

Podstawy Elektroniki

Ćwiczenia komputerowe

Sonda oscyloskopowa - kompensacja częstotliwościowa

Autor: Dariusz Tefelski



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Politechnika Warszawska



Spis treści

1	Cel ćwiczenia	1
2	Wprowadzenie	1
3	Symulacje w Multisim	1
4	Warunek kompensacji sondy	2
5	Kompensacja sondy	5
6	Autorzy i historia opracowania	9





1 Cel ćwiczenia

Celem niniejszego ćwiczenia jest zapoznanie z podstawami metodami wykonywania obliczeń symbolicznych i numerycznych z pakietami SymPy oraz LCapy w środowisku Python z użyciem notatnika Jupyter Lab oraz zrozumienie działania sondy oscyloskopowej pasywnej 10:1, w tym konieczności jej kompensacji częstotliwościowej za pomocą wbudowanego trymera.

2 Wprowadzenie

Na komputerze zainstalowane jest środowisko Python wraz z notatnikiem Jupyter Lab i modułami SymPy (do obliczeń symbolicznych) i LCapy - do łatwego wykonywania obliczeń w przypadku liniowych obwodów elektrycznych.

Dokumentacje do pakietów dostępne są na stronach:

- SymPy: docs.sympy.org/latest
- LCapy: lcapy.readthedocs.io/en/latest

Warto wiedzieć



Aby uruchomić środowisko **Jupyter Lab** należy uruchomić terminal w systemie Windows. Najprościej, to otworzyć menu (albo myszką, albo klawiszem „Windows / Cmd”). Wpisać na klawiaturze **term** i nacisnąć **Enter**. W otwartym terminalu wpisać: **jupyter-lab** i nacisnąć **Enter**.

3 Symulacje w Multisim

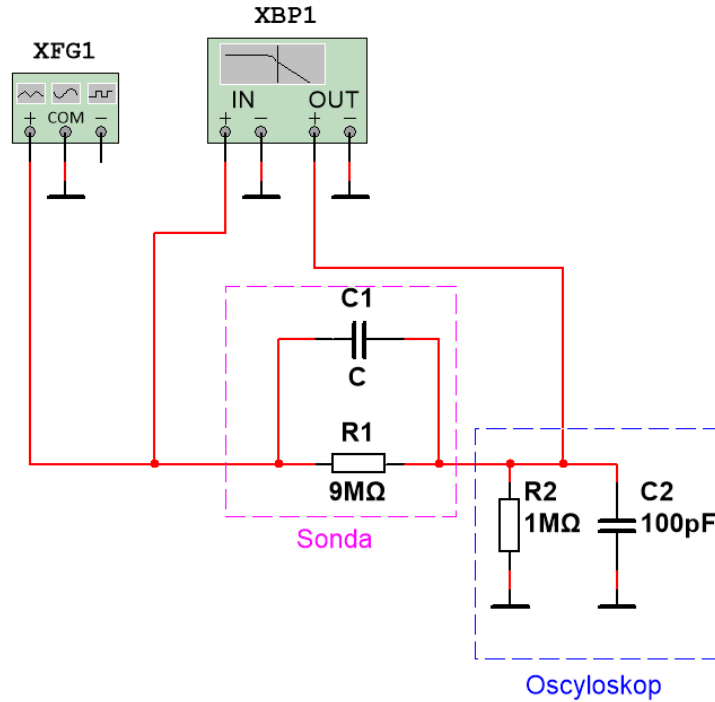
Rozpatrzmy uproszczony model typowej pasywnej sondy oscyloskopowej 10:1. Zadaniem sondy jest przekazanie wiernie (w miarę możliwości) sygnału mierzonego do oscyloskopu. Sondy posiadają trymery (regulowane kondensatory, na schemacie z rys. reprezentuje go kondensator C1), aby dla całego pasma przenoszenia sondy, transmitancja praktycznie nie zależała od częstotliwości. Typowe sondy – 10:1 tłumią sygnał wejściowy 10-krotnie, tzn. sygnał o amplitudzie 1 V będzie na oscyloskopie widoczny jak sygnał o amplitudzie 100 mV. W oscyloskopach cyfrowych konfiguruje się rodzaj podłączonej sondy, tak by wskazania napięcia były prawidłowe.

