

Podstawy Elektroniki

Ćwiczenia komputerowe

Szumy elektryczne

Autor: Dariusz Tefelski



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Politechnika Warszawska



Spis treści

1	Cel ćwiczenia	1
2	Wstęp Teoretyczny	1
2.1	Szum termiczny (Johnsona-Nyquista)	1
2.1.1	Jednostka ASD	1
2.1.2	Dlaczego pierwiastek z Herca ($\sqrt{\text{Hz}}$)?	1
2.2	Szum nadmiarowy (1/f, Flicker Noise)	2
2.3	Szum we wzmacniaczach operacyjnych	2
3	Zadania ćwiczeniowe	2
3.1	Zadanie 1: Obliczenia symboliczne w SymPy	2
3.2	Zadanie 2: Wyznaczenie szumu termicznego rezystora 100k	3
3.3	Zadanie 3: Interaktywna analiza wpływu temperatury i wartości rezystancji na szum termiczny	4
3.4	Zadanie 4: Modelowanie szumów wzmacniacza operacyjnego TL071 w Lcapy	5
3.5	Zadanie 5: Szum nadmiarowy	7
3.5.1	Wyznaczenie gęstości widmowej szumu z przebiegu czasowego metodą Welcha	10
4	Pytania Kontrolne	13
4.1	Wymagane oprogramowanie	13
5	Autorzy i historia opracowania	14





1 Cel ćwiczenia

Celem niniejszego ćwiczenia jest zapoznanie ze sposobami opisu i oceny szumów występujących w obwodach elektrycznych. Poznanie różnych rodzajów szumów oraz ich rozkładu w dziedzinie częstotliwości, czyli tzw. gęstości widmowej szumów.

2 Wstęp Teoretyczny

Szumy elektryczne to niepożądane sygnały elektryczne, które nakładają się na sygnał użyteczny, ograniczając precyzję pomiarów i jakość transmisji danych.

2.1 Szum termiczny (Johnsona-Nyquista)

Jest to szum generowany przez chaotyczny ruch termiczny nośników ładunku w przewodniku (rezystorze). Jego gęstość widmowa mocy (PSD - Power Spectral Density) napięcia jest stała w bardzo szerokim zakresie częstotliwości (szum biały):

$$S_v(f) = 4kTR \quad [V^2/Hz] \quad (1)$$

Wartość skuteczna napięcia szumów (V_{RMS}) w paśmie częstotliwości B wynosi:

$$V_n = \sqrt{4kTRB} \quad (2)$$

gdzie $k \approx 1.38 \times 10^{-23} J/K$ to stała Boltzmanna, T to temperatura w Kelwinach, a R to rezystancja w omach.

Często, szczególnie w notach katalogowych producentów elementów elektronicznych wykorzystywana jest widmowa gęstość amplitudy (ASD - Amplitude Spectral Density), wyrażana nie w jednostce mocy, a w woltach albo amperach.

2.1.1 Jednostka ASD

- Dla szumu napięciowego: $\frac{V}{\sqrt{Hz}}$ (często spotykane jako $\frac{nV}{\sqrt{Hz}}$).
- Dla szumu prądowego: $\frac{A}{\sqrt{Hz}}$ (często spotykane jako $\frac{pA}{\sqrt{Hz}}$).

2.1.2 Dlaczego pierwiastek z Herca (\sqrt{Hz})?

Szum jest procesem losowym rozłożonym w pasmie częstotliwości. Sumowanie szumów o różnych częstotliwościach odbywa się poprzez dodawanie ich mocy (kwadratów amplitud), a nie bezpośrednio amplitud.

- PSD (Power Spectral Density) mierzy moc szumu i ma jednostkę V^2/Hz .
- ASD to pierwiastek kwadratowy z PSD: $ASD = \sqrt{PSD}$.

Dzięki temu, jeśli potrzeba obliczyć całkowite napięcie szumów (V_{RMS}) w konkretnym paśmie (np. od 10 Hz do 10 kHz), wystarczy "skałkować" ASD (podnieść do kwadratu,





scalkować po częstotliwości i wyciągnąć pierwiastek). Dla szumu białego (stałego w całym pasmie) wzór to:

$$V_{RMS} = ASD \cdot \sqrt{\Delta f}$$

Gdzie Δf to szerokość pasma.

