

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.

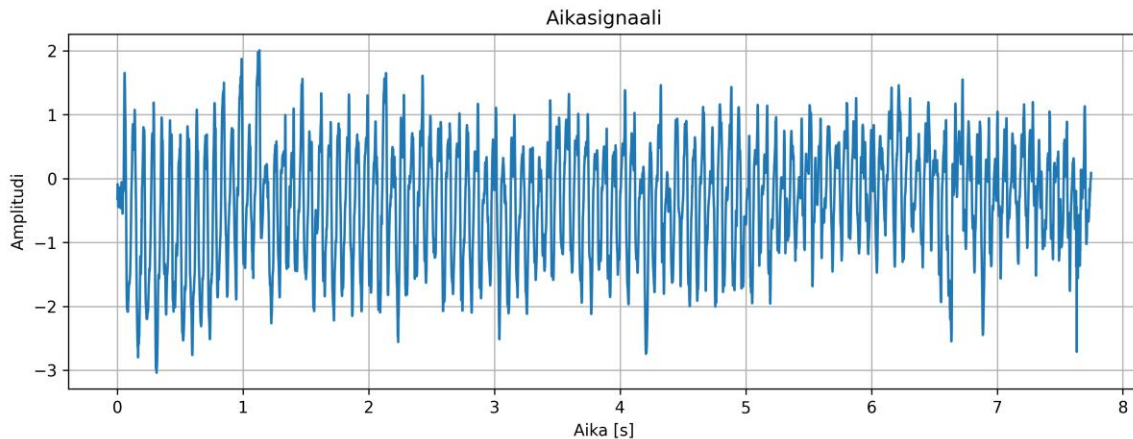


Figure 1 - Aikaisignaali

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:

- Aikaisignaali → 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.

