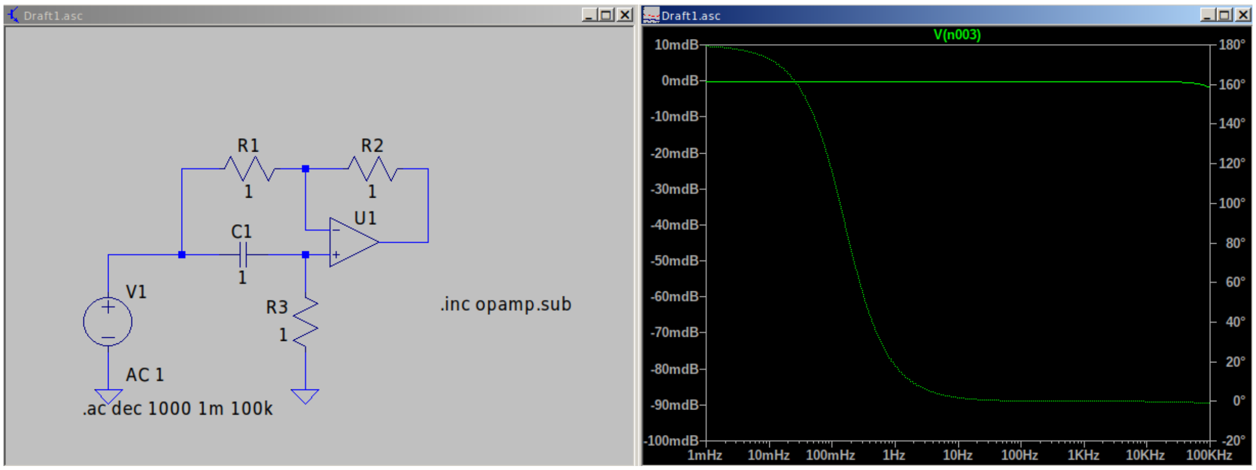
(continued from previous page)

```
print_subtitle('Transferencia')
display(Math( r' \frac{V_o}{V_i} = ' + sp.latex(get_Transfer((eq1, eq2), (Vo, Vi, V)))))
```

1.1 Transferencia

$$\frac{V_o}{V_i} = \frac{S - \frac{G_1 G_3}{C_1 G_2}}{S + \frac{G_3}{C_1}}$$

4. Simule la red normaliza en LTspice, y obtenga su respuesta en frecuencia



5. ¿Qué tipo de filtro es? ¿Qué utilidad podría tener este tipo de circuitos?

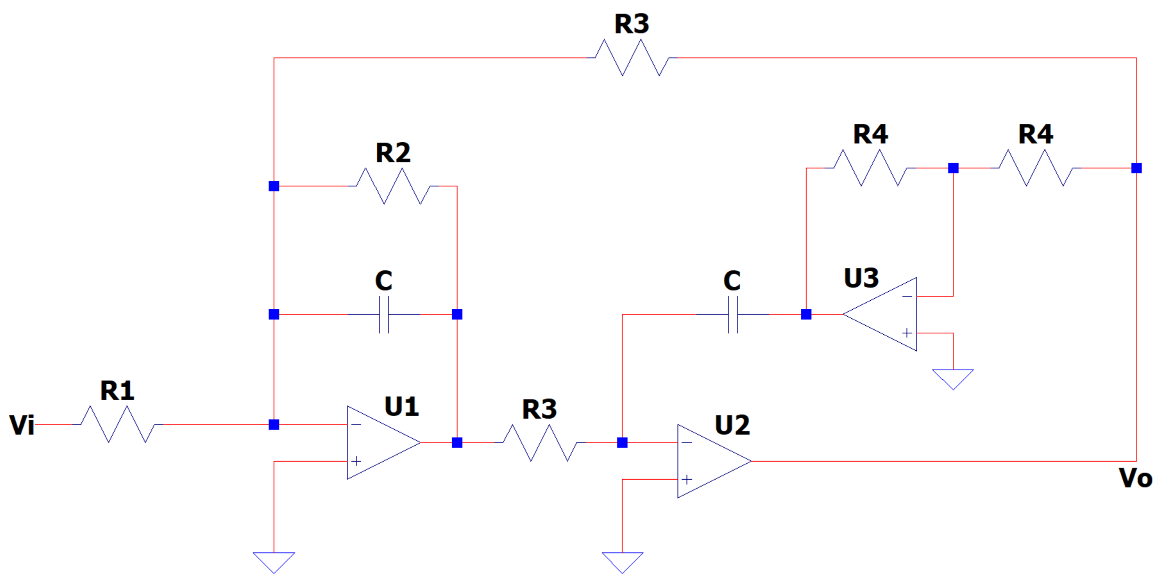
Es un filtro pasa todo que para señales de baja frecuencia le cambia la fase.