Warto wiedzieć



ASD mówi nam, jak "głośny" jest szum na każdej pojedynczej częstotliwości (wąskim pasmie 1 Hz). Jest to standardowa miara używana przez elektroników do porównywania jakości wzmacniaczy.

Informacje dodatkowe



W karcie katalogowej wzmacniacza TL071 znajdziesz informację: Equivalent Input Noise Voltage: $18 \text{ nV}/\sqrt{\text{Hz}}$. To jest właśnie ASD. Przykładowo, jeśli wzmacniacz ma pasmo 10 kHz, to całkowity szum na wejściu wyniesie: $18 \text{ nV} \cdot \sqrt{10000} = 18 \text{ nV} \cdot 100 = 1.8 \mu\text{V}_{RMS}$.

2.2 Szum nadmiarowy (1/f, Flicker Noise)

Występuje w większości elementów elektronicznych (półprzewodniki, rezystory węglowe). Jego moc rośnie wraz ze spadkiem częstotliwości:

$$S(f) \propto \frac{1}{f^\alpha}, \quad \alpha \approx 1 \quad (3)$$

Dominuje on przy niskich częstotliwościach i jest kluczowym problemem w układach prądu stałego (DC).

2.3 Szum we wzmacniaczach operacyjnych

Rzeczywisty wzmacniacz operacyjny modeluje się jako układ idealny z dodanymi źródłami szumu na wejściu:

- e_n – szeregowe źródło szumu napięciowego [$V/\sqrt{\text{Hz}}$].
- i_n – równoległe źródło szumu prądowego [$A/\sqrt{\text{Hz}}$].

Całkowity szum na wyjściu zależy od wzmocnienia układu oraz wartości rezystancji w pętli sprzężenia zwrotnego.

3 Zadania ćwiczeniowe

3.1 Zadanie 1: Obliczenia symboliczne w SymPy

Wyznacz symbolicznie stosunek szumu termicznego dla jednego rezystora R oraz dwóch rezystorów R połączonych równoległe.





Listing 3.1: Wyznaczenie stosunku szumu termicznego jednego rezystora do dwóch rezystorów połączonych równolegle

```
import sympy as sp

k, T, R, B = sp.symbols('k T R B', positive=True)

# Szum pojedynczego rezystora
Vn_single = sp.sqrt(4 * k * T * R * B)

# Rezystancja zastępcza równoległa
R_eq = R / 2
Vn_parallel = sp.sqrt(4 * k * T * R_eq * B)

ratio = Vn_single / Vn_parallel

print(f"Stosunek szumów: {sp.simplify(ratio)}")
```

3.2 Zadanie 2: Wyznaczenie szumu termicznego rezystora 100k

Wyznacz wartość skuteczną szumu termicznego rezystora $100\text{ k}\Omega$ w temperaturze 27°C w paśmie częstotliwości od 1 Hz do 100 kHz. Wyznacz gęstość widmową napięcia szumu termicznego tego rezystora. Obliczenia wykonaj w Jupyter lab. Wykonaj symulację w Multisim i porównaj wyniki.