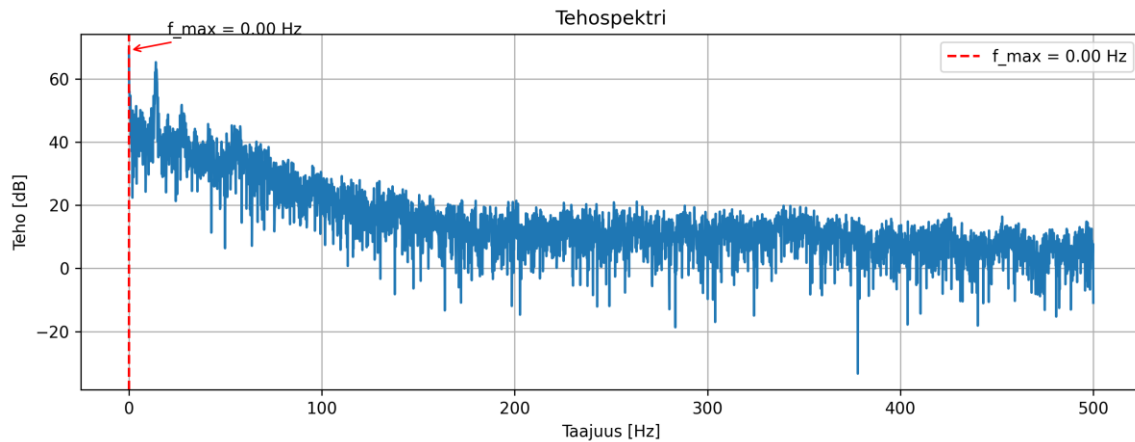
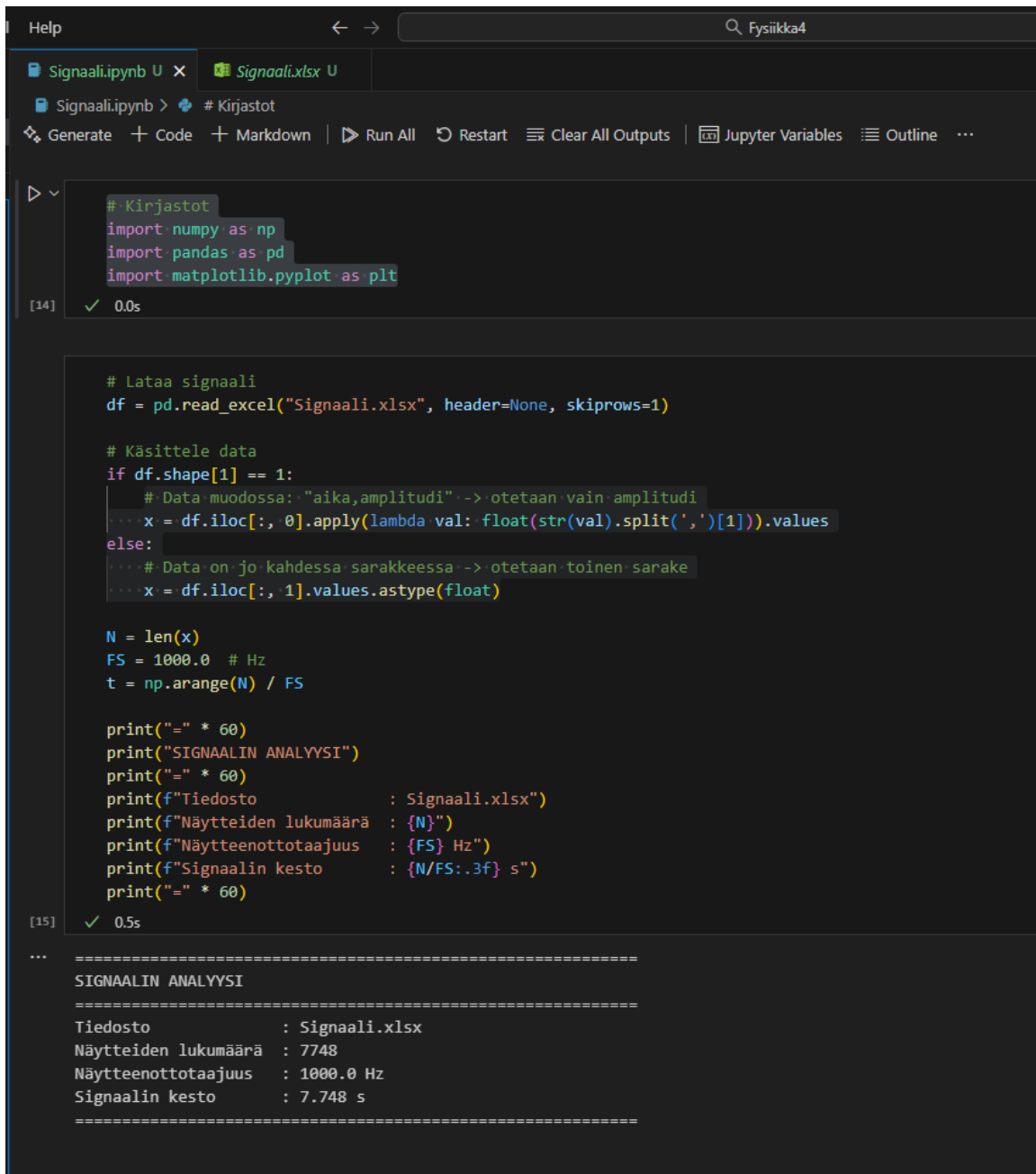


Figure 2 - Tehospektri

Tehokkain taajuus:

f_max = 0.00 Hz

5. Python-koodi



```
Help      ← →      Fysiikka4

Signaali.ipynb U ×  Signaali.xlsx U

Signaali.ipynb > # Kirjastot
Generate + Code + Markdown ▶ Run All ↺ Restart ≡ Clear All Outputs | Jupyter Variables ≡ Outline ...

# Kirjastot
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

[14] ✓ 0.0s

# Lataa signaali
df = pd.read_excel("Signaali.xlsx", header=None, skiprows=1)

# Käsittele data
if df.shape[1] == 1:
    # Data muodossa: "aika,amplitudi" -> otetaan vain amplitudi
    x = df.iloc[:, 0].apply(lambda val: float(str(val).split(',')[1])).values
else:
    # Data on jo kahdessa sarakkeessa -> otetaan toinen sarake
    x = df.iloc[:, 1].values.astype(float)

N = len(x)
FS = 1000.0 # Hz
t = np.arange(N) / FS

print("=" * 60)
print("SIGNAALIN ANALYYSI")
print("=" * 60)
print(f"Tiedosto           : Signaali.xlsx")
print(f"Näytteiden lukumäärä : {N}")
print(f"Näytteenottotaajuus  : {FS} Hz")
print(f"Signaalin kesto      : {N/FS:.3f} s")
print("=" * 60)

[15] ✓ 0.5s

...
=====
SIGNAALIN ANALYYSI
=====
Tiedosto           : Signaali.xlsx
Näytteiden lukumäärä : 7748
Näytteenottotaajuus : 1000.0 Hz
Signaalin kesto      : 7.748 s
=====
```

