

Signaalin Fourier-analyysi – Tehtävä 4

1. Tehtävänanto

- Lataa signaalitiedosto.
- Laske signaalin Fourier-muunnos ja tehospektri.
- Piirrä tehospektrin kuvaaja ja määrittele, mikä on tehokkain taajuus.
- Ilmaise tulokset selkeästi. Palauta PDF, jossa koodi ja kuvaajat ovat selkeästi näkyvillä.

2. Signaalin kuvaus

Signaali on mitattu ajan funktiona. Se näyttää nopeasti vaihtelevalta ja siinä näkyy myös jonkin verran kohinaa. Signaalissa ei näytä olevan mitään selvää yhtä päätaajuutta, vaan vaihtelu on melko sekavaa ja epäsäännöllistä. Mittauksen kokonaiskesto on noin 8 sekuntia.

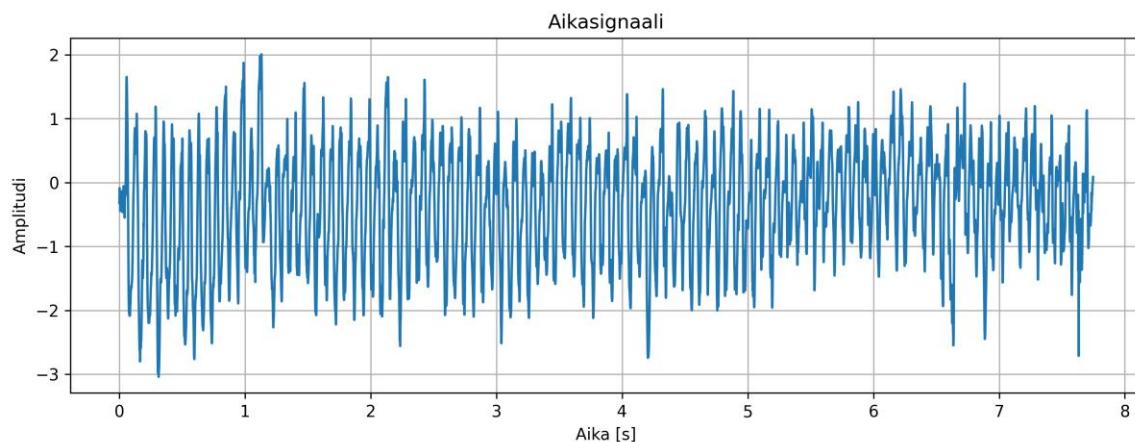


Figure 1 - Aikasignaali

3. Fourier-muunnos

Fourier-muunnos on menetelmä, jonka avulla voidaan nähdä, mitä taajuuksia signaalissa esiintyy.

Pythonissa muunnos laskettiin `numpy.fft.rfft`-toiminnolla, ja se muuttaa aikadomainin signaalin taajuusdomainiin.

Toisin sanoen:

- Aikasignaali → näyttää miten signaali vaihtelee ajassa
- Fourier-muunnos → näyttää mitä taajuuksia signaali sisältää

4. Tehospektri ja tehokkain taajuus

Tehospektri näyttää, kuinka paljon tehoa (eli voimakkuutta) signaalissa on eri taajuuksilla.

Spektri kertoo, että signaalin suurin teho on **taajuudella 0 Hz**.

Tämä tarkoittaa, että signaalissa on paljon **tasakomponenttia**, eli sen keskiarvossa on jonkin verran siirtymää.

Korkeammilla taajuuksilla teho vähenee tasaisesti, mikä on tavallista kohinaisille signaaleille.