Listing 3.2: Wyznaczenie szumu termicznego rezystora $100\text{ k}\Omega$ w temperaturze 27°C w paśmie 100 kHz.

```
import sympy as sp
from IPython.display import display, Markdown

k,T, R, B = sp.symbols('k T R B', positive=True)

values = { k : 1.38e-23, # Stała Boltzmann J/K
           R : 100e3,    # Ohm
           T : 273.15 + 27, # Kelvin, (27 st. C)
           B : 100e3 - 1 # Hz, badane pasmo szumów
         }

vnoise = sp.sqrt(4 * k * T * R * B)
vnoise_asd = sp.sqrt(4 * k * T * R)
display(Markdown(r"""
Wartość skuteczna napięcie szumów na rezystorze 100 $\Omega$
w temperaturze 27$^\circ\text{C}$ w paśmie 100 kHz:
$V_{\text{noise}_{\text{RMS}}}$ = $ %.2F \mu\text{V}$""" % (vnoise.subs(values)/1e-6)))

display(Markdown(r"""Gęstość widmowa szumu białego na powyższym:

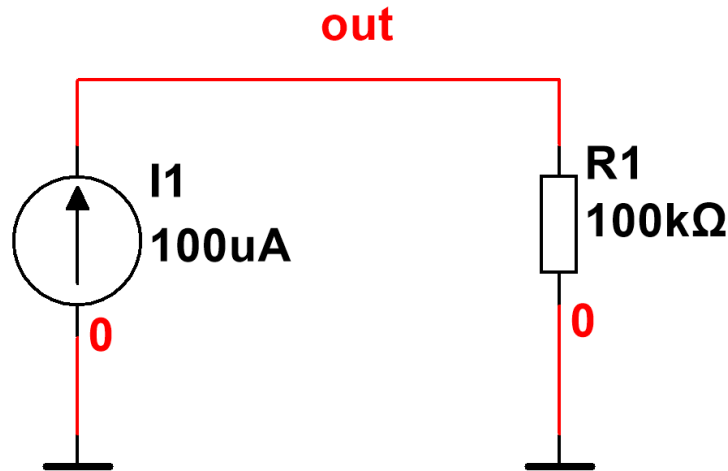
```





Listing 3.2: Wyznaczenie szumu termicznego rezystora $100\text{ k}\Omega$ w temperaturze 27°C w paśmie 100 kHz . (c.d.)

```
$V_{ASD}=\$ %.2f $\frac{nV}{\sqrt{Hz}}$""
% (vnoise_asd.subs(values)/1e-9))
```



Rysunek 1: Multisim: Obwód do wyznaczenia szumu termicznego rezystora. Źródło prądowe $100\mu\text{A}$ oraz rezystor $100\text{k}\Omega$.

3.3 Zadanie 3: Interaktywna analiza wpływu temperatury i wartości rezystancji na szum termiczny

Stwórz interaktywny symulator szumu termicznego, pozwalający zmieniać temperaturę T oraz rezystancję R .

Listing 3.3: Interaktywny symulator szumu termicznego, pozwalający zmieniać temperaturę T oraz rezystancję R

```
import ipywidgets as widgets
from ipywidgets import interact
import numpy as np
import matplotlib.pyplot as plt

def plot_noise(T_c, R_k):
    k = 1.38e-23
    T_k = T_c + 273.15
    R = R_k * 1000
    e_dens = np.sqrt(4 * k * T_k * R)

    f = np.logspace(0, 5, 100)
    plt.figure(figsize=(8,4))
    plt.semilogx(f, np.full_like(f, e_dens * 1e9), color='red')
    plt.ylim(0, 100)
    plt.title(f"Szum termiczny: {e_dens*1e9:.2f} nV/sqrt(Hz)")
```



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Politechnika Warszawska

Plac Politechniki 1
00-661 Warszawa

www.pw.edu.pl



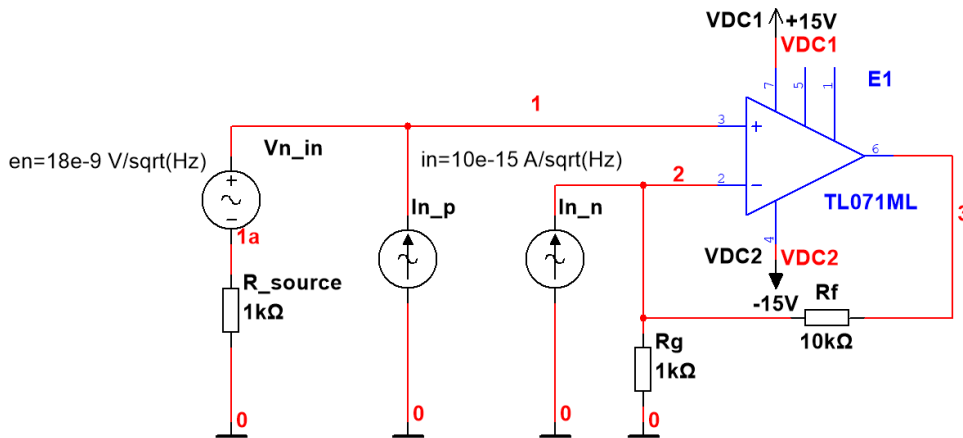
Listing 3.3: Interaktywny symulator szumu termicznego, pozwalający zmieniać temperaturę T oraz rezystancję R (c.d.)

```
plt.xlabel("Częstotliwość [Hz]")
plt.ylabel("Gęstość [nV/sqrt(Hz)]")
plt.show()

_ = interact(plot_noise,
            T_c=widgets.IntSlider(min=-50, max=200, value=25, description='Temp [C]'),
            R_k=widgets.FloatLogSlider(min=0, max=2.5, value=10,
                                       description='R [kOhm]'))
```