CHAPTER

TWO

TP-SEMANAL 2

2.1 Enunciado



Consignas de la actividad:

Hallar la transferencia T=VoVi en función de o y Q

.

Hallar los parámetros o y Q y k

.

Obtener el valor de los componentes para que el circuito se comporte como un Butterworth de 2do orden.

Cómo podría obtener un filtro pasabajo Butterworth de 4to orden, a partir de un prototipo basado en este circuito, y que cumpla con |T(0)|=20dB

.

Bonus:

```
+10  Obtener los valores de la red normalizados en frecuencia e impedancia.
+10   Simulación circuital de todos los experimentos.
```

(continues on next page)

(continued from previous page)

```
+10   Cómo podría obtener un circuito pasabanda con los mismos componentes originales y␣
↪con qué parámetros quedaría diseñado (Ver ejemplo 4.6 en Schaumann).
```

2.2 Obtencion de la T(S)

$$T(S) = \frac{V_0}{V_i}$$

Busco las ecuaciones con Norton:

$$-G_1 V_i - V_{01}(SC + G_2) - G_3 V_0 = 0 \quad (1)$$

$$-G_3 V_{01} - SCV_{03} = 0 \quad (2)$$

$$-G_4 V_{03} - G_4 V_0 = 0 \quad (3)$$

Obtengo de (3):

$$V_0 = -V_{03}$$

Reemplazando (3) en (2):

$$V_{01} = \frac{V_0 SC}{G_3}$$

Reemplazando en (1)

$$-G_1 V_i - \frac{V_0 SC}{G_3}(SC + G2) - G_3 V_0 = 0$$

Obtengo:

$$T(S) = -\frac{G_1}{G_3} \frac{\left(\frac{G_3}{C}\right)^2}{S^2 + S\frac{G_2}{C} + \left(\frac{G_3}{C}\right)^2} \quad (4)$$

Los parametros son:

$$w_0 = \frac{G_3}{C} \quad (5)$$

$$Q = \frac{G_3}{G_2} \quad (6)$$

$$K = \frac{G_1}{G_3} \quad (7)$$

Colocando (5), (6) y (7) en (4):

$$T(S) = \frac{-Kw_o^2}{S^2 + S\frac{w_0}{Q} + w_0^2} \quad (8)$$

Entonces normalizando en freq:

$$\$ = \frac{S}{w_0} \quad (9)$$

Reemplazando

$$\frac{V_0}{V_i} = \frac{-K}{\$^2 + \frac{\$}{Q} + 1} \quad (10)$$

La impedancia es:

$$Z = R_1 = \frac{1}{G_1} \quad (11)$$

Normalizo en impedancia:

$$Z' = \frac{Z}{R_1} = 1 \quad (12)$$

Agregando la normalizacion de impedancia (12) en (10):

$$T(S) = \frac{-1}{\$^2 + \frac{\$}{Q} + 1} \quad (13)$$

2.3 Los valores para conseguir un buttherworth de segundo orden son:

Dado que $Q = \frac{1}{2cos(\phi)}$, los polos para un segundo orden estan en $\pi/4$ y $-\pi/4$. Por lo tanto:

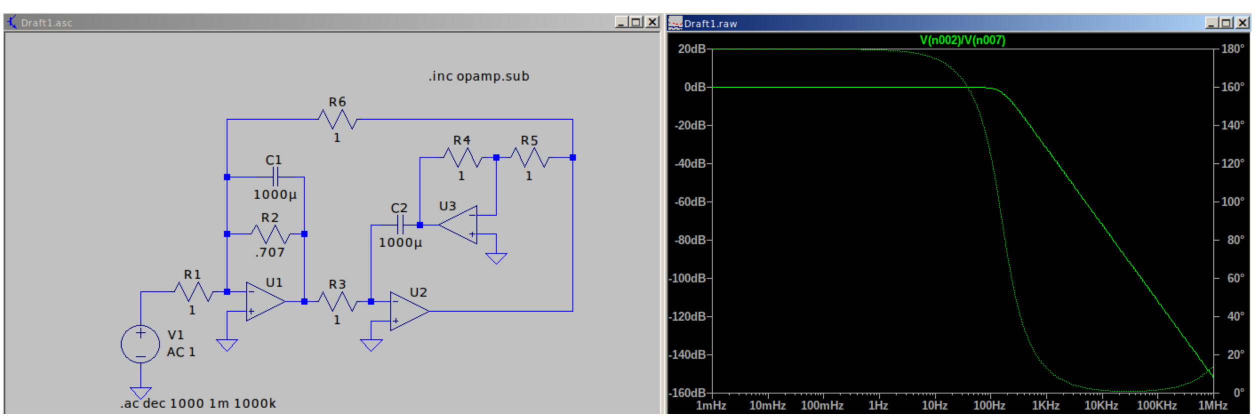
$$Q = \frac{1}{2cos(\pi/4)} = \frac{1}{\sqrt{2}} \quad (14)$$

