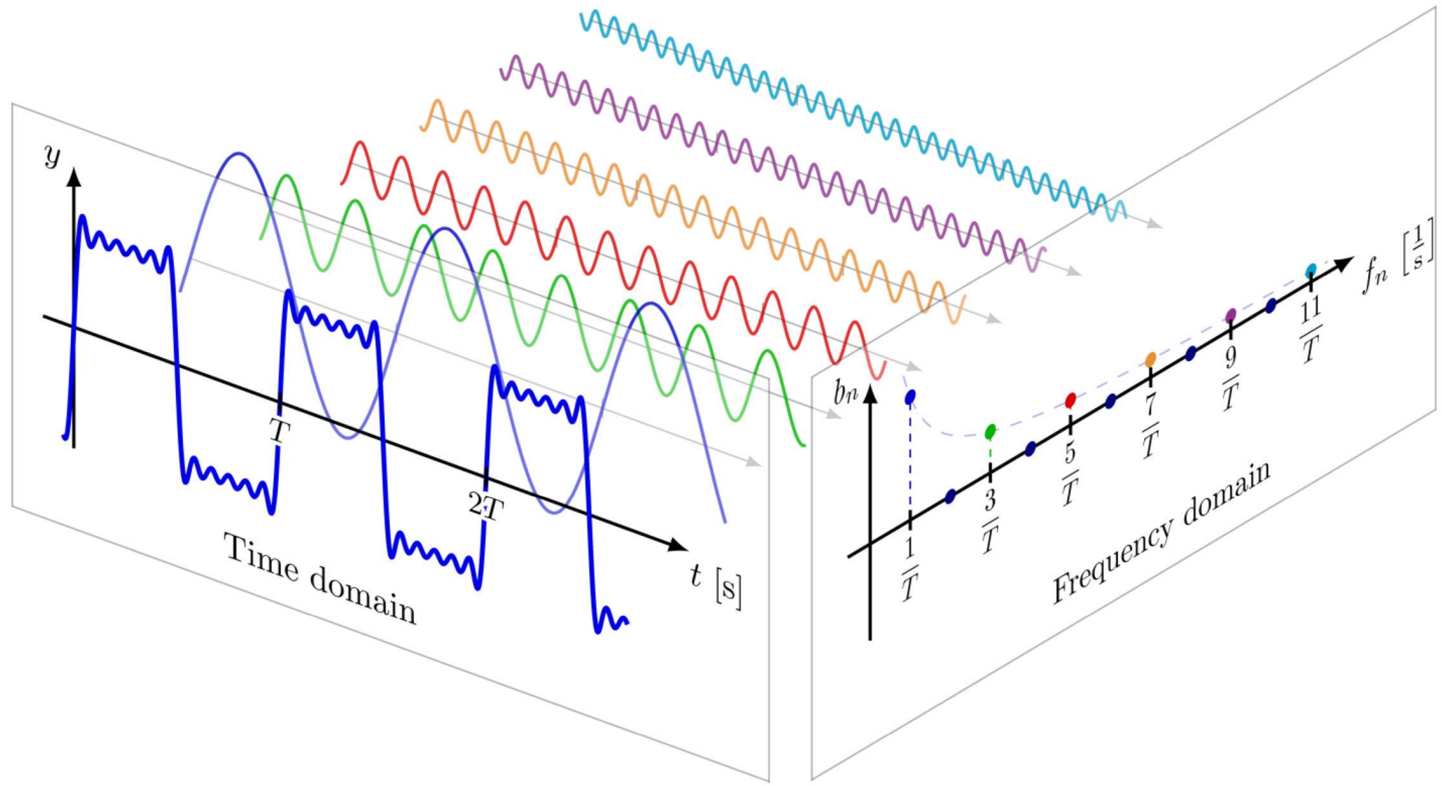
CURS 8 PD

Analiza spectrală a semnalelor



Asist. drd. ing. Iulian Vasiliev – 2024

Introducere

Transformata Fourier

Energia unui semnal

Spectrograma

Transformata Fourier

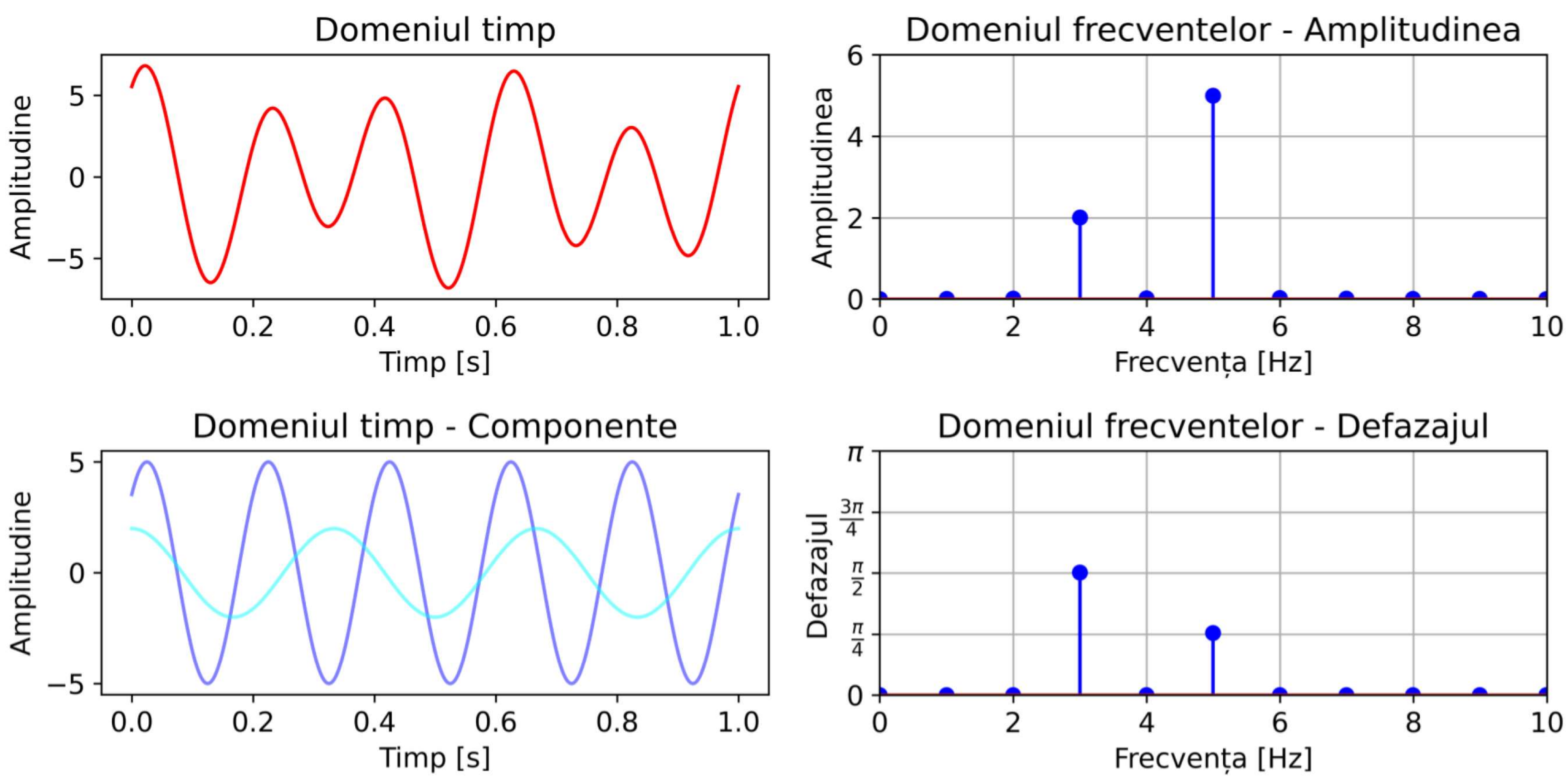
Energia unui semnal

Introducere

Spectrograma

- Până acum am privit un semnal din perspectiva evoluției amplitudinii acestuia în timp.
- Pentru o mare parte a aplicațiilor de procesare a datelor este util și necesar să analizăm semnalul din punctul de vedere al oscilațiilor ce compun semnalul.
- **Analiza spectrală** reprezintă analiza semnalului în domeniul frecvențelor și implică transformarea semnalului din domeniul timp în domeniul frecvențelor.