3.4 Zadanie 4: Modelowanie szumów wzmacniacza operacyjnego TL071 w Lcapy

Oblicz gęstość widmową szumu na wyjściu wzmacniacza nieodwracającego o wzmacnieniu $G = 11$ ($R_f = 10k, R_g = 1k$) przy użyciu parametrów TL071 ($e_n = 18nV/\sqrt{Hz}, i_n = 0.01pA/\sqrt{Hz}$). Dane z noty katalogowej: <https://www.ti.com/lit/ds/symlink/tl071.pdf>



Rysunek 2: Obwód do wyznaczenia szumu wzmacniacza operacyjnego TL071.

Listing 3.4: Analiza szumowa TL071 w Lcapy. Wyznaczenie gęstości widmowej szumu na wyjściu wzmacniacza nieodwracającego o wzmacnieniu $G = 11$ ($R_f = 10k, R_g = 1k$) przy użyciu parametrów TL071 ($e_n = 18nV/\sqrt{Hz}, i_n = 0.01pA/\sqrt{Hz}$).

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

# Parametry TL071
en = 18e-9 # 18 nV/sqrt(Hz) Równoważne napięcie
           # szumów wejściowych [V/sqrt(Hz)]
in_ = 10e-15 # 10 fA/sqrt(Hz) Równoważne natężenie prądu
```



Fundusze Europejskie dla Rozwoju Społecznego



Rzeczpospolita Polska

Dofinansowane przez Unię Europejską



Politechnika Warszawska

Plac Politechniki 1
00-661 Warszawa

www.pw.edu.pl



Listing 3.4: Analiza szumowa TL071 w Lcapy. Wyznaczenie gęstości widmowej szumu na wyjściu wzmacniacza nieodwracającego o wzmocnieniu $G = 11$ ($R_f = 10k, R_g = 1k$) przy użyciu parametrów TL071 ($e_n = 18nV/\sqrt{Hz}, i_n = 0.01pA/\sqrt{Hz}$. (c.d.)

```
# szumów wejściowych A/sqrt(Hz)
T = 290 # Temperatura [K], 17 st C.
k = 1.38e-23 # Stała Boltzmana [J/K]
A = 1e6 # Wzmocnienie z otwartą pętlą sprzężenia zwrotnego [V/V]

# Rezystory
R_f = 10e3 #10 kOhm
R_g = 1e3 #1 kOhm
# Model obwodu w Lcapy
# 0: masa
# 1: wejście (+) TL071
# 1a: węzeł przed rezystorem źródła
# 2: wejście (-) TL071
# 3: wyjście opampa
# Uwaga: używamy dużego wzmocnienia 1e6 zamiast
# 'opamp' dla stabilności numerycznej
cct = Circuit(f"""
R_source 1a 0 1000
Vn_in 1 1a noise {en}
In_p 1 0 noise {in_}
In_n 2 0 noise {in_}
R_g 2 0 {R_g}
R_f 3 2 {R_f}
E1 3 0 1 2 {A}
; autoground=tground, label_ids=true, draw_nodes=connections
; resistor_style=european
""")