Reemplazando (14) en (13):

$$T(S) = \frac{-1}{\$^2 + \$\sqrt{2} + 1} \quad (15)$$

Entonces para hacer que (13) satisfaga (15) y de la ecuacion (6), dado que G_3 aparece en todos lados, se establece $G_3 = 1$ y entonces $G_2 = 1/\sqrt{2}$

2.3.1 Simulacion

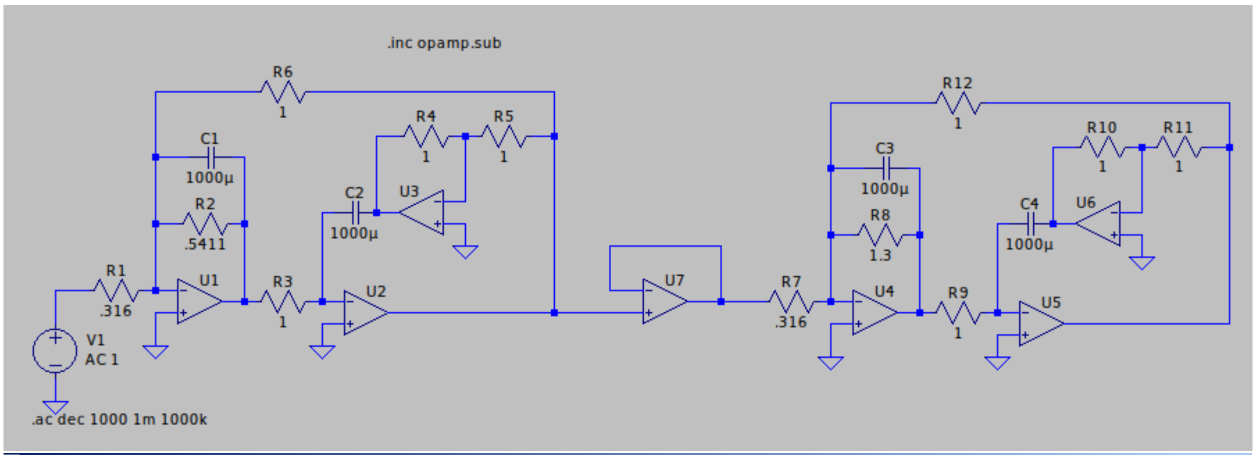


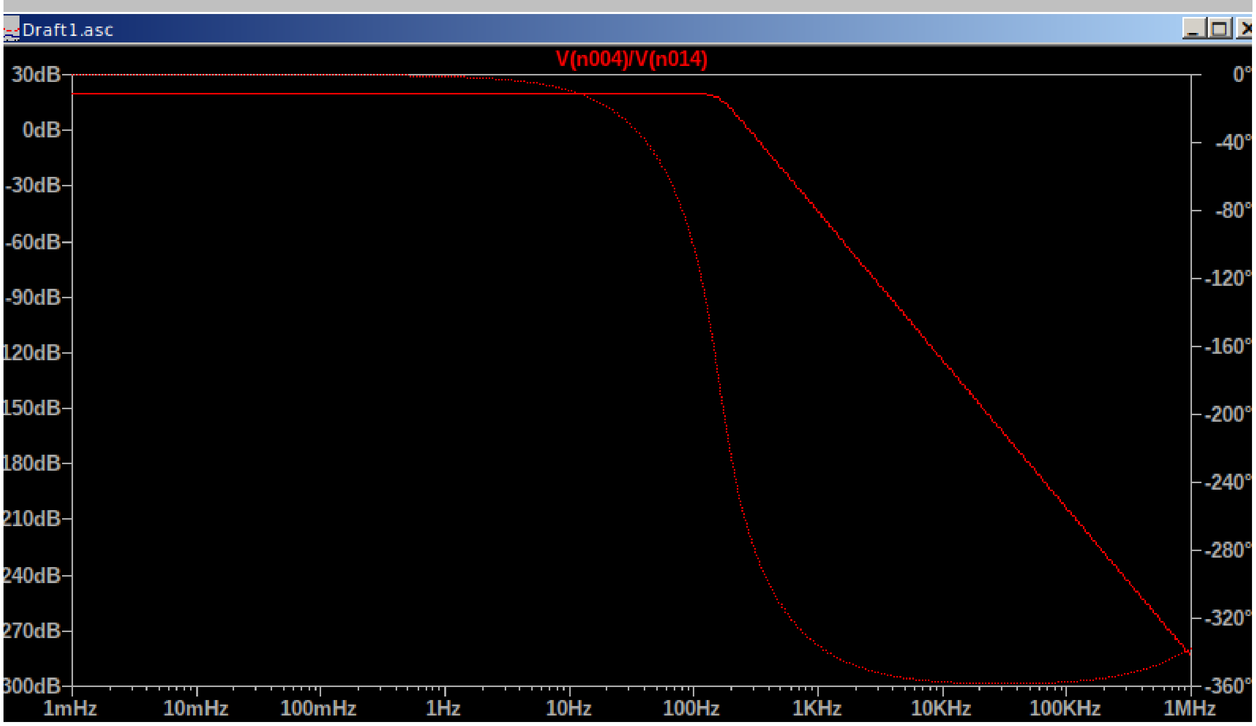
2.4 Para obtener un buttherworth de 4to orden:

Este se compone de dos transferencias, donde un par de polos esta a $S_1 = \pi/8$ y el otro $S_2 = 3\pi/8$ entonces la T(S) queda:

$$T(S) = \frac{K^2}{(\$^2 + 2\$cos(\pi/8) + 1)(\$^2 + 2\$cos(3\pi/8) + 1)}$$

2.4.1 Simulacion





CHAPTER THREE

GUIA 1 : MEDIDAS ELECTRONICAS 1 UTN FRBA

GUÍA 1 : CÁLCULO DE INCERTIDUMBRE

1) Se realizan 20 mediciones con un multímetro digital, repetidas en las mismas condiciones ambientales, obteniéndose una media aritmética de 100,145 V y una desviación estándar experimental (S) de 1,489V.

El multímetro posee las siguientes especificaciones

- Rango: 200V
- Dígitos: 3 y ½
- Error máximo = ±(0,5% lectura + 3 digitos)

Expresar el resultado de la medición con una probabilidad del 95%

[20]:

```
import numpy as np

CANT_MEDICONES = 20
V_MEAN = 100.145
V_STD = 1.489

RANGO = 200
ERROR_LECTURA = 0.5
ERROR_CUENTA = 3

print("Analisis Tipo A")

u_i = V_STD / np.sqrt(CANT_MEDICONES)
print("u_i(V) = %0.2f" %u_i)

print ("Analisis Tipo B")

VOLTS_X_CUENTA = 199.9 / 1999
ERROR_TIPO_B_TOTAL = (ERROR_LECTURA / 100) * V_MEAN + ERROR_CUENTA * VOLTS_X_CUENTA
u_j = ERROR_TIPO_B_TOTAL / np.sqrt(3)

print("u_j(V) = %0.2f" %u_j)
```

(continues on next page)

(continued from previous page)

```
print ("Analisis Incentidumbre Combinada")

u_c = np.sqrt(u_i**2 + u_j**2)

print("u_c(V) = %0.2f" %u_c)

print ("Calculo de Veff (Grados efectivos de libertad)")

V_eff = (u_c**4) / (u_i**4 / (CANT_MEDICONES - 1))

print("V_eff = %.2f" % V_eff)

print ("Suponiendo q es Tipo B dominante!")

print(f"u_i/u_j = %.2f" %(u_i/u_j))

K = 1.9

print ("RESULTADO!")

print("V = %0.2f +- %0.2f @ 95%%; k = %0.2f" %(V_MEAN, u_c * K, K))
```

```
Analisis Tipo A
u_i(V) = 0.33
Analisis Tipo B
u_j(V) = 0.46
Analisis Incentidumbre Combinada
u_c(V) = 0.57
Calculo de Veff (Grados efectivos de libertad)
V_eff = 162.88
Suponiendo q es Tipo B dominante!
u_i/u_j = 0.72
RESULTADO!
V = 100.14 +- 1.08 @ 95%; k = 1.90
```

2) Considere cinco conjuntos independientes de observaciones simultáneas de las tres magnitudes de entrada V, I, y φ se obtienen en condiciones similares.

Número de grupo k	Magnitudes de entrada V (V)	I (mA)	ϕ (rad)
1	5,007	19,663	1,0456
2	4,994	19,639	1,0438
3	5,005	19,640	1,0468
4	4,990	19,685	1,0428
5	4,999	19,678	1,0433
Media aritmética	$\bar{V} = 4{,}9990$	$\bar{I} = 19{,}6610$	$\bar{\phi} = 1{,}044\ 46$
Desviación estándar experimental del método	$s(\bar{V}) = 0{,}0032$	$s(\bar{I}) = 0{,}0095$	$s(\bar{\phi}) = 0{,}000\ 75$
Coeficiente de correlación			
$r(\bar{V},\bar{I}) = -0{,}36$ $r(\bar{V},\bar{\phi}) = 0{,}86$ $r(\bar{I},\bar{\phi}) = -0{,}65$			

a) Cuáles son las circunstancias en las que debe observarse la correlación entre las magnitudes de entrada. ¿Un coeficiente de correlación igual a 0 (cero) indica que la correlación es alta o baja?

b) Calcule la incertidumbre combinada de las cantidades R, X y Z.