Figure 3 - Python koodi 1

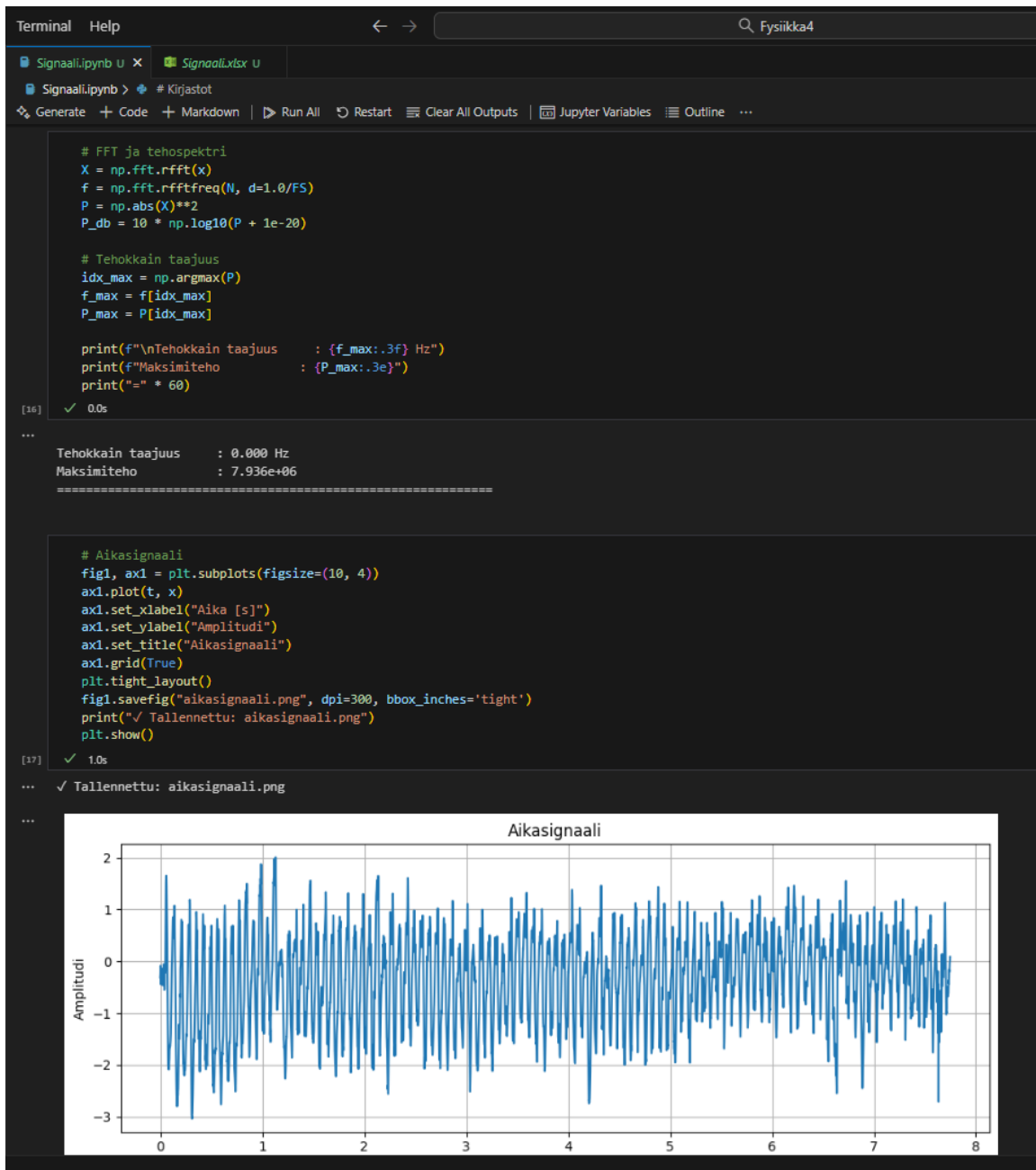


Figure 4 - Python koodi 2