# Dodanie szumów termicznych rezystorów (model Johnsona-Nyquista)
# Domyślnie używa T=290K
cct_noise = cct.noise_model()

# Częstotliwości do analizy
f = np.logspace(1, 6, 200)

# Obliczanie wypadkowej gęstość widmowej napięcia (ASD) na wyjściu
# (węzeł 3)
# lcapy.n zwraca ASD w V/sqrt(Hz) ASD -
# Amplitude Spectral Density [V/sqrt(Hz)]
# Musimy podać wartości dla T i k_B, które lcapy wstawiło jako symbole
output_noise_asd = cct_noise[3].V.n.subs({'T': T, 'k_B': k}).evaluate(f)

# Przeliczenie na nV/sqrt(Hz) dla lepszej czytelności
output_noise_nv = output_noise_asd * 1e9
```





Listing 3.4: Analiza szumowa TL071 w Lcapy. Wyznaczenie gęstości widmowej szumu na wyjściu wzmacniacza nieodwracającego o wzmocnieniu $G = 11$ ($R_f = 10k, R_g = 1k$) przy użyciu parametrów TL071 ($e_n = 18nV/\sqrt{Hz}, i_n = 0.01pA/\sqrt{Hz}$. (c.d.)

```
# Wykres
plt.figure(figsize=(10, 6))
plt.semilogx(f, output_noise_nv)
plt.title("Gęstość widmowa szumu wyjściowego wzmacniacza TL071")
plt.xlabel("Częstotliwość [Hz]")
plt.ylabel("Gęstość szumu [nV/sqrt(Hz)]")
plt.grid(True, which="both", ls="-", alpha=0.5)

# Dodanie informacji o składowych (opcjonalnie)
v_gain = 1 + R_f/R_g
theoretical_min = en * v_gain * 1e9 # nV/sqrt(Hz)
plt.axhline(y=theoretical_min, color='r', linestyle='--',
label=f'Tylko szum TL071 (~{theoretical_min:.1f} nV/sqrt(Hz))')
plt.legend()

plt.show()
```

3.5 Zadanie 5: Szum nadmiarowy

Szumy termiczne występują zawsze. Dodatkowo występują szumy nadmiarowe, których wpływ jest przeważający dla niskich częstotliwości. Porównanie w dziedzinie częstotliwości przedstawia listing 3.5, natomiast porównanie szumu termicznego oraz szumu termicznego+szumu nadmiarowego w dziedzinie czasu przedstawia listing 3.6.

Listing 3.5: Porównanie szumów termicznych oraz nadmiarowych

```
import numpy as np
import matplotlib.pyplot as plt

# 1. Parametry fizyczne
R_val = 100e3 # 100k Ohm
T = 300 # Temperatura w Kelvinach
k_B = 1.38e-23 # Stała Boltzmanna

# 2. Obliczanie składowych szumu
# Częstotliwości do analizy: 1 Hz do 100 kHz
f = np.logspace(0, 5, 500)

# a) Szum termiczny (Biały) - ASD jest stałe
v_thermal_asd_val = np.sqrt(4 * k_B * T * R_val)
thermal_component = np.full_like(f, v_thermal_asd_val) # wypełnienie
# tablicy jedną wartością szumu termicznego.
```





Listing 3.5: Porównanie szumów termicznych oraz nadmiarowych (c.d.)

```
# b) Szum nadmiarowy (1/f)
# Dobieramy K tak, aby szum 1/f był równy termicznemu przy 100 Hz
# (f_corner = 100)
# K / sqrt(100) = v_thermal_asd_val => K = v_thermal_asd_val * 10
K_excess = v_thermal_asd_val * np.sqrt(100)
excess_component = K_excess / np.sqrt(f)

# 3. Sumowanie szumów (geometryczne)
total_noise_asd = np.sqrt(thermal_component**2 + excess_component**2)

# 4. Wykres
plt.figure(figsize=(12, 7))

# Składowe (przeliczone na nV/sqrt(Hz))
plt.loglog(f, thermal_component * 1e9, '--',
label=r'Szum termiczny (Biały)', color='blue', alpha=0.6)
plt.loglog(f, excess_component * 1e9, '--',
label=r'Szum nadmiarowy (1/f)', color='red', alpha=0.6)