Listing 3.1: Kompensacja częstotliwościowa sondy

```
R1 = 9e6      # 9 MOhm
R2 = 1e6      # 1 MOhm
C2 = 100e-12 # 100 pF

#Warunek kompensacji sondy R1*C1 = R2*C2
C1 = R2*C2/R1
print("C_comp = ",C1)
```





Rysunek 1: Badanie modelu sondy oscyloskopowej, charakterystyka amplitudowa i fazowa w zależności od częstotliwości

Zadanie



Zadanie 1 Wyznacz wartość trymera $C1_{comp}$ z warunku kompensacji z listing 3.1, a następnie wykonaj symulację sondy w dziedzinie częstotliwości dla:

- Sondy skompensowanej $C1 = C_{comp}$
- Sondy niedokompensowanej $C1 \approx 0.1C_{comp}$
- Sondy przekompensowanej $C1 \approx 10C_{comp}$

Wyjaśnij zaobserwowane zmiany na charakterystyce amplitudowej i fazowej. Następnie zamiast analizatora Bodego podłącz oscyloskop i ustaw na generatorze sygnał prostokątny o częstotliwości 1 kHz. Zaobserwuj jak zachowuje się sygnał w kanale B oscyloskopu w zależności czy sonda jest skompensowana, niedokompensowana, przekompensowana.

4 Warunek kompensacji sondy

Transmitancja napięciowa sondy to ułamek liczb zespolonych. Aby transmitancja nie zależała od częstotliwości, wynik dzielenia dwóch liczb zespolonych powinien być liczbą rzeczywistą.

1. Postać ogólna transmitancji





Po uproszczeniu wzoru na dzielnik napięcia, otrzymujemy transmitancję w formie:

$$H(j\omega) = \frac{V_{out}}{V_{in}} = \frac{A + j\omega B}{C + j\omega D}$$

Gdzie:

- $A = R_2$
- $B = R_1 R_2 C_1$
- $C = R_1 + R_2$
- $D = R_1 R_2 (C_1 + C_2)$

2. Warunek kompensacji sondy

Chcemy, aby sonda była „idealnie przezroczysta” dla sygnału, tzn.

- **Amplituda:** Stosunek napięć musi wynosić zawsze 1/10 (niezależnie od tego, czy mierzymy napięcie stałe (DC), czy sygnał o częstotliwości np. 100 MHz).
- **Faza:** Przesunięcie fazowe musi wynosić 0 (sygnał wyjściowy nie może być opóźniony względem wejściowego).

Oznacza to, że transmitancja $H(j\omega)$ musi być równa pewnej **stałej rzeczywistej** k (w naszym przypadku $k = 0,1$):

$$\frac{A + j\omega B}{C + j\omega D} = k$$

3. Co to oznacza fizycznie?

Jeśli iloraz części rzeczywistej do urojonej jest taki sam dla licznika i mianownika, to:

- Wektory (liczby zespolone) licznika i mianownika leżą na tej samej prostej na płaszczyźnie zespolonej – mają ten sam argument (kąt), więc ich fazy się odejmują i przesunięcie fazowe wynosi 0.
- Wzrost częstotliwości ω powoduje, że oba wektory (licznik i mianownik) ”wydłużają się” w dokładnie takim samym tempie, więc ich **stosunek pozostaje stały**.

4. Ostateczne wyprowadzenie

Podstawiając nasze fizyczne parametry do warunku $\frac{A}{C} = \frac{B}{D}$:

$$\frac{R_2}{R_1 + R_2} = \frac{R_1 R_2 C_1}{R_1 R_2 (C_1 + C_2)}$$

Skracamy $R_1 R_2$ po prawej stronie:

$$\frac{R_2}{R_1 + R_2} = \frac{C_1}{C_1 + C_2}$$



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Politechnika Warszawska



Mnożymy na krzyż:

$$R_2C_1 + R_2C_2 = R_1C_1 + R_2C_1$$

Odejmujemy R_2C_1 z obu stron i otrzymujemy warunek kompensacji sondy:

$$R_1C_1 = R_2C_2$$

Założenie o równości ilorazów części rzeczywistych i urojonych to matematyczny sposób na powiedzenie: „Chcemy, aby licznik i mianownik zmieniały się wraz z częstotliwością w identyczny sposób, tak aby ich wzajemny stosunek pozostał stały”. Gdyby te ilorazy były różne, mianownik rósłby szybciej lub wolniej niż licznik, a wynik dzielenia zależałby od tego, jak szybko „kręci się” wektor napięcia (czyli od częstotliwości).

Możemy też przeprowadzić analizę, korzystając z pakietu obliczeń symbolicznych SymPy i wyznaczyć transmitację w formie zespolonej, wyciągnąć jej część urojoną i sprawdzić dla jakiego C_1 część urojona będzie wynosić 0 - co oznacza, że przesunięcie fazowe (argument liczby zespolonej) wynosić będzie 0. Analizę przedstawia listing 4.1.