[21]:
```
V   = np.array([5.007, 4.994, 5.005, 4.990, 4.999])
I   = np.array([19.663, 19.639, 19.640, 19.685, 19.678])
PHI = np.array([1.0456, 1.0438, 1.0468, 1.0428, 1.0433])

V_media = V.mean()
I_media = I.mean()
PHI_media = PHI.mean()

u_i_V = V.std(ddof=1) / np.sqrt(V.size)
u_i_I = I.std(ddof=1) / np.sqrt(I.size)
u_i_PHI = PHI.std(ddof=1) / np.sqrt(PHI.size)

print ("u_i_V = %0.4f\nu_i_I = %0.4f\nu_i_phi = %0.4f" %(u_i_V, u_i_I, u_i_PHI))

CORRELACION_MATRIX = np.corrcoef([V,I,PHI])

corr_VI = CORRELACION_MATRIX[0][1]
corr_VPHI = CORRELACION_MATRIX[0][2]
corr_IPHI = CORRELACION_MATRIX[1][2]

print ("RESULTADO! PARA Z")

Z_media = V_media / I_media * np.exp(1j * PHI_media)
```

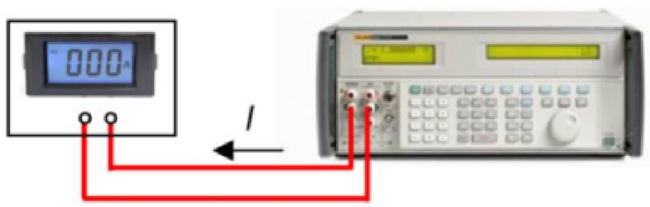
(continues on next page)

(continued from previous page)

```
dZ_dV = 1 / Z_media
dZ_dI = - V_media / I_media**2
dZ_dPHI = 1j * Z_media * PHI_media * np.exp(1j * PHI_media)

u_c_Z = np.sqrt((dZ_dV * u_i_V)**2 + (dZ_dI * u_i_I)**2 + (dZ_dPHI * u_i_PHI)**2 + 2 *␣
↪(dZ_dV * dZ_dI * corr_VI * u_i_V * u_i_I + dZ_dV * dZ_dPHI * corr_VPHI * u_i_V * u_i_
↪PHI + dZ_dI * dZ_dPHI * corr_IPHI * u_i_I * u_i_PHI))

K = 2.87 # De tabla T-Student con vi = 4

print ("Z = ({:.4f}) +- ({:.4f}) @ 95%%; k = {:.2f}".format(Z_media, u_c_Z * K, K))

print ("RESULTADO! PARA R")

R_media = np.abs(Z_media) * np.cos(PHI_media)

dR_dV = np.cos(PHI_media) / I_media
dR_dPHI = - np.abs(Z_media) * np.sin(PHI_media)
dR_dI = - V_media / I_media**2 * np.cos(PHI_media)

u_c_R = np.sqrt((dR_dV * u_i_V)**2 + (dR_dI * u_i_I)**2 + (dR_dPHI * u_i_PHI)**2 + 2 *␣
↪(dR_dV * dR_dI * corr_VI * u_i_V * u_i_I + dR_dV * dR_dPHI * corr_VPHI * u_i_V * u_i_
↪PHI + dR_dI * dR_dPHI * corr_IPHI * u_i_PHI * u_i_I))

print ("R = ({:.4f}) +- ({:.4f}) @ 95%%; k = {:.2f}".format(R_media, u_c_R * K, K))
```

```
u_i_V = 0.0032
u_i_I = 0.0095
u_i_phi = 0.0008
RESULTADO! PARA Z
Z = (0.1277+0.2198j) +- (0.0179-0.0316j) @ 95%%; k = 2.87
RESULTADO! PARA R
R = (0.1277) +- (0.0002) @ 95%%; k = 2.87
```

3) Se busca calibrar la función amperímetro de alterna de un multímetro digital de 3 ½ dígitos. Se controlará en este ejercicio sólo el punto de fondo de escala de 10A, con 50 Hz. Se empleará un calibrador (aparato que provee la corriente necesaria y la indicación de su valor, 10A en este caso, con alta exactitud).