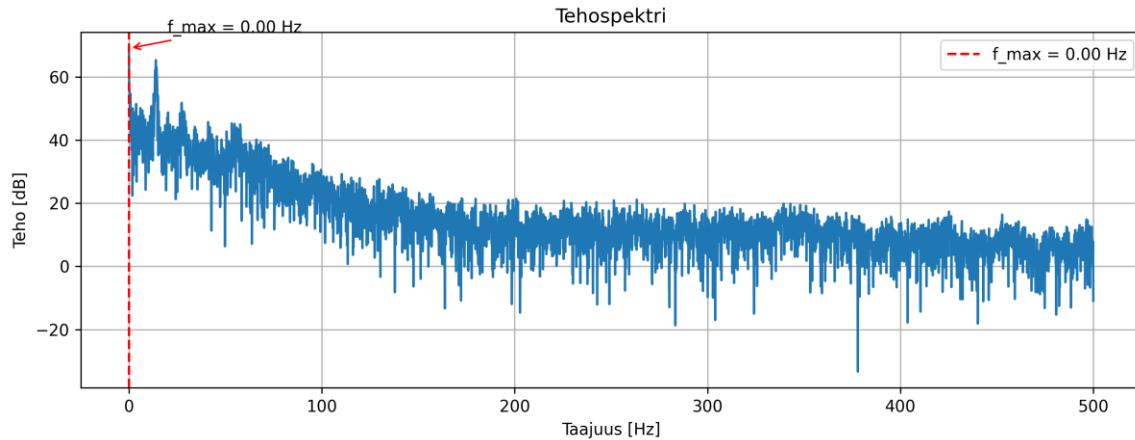


Figure 2 - Tehospektri

Tehokkain taajuus:

$f_{\max} = 0.00 \text{ Hz}$

5. Python-koodi

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** Help, Fysiikka4
- File List:** Signaali.ipynb (marked as modified), Signaali.xlsx (marked as modified)
- Toolbar:** Generate, Code, Markdown, Run All, Restart, Clear All Outputs, Jupyter Variables, Outline, ...
- Cell 14:** Contains code for importing numpy, pandas, and matplotlib.pyplot. It also includes a comment for reading the file and importing it into df. The cell status is [14] ✓ 0.0s.
- Cell 15:** Contains code for reading the Excel file, determining the data type (single column vs. two columns), extracting the signal data (x), calculating the number of samples (N), sampling frequency (FS), and time vector (t). It then prints a header "SIGNAALIN ANALYYSI" and displays the Tiedosto, sample count, sampling frequency, and duration. The cell status is [15] ✓ 0.5s.
- Output:** The output of Cell 15 is displayed below the cell, showing the printed text: "SIGNAALIN ANALYYSI", "Tiedosto : Signaali.xlsx", "Näytteiden lukumäärä : 7748", "Näytteenottotaajuus : 1000.0 Hz", and "Signaalin kesto : 7.748 s".

Figure 3 - Python koodi 1

Terminal Help

Signaali.ipynb # Kirjastot

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline ...

```
# FFT ja tehoseketri
X = np.fft.rfft(x)
f = np.fft.rfftfreq(N, d=1.0/FS)
P = np.abs(X)**2
P_db = 10 * np.log10(P + 1e-20)

# Tehokkain taajuus
idx_max = np.argmax(P)
f_max = f[idx_max]
P_max = P[idx_max]

print(f"\nTehokkain taajuus : {f_max:.3f} Hz")
print(f"Maksimiteho : {P_max:.3e}")
print("= 60")

[16] ✓ 0.0s
...
Tehokkain taajuus : 0.000 Hz
Maksimiteho : 7.936e+06
=====
```

```
# Aikasignaali
fig1, ax1 = plt.subplots(figsize=(10, 4))
ax1.plot(t, x)
ax1.set_xlabel("Aika [s]")
ax1.set_ylabel("Amplitudi")
ax1.set_title("Aikasignaali")
ax1.grid(True)
plt.tight_layout()
fig1.savefig("aikasignaali.png", dpi=300, bbox_inches='tight')
print("✓ Tallennettu: aikasignaali.png")
plt.show()

[17] ✓ 1.0s
...
✓ Tallennettu: aikasignaali.png
```

...