# Suma
plt.loglog(f, total_noise_asd * 1e9, color='black', linewidth=2.5,
label=r'Sumaryczna gęstość ASD ( $\sqrt{V_{th}^2 + V_{ex}^2}$ )')

# Opisy
plt.title(rf'Ilustracja różnych typów szumu dla rezystora {R_val/1e3:.0f}k
 $\Omega$ ', fontsize=14)
plt.xlabel("Częstotliwość [Hz]", fontsize=12)
plt.ylabel(r"Gęstość napięciowa [nV/ $\sqrt{\text{Hz}}$ ]", fontsize=12)
plt.grid(True, which="both", ls="--", alpha=0.3)

# Zaznaczenie punktu przegięcia (corner frequency)
plt.scatter([100], [np.sqrt(2)*v_thermal_asd_val*1e9], color='green',
zorder=5)
plt.annotate('Częstotliwość narożna (100 Hz)', xy=(100,
np.sqrt(2)*v_thermal_asd_val*1e9),
xytext=(150, 100), arrowprops=dict(facecolor='black',
shrink=0.05, width=1, headwidth=8))

plt.legend(fontsize=11)
plt.tight_layout()

plt.show()
```





Listing 3.6: Wygenerowane przebiegi czasowe szumu termicznego oraz szumu termicznego + szum nadmiarowy

```
import numpy as np
import matplotlib.pyplot as plt

# 1. Parametry symulacji
fs = 1_000_000      # Częstotliwość próbkowania (1 MHz)
duration = 0.01     # Czas trwania (10 ms)
N = int(fs * duration) # Liczba próbek
t = np.linspace(0, duration, N)

# Parametry fizyczne
R = 100_000        # 100 kOhm
T = 27 + 273.15   # 300.15 K
k_B = 1.38e-23

# 2. Szum termiczny (Biały)
# ASD_thermal = sqrt(4 * k_B * T * R) [V/sqrt(Hz)]
asd_thermal = np.sqrt(4 * k_B * T * R)

# Odchylenie standardowe w dziedzinie czasu (RMS)
# V_rms = ASD * sqrt(fs / 2)
sigma_thermal = asd_thermal * np.sqrt(fs / 2)
noise_thermal = np.random.normal(0, sigma_thermal, N)

# 3. Szum nadmiarowy (1/f) - Generowanie przez FFT
# Generujemy biały szum, przechodzimy do f, skalujemy przez 1/sqrt(f)
# i wracamy
white_for_1f = np.random.normal(0, 1, N)
fft_1f = np.fft.rfft(white_for_1f)
freqs = np.fft.rfftfreq(N, 1/fs)
freqs[0] = freqs[1] # Uniknięcie dzielenia przez zero w DC

# Skalowanie widma: ASD ~ 1/sqrt(f)
fft_1f = fft_1f / np.sqrt(freqs)

# Powrót do dziedziny czasu i skalowanie amplitudy
noise_excess = np.fft.irfft(fft_1f, n=N)
# Skalujemy szum 1/f do widocznego poziomu
noise_excess = noise_excess * (sigma_thermal * 20 / np.std(noise_excess))

# 4. Sumaryczny szum
noise_total = noise_thermal + noise_excess

# 5. Wykresy
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 10), sharex=True)
```





Listing 3.6: Wygenerowane przebiegi czasowe szumu termicznego oraz szumu termicznego + szum nadmiarowy (c.d.)

```
# Wykres 1: Tylko szum termiczny
ax1.plot(t * 1000, noise_thermal * 1e6, color='blue', lw=0.5, alpha=0.8)
ax1.set_title(fr"Szum termiczny rezystora {R/1000:.0f}
              k\Omega (T=27°C)", fontsize=14)
ax1.set_ylabel("Napięcie [μV]")
ax1.grid(True, alpha=0.3)

# Wykres 2: Szum termiczny + nadmiarowy
ax2.plot(t * 1000, noise_total * 1e6, color='black', lw=0.5, alpha=0.8)
ax2.set_title("Szum termiczny + szum nadmiarowy (1/f)", fontsize=14)
ax2.set_xlabel("Czas [ms]")
ax2.set_ylabel("Napięcie [μV]")
ax2.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

3.5.1 Wyznaczenie gęstości widmowej szumu z przebiegu czasowego metodą Welcha

Metodę Welcha rozkłada przebiegi czasowe na nakładające się periodogramy. Przejście z dziedziny czasu do dziedziny częstotliwości, symuluje działanie analizatora widma.