Listing 4.1: Warunek kompensacji sondy

```
import sympy as sp

r1,r2,c1,c2,w = sp.symbols("r1 r2 c1 c2 w",real=True,positive=True)
j = sp.I
zc1 = 1/(j*w*c1) #impedancja kondensatora
zc2 = 1/(j*w*c2)
z1 = (r1 * zc1)/(r1+zc1) # połączenie równoległe R1 i C1, impedancja
↳ zastępcza
z2 = (r2 * zc2)/(r2+zc2)
ku = z2 / (z1+z2) #transmitancja
print('ku:')
display(ku)
print('uproszczenie ku:')
ku = sp.simplify(ku)
display(ku)
kk = sp.factor(ku)
print('Rozkład na czynniki (faktoryzacja) ku:')
display(kk)

print('Wyciągnięcie części urojonej Imaginary(ku):')
ip = sp.im(kk)
display(ip)
print("Rozkład na czynniki części urojonej ku:")
fip = sp.factor(ip)
display(fip)
print("Rozwiązanie wartość C1, dla której część urojona ku = 0:")
c1c = sp.solve(fip,c1)[0]
display(c1c)
```





Listing 4.1: Warunek kompensacji sondy (c.d.)

```

cv = c1c.subs([(r1,9.0e6),(r2,1.0e6),(c2,100.0e-12)])
c1_comp = sp.solve(sp.Eq(cv,c1),c1)[0]
print("Dla skompensowanej sondy C1 [F]:")
display(c1_comp)

print('Wyciągnięcie części rzeczywistej Real(ku):')
rp = sp.re(kk)
display(rp)

print('Podstawienie za c1 wyrażenia c2*r2/r1:')
rpp = rp.subs(c1,c2*r2/r1)
rpp = rpp.simplify()
display(rpp)
print('Zatem część rzeczywista ku nie zależy od częstości kołowej w')

```

5 Kompensacja sondy

Wykresy charakterystyki amplitudowej i fazowej dla sondy skompensowanej, niedokompensowanej i przekompensowanej przedstawia listing 5.1. Odpowiedź na skok jednostkowy dla powyższych przypadków przedstawia listing 5.2.

Listing 5.1: Charakterystyka amplitudowa i fazowa od częstotliwości

```

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

# 1. Definicja symboli
# f - częstotliwość [Hz], R, C - parametry obwodu
f = sp.symbols('f', real=True, positive=True)
R1, R2, C1, C2 = sp.symbols('R1 R2 C1 C2', real=True, positive=True)
j = sp.I
omega = 2 * sp.pi * f

# 2. Impedancje (połączenie równoległe R i C)
# Z = R || (1/j*omega*C) = R / (1 + j*omega*R*C)
Z1 = R1 / (1 + j * omega * R1 * C1)
Z2 = R2 / (1 + j * omega * R2 * C2)

# 3. Transmitancja H(f) = Vout / Vin = Z2 / (Z1 + Z2)
H = Z2 / (Z1 + Z2)

# 4. Parametry rzeczywiste (standardowa sonda 10:1 i oscyloskop)
params = {
    R1: 9e6,          # 9 MOhm
    R2: 1e6,         # 1 MOhm (wejście oscyloskopu)

```





Listing 5.1: Charakterystyka amplitudowa i fazowa od częstotliwości (c.d.)

```
C2: 100e-12, # 100 pF (pojemność wejściowa + kabel)
}

# Obliczenie idealnej pojemności kompensacyjnej (R1*C1 = R2*C2)
C1_ideal = (params[R2] * params[C2]) / params[R1] # ok. 11.11 pF
C1_comp = round(C1_ideal,13) # zmniejszenie dokładności wyniku,
# unika błędu rysowania

# 5. Przygotowanie trzech przypadków kompensacji
cases = {
    "Niedokompensowana (1 pF)": 1e-12,
    "Skompensowana (Idealna)": C1_comp,
    "Przekompensowana (100 pF)": 100e-12
}

# 6. Generowanie danych do wykresów
freqs = np.logspace(0, 7, 500) # od 1 Hz do 10 MHz

plt.figure(figsize=(12, 10))

# Wykres Modułu (Amplitudy)
ax1 = plt.subplot(2, 1, 1)
# Wykres Fazy
ax2 = plt.subplot(2, 1, 2)

for label, c1_val in cases.items():
    # Podstawienie konkretnej wartości C1 do transmitancji
    H_numeric = H.subs({'**params', C1: c1_val})
    # Zamiana wyrażenia SymPy na funkcję numeryczną NumPy (lambdify)
    # Moduł: |H(f)|
    mag_func = sp.lambdify(f, sp.Abs(H_numeric), 'numpy')
    # Faza w stopniach: arg(H(f)) * 180 / pi
    phase_func = sp.lambdify(f, sp.arg(H_numeric) * 180 / sp.pi, 'numpy')

    # Rysowanie
    ax1.semilogx(freqs, mag_func(freqs), label=label)
    ax2.semilogx(freqs, phase_func(freqs), label=label)

# Stylizacja wykresu Amplitudy
ax1.set_title("Charakterystyka amplitudowa sondy 10:1", fontsize=14)
ax1.set_ylabel("Moduł transmitancji |Vout/Vin|", fontsize=12)
ax1.set_ylim(0, 0.6) # 0.25
ax1.axhline(0.1, color='black', linestyle='--', alpha=0.5, label='Ideal
↪ 1/10')
ax1.grid(True, which="both", ls="--", alpha=0.5)
ax1.legend()
```