Figure 4 - Python koodi 2

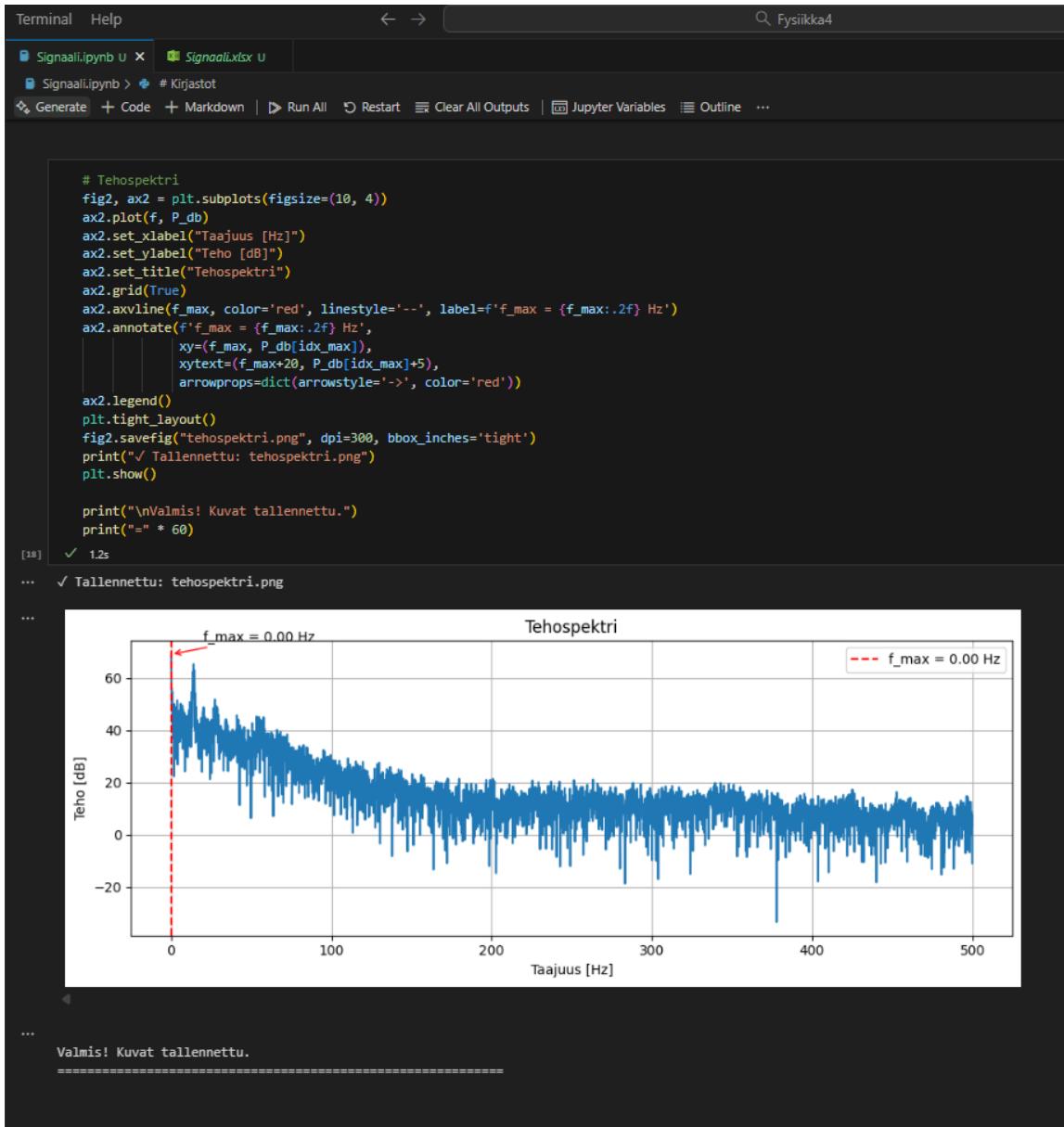


Figure 5 - Python koodi 3

6. Yhteenveto

Fourier-analyysi osoitti, että signaalissa ei ole mitään yhtä selkeää päätaajuutta, vaan se näyttää kohinaiselta ja sisältää monia eri taajuuksia. Suurin teho on taajuudella **0 Hz**, mikä tarkoittaa, että signaalissa on selkeä tasakomponentti. Tehospektri laskee tasaisesti, mikä tukee sitä, että signaali on melko kohinanainen.