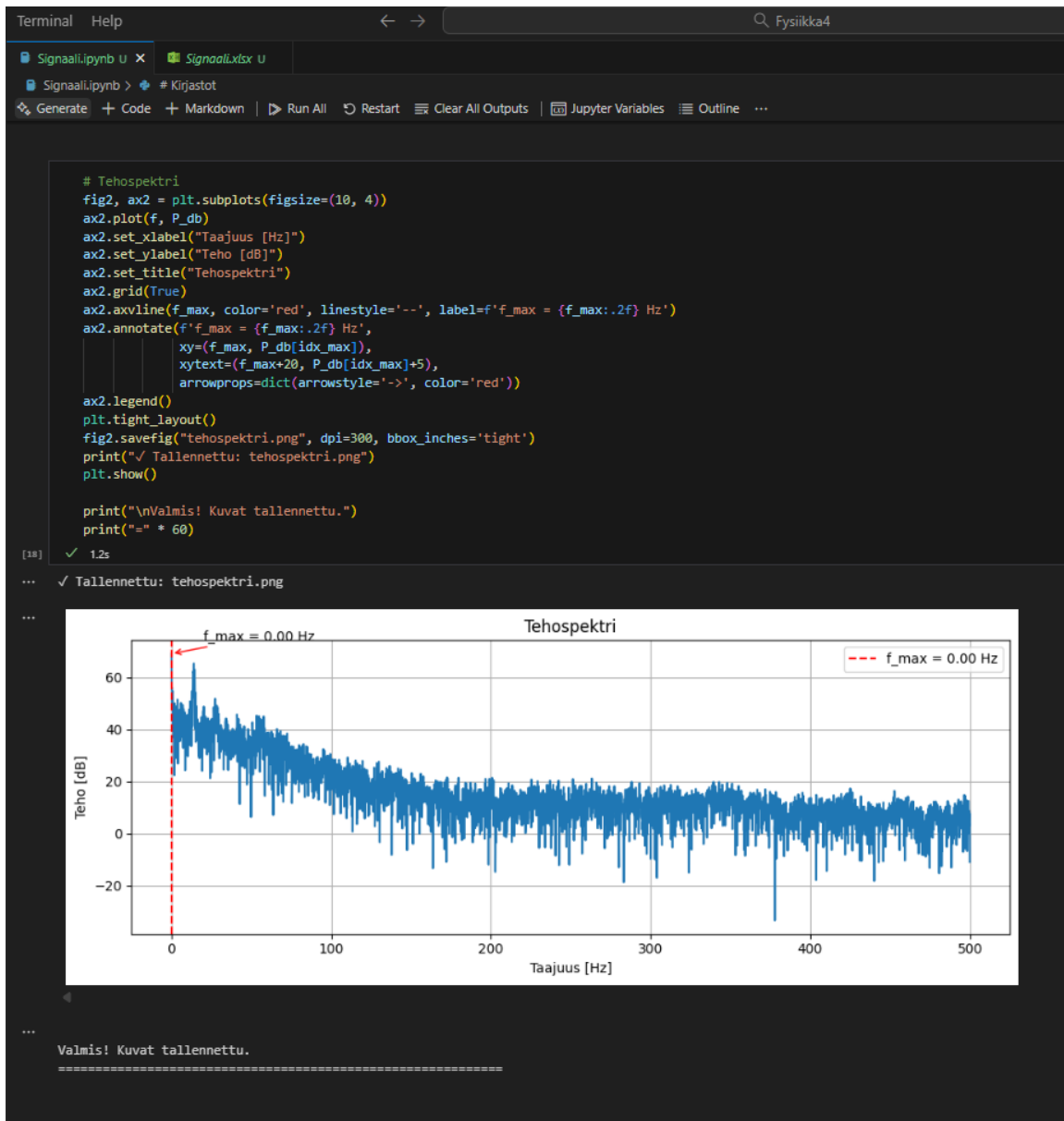


Figure 5 - Python koodi 3

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 kohinainen.