Se toman 5 mediciones sucesivas en el instrumento a contrastar que arrojan los siguientes valores:

N	**1**	**2**	**3**	**4**	**5**
I [A]	10,01	10,00	10,02	10,01	10,00

El fabricante del calibrador específica en su catálogo que la incertidumbre expandida de este dispositivo es ±(0,05% lectura + 2 mA) con una probabilidad de 99% y distribución gaussiana. Estime el error y la incertidumbre expandida en el error de la medida de 10 A, con un factor de cobertura del 95 %.

[5]:
```python
I = np.array([10.01, 10, 10.02, 10.01, 10])

I_media = I.mean()

u_i = I.std(ddof=1) / np.sqrt(I.size)

ERROR_LECTURA = 0.05
ERROR_I = 2
ERROR_TIPO_B_TOTAL = (ERROR_LECTURA / 100) * I_media + ERROR_I
u_j = ERROR_TIPO_B_TOTAL / 2.576 # el valor cte sale del 99% de la tabla

u_c = np.sqrt(u_i**2 + u_j**2)

V_eff = (u_c**4) / (u_i**4 / (len(I) - 1))

print (f"u_i = {u_i:.4f} u_j = {u_j:.4f}")
print (f"V_eff = {V_eff:.4f}")
print ("Dado q V_eff >> 30 => TIPO B dominante")

K = 1.655
print (f"ui/uj = {(u_i/u_j)} => de la tabla K = {K}")
print ("I = {:.4f} +- {:.4f} @ 95%%; k = {:.2f}".format(I_media, u_c * K, K))
```

```
u_i = 0.0037 u_j = 0.7783
```

(continues on next page)

(continued from previous page)

```
V_eff = 7490339992.1450
Dado q V_eff >> 30 => TIPO B dominante
ui/uj = 0.004807227032130344 => de la tabla K = 1.655
I = 10.0080 +- 1.2882 @ 95%%; k = 1.66
```

4) En una resistencia alimentada con una fuente de corriente de 10A, ±0,1% según expresaba su certificado de calibración con distribución normal y un intervalo de confianza del 95%, se obtuvo una medición de 123,38V y un desvío estándar experimental de 50mV con un voltímetro digital de de 4 ¾ dígitos y un error de ±(0,04%+1d) rangos de 400mV, 4V, 40V, 400V.

Utilizando dicha resistencia como medidor indirecto de corriente, se midió sobre ella una tensión, con el voltímetro anterior, obteniéndose una indicación de 346,42mV y un desvío estándar experimental de 0,50mV. Determinar:

a. Característica de la resistencia.
b. El resultado de ambas mediciones.

c. Indique la potencia disipada en ambas mediciones.

[45]:
```python
# I Distribucion normal y confianza 95%
I_MEAN = 10
ERROR_I = .1

V_MEAN = 123.38
V_STD  = .05
ERROR_DIGITOS = 1
ERROR_LECTURE = .04

print ("Hallare R:")

# R = V/I
R_MEAN = V_MEAN / I_MEAN

u_i_V = V_STD
u_j_V = (ERROR_LECTURA * V_MEAN / 100) + (399.99 * ERROR_DIGITOS / 39999)
u_c_V = np.sqrt(u_i_V**2 + u_j_V**2)

u_c_I = ERROR_I * I_MEAN / 2 # debido a q es una normal

dR_dV = 1 / I_MEAN
dR_dI = - V_MEAN / I_MEAN**2

u_c_R = np.sqrt((dR_dV * u_c_V)**2 + (dR_dI * u_c_I)**2)

K = 2
print ("R = {:.4f} +- {:.4f} @ 95%%; k = {:.2f}".format(R_MEAN, u_c_R * K, K))

print ("Potencia disipada:")

P_MEAN = I_MEAN**2 * R_MEAN
```

(continues on next page)