Co przedstawiają wykresy?

1. *Górne wykresy (Czas)*: Pokazują surowe dane napięcia w czasie. Szum termiczny jest jednostajny, natomiast szum sumaryczny wykazuje wyraźne wolnozmiennie fluktuacje (efekt $1/f$).
2. *Dolny lewy (ASD Termiczny)*: Pokazuje gęstość widmową wyliczoną z danych czasowych. Niebieska linia oscyluje wokół czerwonej linii teoretycznej ($\approx 40.7 \text{ nV}/\sqrt{\text{Hz}}$), co potwierdza poprawność generowania szumu białego.
3. *Dolny prawy (ASD Sumaryczny)*: Wyraźnie pokazuje charakterystyczne nachylenie dla niskich częstotliwości. Przy wysokich częstotliwościach wykres dąży do poziomu szumu termicznego (niebieska przerywana linia), co ilustruje, jak szum biały "przykrywa" szum $1/f$ powyżej częstotliwości narożnej.

Wykorzystana *metoda Welcha* (uśrednianie periodogramów) pozwoliła na uzyskanie czytelnych wykresów widmowych bez nadmiernych szumów numerycznych, które pojawiają się przy zwykłej transformacji FFT pojedynczego bloku danych.





Listing 3.7: Wyznaczenie gęstości widmowej szumu z przebiegu czasowego metodą Welch dla szumu termicznego oraz szumu termicznego + szumu nadmiarowego przy częstotliwości narożnej 10 kHz.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import welch

# 1. Parametry symulacji
fs = 1_000_000      # Częstotliwość próbkowania (1 MHz)
duration = 1.0      # Wydłużony czas (100 ms) dla lepszej
    ↪ rozdzielczości widmowej
N = int(fs * duration)
t = np.linspace(0, duration, N)

# Parametry fizyczne
R = 100_000         # 100 kOhm
T = 27 + 273.15    # 300.15 K
k_B = 1.38e-23

# 2. Generowanie szumu termicznego (Biały)
asd_thermal_theoretical = np.sqrt(4 * k_B * T * R)
sigma_thermal = asd_thermal_theoretical * np.sqrt(fs / 2)
noise_thermal = np.random.normal(0, sigma_thermal, N)

# 3. Generowanie szumu nadmiarowego (1/f) - Metoda FFT
# Generujemy widmo 1/sqrt(f)
white_for_1f = np.random.normal(0, 1, N)
fft_1f = np.fft.rfft(white_for_1f)
freqs_fft = np.fft.rfftfreq(N, 1/fs)
freqs_fft[0] = freqs_fft[1] # Uniknięcie dzielenia przez 0
fft_1f = fft_1f / np.sqrt(freqs_fft)

noise_excess = np.fft.irfft(fft_1f, n=N)
# Skalujemy szum 1/f (f_corner ~ 10 kHz dla widoczności)
noise_excess = noise_excess * (asd_thermal_theoretical * np.sqrt(10000) /
    np.std(noise_excess))

# 4. Sumaryczny szum
noise_total = noise_thermal + noise_excess

# 5. Wyznaczenie gęstości widmowej (ASD) za pomocą metody Welch
# Metoda Welch to estymacja gęstości widmowej mocy (PSD) oparta na FFT
f_welch, psd_thermal = welch(noise_thermal, fs, nperseg=8192)
asd_thermal_calc = np.sqrt(psd_thermal)

f_welch, psd_total = welch(noise_total, fs, nperseg=8192)
asd_total_calc = np.sqrt(psd_total)
```





Listing 3.7: Wyznaczenie gęstości widmowej szumu z przebiegu czasowego metodą Welch dla szumu termicznego oraz szumu termicznego + szumu nadmiarowego przy częstotliwości narożnej 10 kHz. (c.d.)