Listing 5.1: Charakterystyka amplitudowa i fazowa od częstotliwości (c.d.)

```

# Stylizacja wykresu Fazy
ax2.set_title("Charakterystyka fazowa sondy 10:1", fontsize=14)
ax2.set_xlabel("Częstotliwość [Hz]", fontsize=12)
ax2.set_ylabel("Przesunięcie fazowe [stopnie]", fontsize=12)
ax2.set_ylim(-60, 60)
ax2.axhline(0, color='black', linestyle='--', alpha=0.5)
ax2.grid(True, which="both", ls="-", alpha=0.5)
ax2.legend()

plt.tight_layout()
plt.show()

# 7. Wyświetlenie transmitancji w formie symbolicznej
print("Symboliczny wzór transmitancji H(f):")
sp.pprint(sp.simplify(H))

```

Listing 5.2: Odpowiedź na skok jednostkowy

```

from lcapy import Circuit
import numpy as np
import matplotlib.pyplot as plt

def simulate_probe(C_p_val):
    # Definicja obwodu w formacie lcapy (netlist)
    # Node 1: Wejście, out: Wyjście (do oscyloskopu), Node 0: Masa
    cct = Circuit()
    cct.add(f'V1 1 0 step 1') # Sygnał wejściowy (skok
    ↪ 1V)
    cct.add(f'R1 1 out 9e6') # Rezystor sondy 9M
    cct.add(f'C1 1 out {C_p_val}e-12') # Trymer sondy (pF)
    cct.add(f'R2 out 0 1e6') # Rezystor oscyloskopu 1M
    cct.add(f'C2 out 0 100e-12') # Pojemność wejściowa 100pF

    # Obliczanie odpowiedzi czasowej na węźle 2
    t = np.linspace(0, 0.005, 10000) # czas od 0 do 5ms
    response = cct['out'].v.evaluate(t)
    return t, response

cases = {
    "Niedokompensowana (C1 = 1pF)": 1,
    "Skompensowana (C1 = 11.11pF)": 11.11,
    "Przekompensowana (C1 = 100pF)": 100
}

plt.figure(figsize=(10, 6))

```





Listing 5.2: Odpowiedź na skok jednostkowy (c.d.)

```
for label, Cp in cases.items():
    t, v_out = simulate_probe(Cp)
    plt.plot(t * 1e3, v_out, label=label)

plt.axhline(0.1, color='red', linestyle='--', label='Idealne 0.1V (10:1)')
plt.title("Odpowiedź skokowa sondy 10:1 dla różnych nastaw trymera")
plt.xlabel("Czas [ms]")
plt.ylabel("Napięcie na oscyloskopie [V]")
plt.legend()
plt.grid(True)
plt.ylim(0,0.6)
plt.xlim(-0.1,1)
plt.show()
```

Informacje dodatkowe



Wyjaśnienie dlaczego sondy oscyloskopowe z przełącznikiem x10 – x1 mają znacznie niższe pasmo przenoszenia dla trybu x1, niż dla x10. Np. Sonda 250 MHz, ma dla trybu x10 pasmo 250 MHz, a dla trybu x1 tylko 8 MHz.
<https://www.youtube.com/watch?v=0iAmER10Jh4>



6 Autorzy i historia opracowania

- dr inż. Dariusz Tefelski - wersja z 2026r.



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczypospolita
Polska

Dofinansowane przez
Unię Europejską



Politechnika Warszawska

Plac Politechniki 1
00-661 Warszawa

www.pw.edu.pl