(continued from previous page)

```
dP_dI = 2 * I_MEAN * R_MEAN
dP_dR = I_MEAN**2

u_c_P = np.sqrt((dP_dI * u_c_I)**2 + (dP_dR * u_c_R)**2)

print ("P = {:.4f} +- {:.4f} @ 95%%; k = {:.2f}".format(P_MEAN, u_c_P * K, K))

print ("Hallare I:")

# I = V/R
V_MEAN = .34642
V_STD = .05

I_MEAN = V_MEAN / R_MEAN

u_i_V = V_STD
u_j_V = (ERROR_LECTURA * V_MEAN / 100) + (.39999 * ERROR_DIGITOS / 39999) + V_MEAN / u_c_
↪R

u_c_V = np.sqrt(u_i_V**2 + u_j_V**2)

P_MEAN = I_MEAN**2 * R_MEAN

dI_dV = 1 / R_MEAN
dI_dR = - V_MEAN / R_MEAN**2

u_c_I = np.sqrt((dI_dV * u_c_V)**2 + (dI_dR * u_c_R)**2)

print ("I = {:.4f} +- {:.4f} @ 95%%; k = {:.2f}".format(I_MEAN, u_c_I * K, K))

print ("Potencia disipada:")

u_c_P = np.sqrt((dP_dI * u_c_I)**2 + (dP_dR * u_c_R)**2)

print ("P = {:.4f} +- {:.4f} @ 95%%; k = {:.2f}".format(P_MEAN, u_c_P * K, K))
```

```
Hallare R:
R = 12.3380 +- 1.2402 @ 95%%; k = 2.00
Potencia disipada:
P = 1233.8000 +- 276.1726 @ 95%%; k = 2.00
Hallare I:
I = 0.0281 +- 0.0912 @ 95%%; k = 2.00
Potencia disipada:
P = 0.0097 +- 126.0467 @ 95%%; k = 2.00
```

[3]:
```
# -*- coding: utf-8 -*-

import numpy as np
import pandas as pd
# import openpyxl
```

(continues on next page)

(continued from previous page)

```
# Crear un nuevo libro de trabajo
# wb = openpyxl.Workbook()

# Seleccionar la hoja activa (por defecto es la primera hoja)
# hoja = wb.active

# Especifica la ruta de tu archivo Excel
ruta_archivo = 'TP1_Medidas1.xlsx'

# Lee el archivo Excel en un DataFrame de pandas
datos = pd.read_excel(ruta_archivo)

# Especifica el tamaño del chunk (10 en este caso)
tamaño_chunk = 10

format1 = "%-20s %-20s %-20s %-25s %-20s %-20s %-20s %-15s %-20s"
format2 = "%-20f %-20f %-20f %-25f %-20f %-20f %-20f %-15s %-20f"

ERROR_LECTURA_HB113C = .5
ERROR_CUENTAS_HB113C =  5
CANT_MAX_CUENTAS = 2000
K = 2

for k in [0,1]:

    # hoja['A%i' % (k*20)] = "Valor Verdadero"
    # hoja['B%i' % (k*20)] = "Valor Medido"
    # hoja['C%i' % (k*20)] = "Desviacion"
    # hoja['D%i' % (k*20)] = "Incertidumbre Expandida"
    # hoja['E%i' % (k*20)] = "Valor Nominal"
    # hoja['F%i' % (k*20)] = "Valor Escala"
    # hoja['G%i' % (k*20)] = "Factor de Expansion"
    # hoja['H%i' % (k*20)] = "Verifica"
    # hoja['I%i' % (k*20)] = "U_j"

    print(format1 % ("Valor Verdadero", "Valor Medido", "Desviacion", "Incertidumbre␣
↪Expandida", "Valor Nominal", "Valor Escala", "Factor de Expansion", "Verifica", "U_j"))

    # Itera sobre el rango de índices en grupos de 10
    for i in range(0, len(datos), tamaño_chunk):

        valorPatron  = datos.iloc[i:i+tamaño_chunk, 0]
        valorMedido1 = datos.iloc[i:i+tamaño_chunk, k+1]
        valorNominal = datos.iloc[i, 3]
        valorEscala  = datos.iloc[i, 4]

        ERROR_VOLT_X_CUENTA_HB113C = valorEscala / CANT_MAX_CUENTAS

        u_j_HB113C = (ERROR_LECTURA_HB113C / 100) * valorMedido1.mean() + ERROR_VOLT_X_
↪CUENTA_HB113C * ERROR_CUENTAS_HB113C

        desviacion = valorMedido1.mean() - valorPatron.mean()
```

(continues on next page)