```
# 6. Wykresy
fig, axs = plt.subplots(2, 2, figsize=(15, 10))

# a) Dziedzina czasu: Szum Termiczny
axs[0, 0].plot(t[:1000] * 1000, noise_thermal[:1000] * 1e6, color='blue',
              lw=0.5)
axs[0, 0].set_title("Dziedzina czasu: Szum Termiczny (fragment)",
                  fontsize=12)
axs[0, 0].set_ylabel("Napięcie [ $\mu$ V]")
axs[0, 0].grid(True, alpha=0.3)

# b) Dziedzina czasu: Szum Sumaryczny
axs[0, 1].plot(t[:1000] * 1000, noise_total[:1000] * 1e6, color='black',
              lw=0.5)
axs[0, 1].set_title("Dziedzina czasu: Termiczny + Nadmiarowy (fragment)",
                  fontsize=12)
axs[0, 1].set_ylabel("Napięcie [ $\mu$ V]")
axs[0, 1].grid(True, alpha=0.3)

# c) Dziedzina częstotliwości: ASD Szumu Termicznego
axs[1, 0].loglog(f_welch, asd_thermal_calc * 1e9, color='blue',
                label='Obliczone z FFT')
axs[1, 0].axhline(y=asd_thermal_theoretical * 1e9, color='red', ls='--',
                 label='Teoretyczne (Białe)')
axs[1, 0].set_title("Gęstość widmowa ASD (Szum Termiczny)", fontsize=12)
axs[1, 0].set_xlabel("Częstotliwość [Hz]")
axs[1, 0].set_ylabel(r"ASD [ $\text{nV}/\sqrt{\text{Hz}}$ ]")
axs[1, 0].grid(True, which="both", alpha=0.3)
axs[1, 0].legend()

# d) Dziedzina częstotliwości: ASD Szumu Sumarycznego
axs[1, 1].loglog(f_welch, asd_total_calc * 1e9, color='black',
                label='Obliczone z FFT')
axs[1, 1].loglog(f_welch, asd_thermal_calc * 1e9, '--', color='blue',
                 alpha=0.5, label='Tylko termiczny')
axs[1, 1].set_title("Gęstość widmowa ASD (Termiczny + 1/f)", fontsize=12)
axs[1, 1].set_xlabel("Częstotliwość [Hz]")
axs[1, 1].set_ylabel(r"ASD [ $\text{nV}/\sqrt{\text{Hz}}$ ]")
axs[1, 1].grid(True, which="both", alpha=0.3)
axs[1, 1].legend()

plt.tight_layout()
plt.show()
```





4 Pytania Kontrolne

1. **Dlaczego szum termiczny nazywamy szumem białym?** Odp: Ponieważ jego gęstość widmowa mocy jest stała w całym rozpatrywanym zakresie częstotliwości, analogicznie do światła białego, które zawiera wszystkie składowe widzialne.
2. **Jakie parametry wzmacniacza operacyjnego decydują o jego szumach przy wysokich impedancjach źródła?** Odp: Przy wysokich impedancjach źródła kluczowy staje się szum prądowy (i_n), ponieważ płynąc przez rezystancję źródła, generuje on spadek napięcia szumów proporcjonalny do tej rezystancji.
3. **Jak zredukować wpływ szumu 1/f w układzie pomiarowym?** Odp: Można zastosować techniki modulacji (chopper amplifiers), które przenoszą sygnał użyteczny w zakres wyższych częstotliwości, gdzie szum 1/f jest znacznie mniejszy.

4.1 Wymagane oprogramowanie

Do wykonania ćwiczenia wymagany jest Python 3 z zainstalowanymi modułami:

- Sympy — do obliczeń symbolicznych i transformacji,
- Lcapy — do modelowania obwodów elektrycznych,
- matplotlib, numpy — do wizualizacji wyników.
- Scipy — do obliczeń na sygnałach



5 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