(continued from previous page)

```
        u_i = valorMedido1.std(ddof=1)/np.sqrt(10)
        incertidumbre_expandida = np.sqrt(u_i**2 + (valorPatron.std(ddof=1)/np.
↪sqrt(10))**2)

        verifica = "Si" if (incertidumbre_expandida <= u_j_HB113C) else "No"

        print (format2 % (valorPatron.mean(), valorMedido1.mean(), desviacion,␣
↪incertidumbre_expandida, valorNominal, valorEscala, K, verifica, u_j_HB113C))

        # hoja['A%i' % (i/10 + 2 + k*20)] = valorPatron.mean()
        # hoja['B%i' % (i/10 + 2 + k*20)] = valorMedido1.mean()
        # hoja['C%i' % (i/10 + 2 + k*20)] = desviacion
        # hoja['D%i' % (i/10 + 2 + k*20)] = incertidumbre_expandida
        # hoja['E%i' % (i/10 + 2 + k*20)] = valorNominal
        # hoja['F%i' % (i/10 + 2 + k*20)] = valorEscala
        # hoja['G%i' % (i/10 + 2 + k*20)] = K
        # hoja['H%i' % (i/10 + 2 + k*20)] = verifica
        # hoja['I%i' % (i/10 + 2 + k*20)] = u_j_HB113C

    print()

# Guardar el libro de trabajo
# wb.save('archivo_excel.xlsx')
```

```
Valor Verdadero     Valor Medido        Desviacion          Incertidumbre Expandida  ␣
↪Valor Nominal        Valor Escala        Factor de Expansion  Verifica        U_j
0.010833            0.010760            -0.000073           0.000151                 ␣
↪0.010000             0.200000            2.000000            Si              0.000554
0.050346            0.095690            0.045344            0.045368                 ␣
↪0.050000             0.200000            2.000000            No              0.000978
0.049975            0.049000            -0.000975           0.000014                 ␣
↪0.050000             2.000000            2.000000            Si              0.005245
0.049382            0.040000            -0.009382           0.000019                 ␣
↪0.050000             20.000000           2.000000            Si              0.050200
0.099083            0.099170            0.000087            0.000086                 ␣
↪0.100000             0.200000            2.000000            Si              0.000996
0.101520            0.101000            -0.000520           0.000096                 ␣
↪0.100000             2.000000            2.000000            Si              0.005505
0.102243            0.090000            -0.012243           0.000104                 ␣
↪0.100000             20.000000           2.000000            Si              0.050450
1.201030            1.209100            0.008070            0.003102                 ␣
↪1.000000             2.000000            2.000000            Si              0.011045
1.178752            1.159000            -0.019752           0.017817                 ␣
↪1.000000             20.000000           2.000000            Si              0.055795
2.067252            1.911000            -0.156252           1.271667                 ␣
↪1.000000             1000.000000         2.000000            Si              2.509555
10.049718           10.096000           0.046282            0.010808                 ␣
↪10.000000            20.000000           2.000000            Si              0.100480
10.052220           9.900000            -0.152220           0.100468                 ␣
↪10.000000            200.000000          2.000000            Si              0.549500
11.629100           10.610000           -1.019100           2.195704                 ␣
↪10.000000            1000.000000         2.000000            Si              2.553050
```

(continues on next page)

(continued from previous page)

```
25.063640           25.090000           0.026360            0.010391            ␣
↪25.000000           200.000000          2.000000            Si                 0.625450
25.038889           25.000000           -0.038889           0.000401            ␣
↪25.000000           1000.000000         2.000000            Si                 2.625000

Valor Verdadero     Valor Medido        Desviacion          Incertidumbre Expandida  ␣
↪Valor Nominal       Valor Escala        Factor de Expansion  Verifica           U_j
0.010833            0.011900            0.001067            0.000162            ␣
↪0.010000            0.200000            2.000000            Si                 0.000559
0.050346            0.049600            -0.000746           0.000004            ␣
↪0.050000            0.200000            2.000000            Si                 0.000748
0.049975            0.048000            -0.001975           0.000014            ␣
↪0.050000            2.000000            2.000000            Si                 0.005240
0.049382            0.040000            -0.009382           0.000019            ␣
↪0.050000            20.000000           2.000000            Si                 0.050200
0.099083            0.097740            -0.001343           0.000056            ␣
↪0.100000            0.200000            2.000000            Si                 0.000989
0.101520            0.098900            -0.002620           0.000139            ␣
↪0.100000            2.000000            2.000000            Si                 0.005495
0.102243            0.091000            -0.011243           0.001005            ␣
↪0.100000            20.000000           2.000000            Si                 0.050455
1.201030            1.183500            -0.017530           0.003072            ␣
↪1.000000            2.000000            2.000000            Si                 0.010918
1.178752            1.134000            -0.044752           0.015026            ␣
↪1.000000            20.000000           2.000000            Si                 0.055670
2.067252            1.888000            -0.179252           1.255293            ␣
↪1.000000            1000.000000         2.000000            Si                 2.509440
10.049718           9.872000            -0.177718           0.008050            ␣
↪10.000000           20.000000           2.000000            Si                 0.099360
10.052220           9.720000            -0.332220           0.080584            ␣
↪10.000000           200.000000          2.000000            Si                 0.548600
11.629100           10.570000           -1.059100           2.166545            ␣
↪10.000000           1000.000000         2.000000            Si                 2.552850
25.063640           24.550000           -0.513640           0.061978            ␣
↪25.000000           200.000000          2.000000            Si                 0.622750
25.038889           24.000000           -1.038889           0.000401            ␣
↪25.000000           1000.000000         2.000000            Si                 2.620000
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search