Exemplu: Fie semnalul $x(t) = 5 \cdot \sin\left(2 \cdot \pi \cdot t \cdot 5 + \frac{\pi}{4}\right) + 2 \cdot \sin\left(2 \cdot \pi \cdot t \cdot 3 + \frac{\pi}{2}\right)$



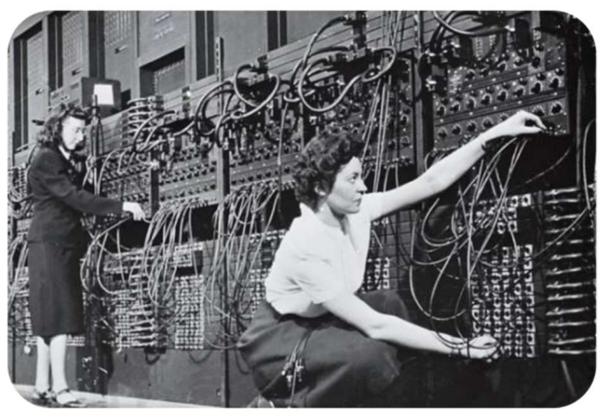
Jean Baptiste Fourier (1768–1830), încercând să rezolve problema încălzirii tunurilor în timpul invaziei lui Napoleon în Egipt, dezvoltă **seriile Fourier**. Fourier a arătat cum funcții periodice pot fi exprimate ca o sumă de funcții sinusoidale, punând bazele analizei Fourier. **Transformata Fourier** a rezultat în urma generalizării, de către același matematician, a **seriilor Fourier**.

Transformata Fourier este continuă, iar aplicațiile sale inițiale erau pentru semnale periodice continue. Apariția calculatoarelor și nevoia analizei semnalelor digitale au condus la **Transformata Fourier Discretă**.

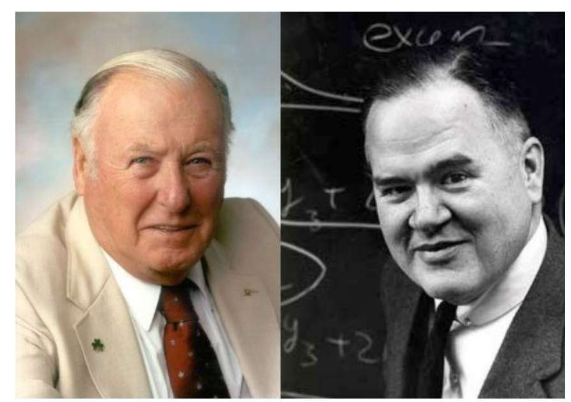
Calculul **Transformatei Fourier Discrete** era complex și necesita o putere de calcul mare, ceea ce făcea **Transformata Fourier Discretă** nefezabilă pentru aplicații practice la scară mare. Astfel, James Cooley și John Tukey, în anii 60, au publicat algoritmul **Transformatei Fourier Rapide** permițând utilizarea eficientă a analizei spectrale în domeniul procesării semnalelor digitale.

Secolul XIX

Anii 1930

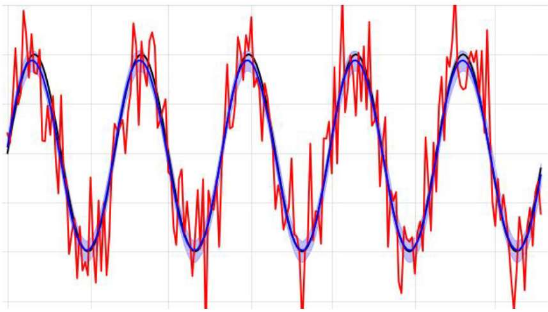


Anii 1960

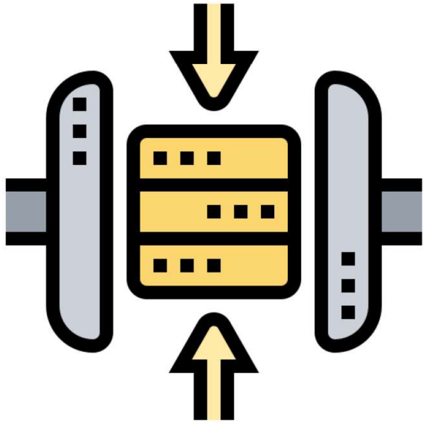


James William Cooley   John Wilder Tukey

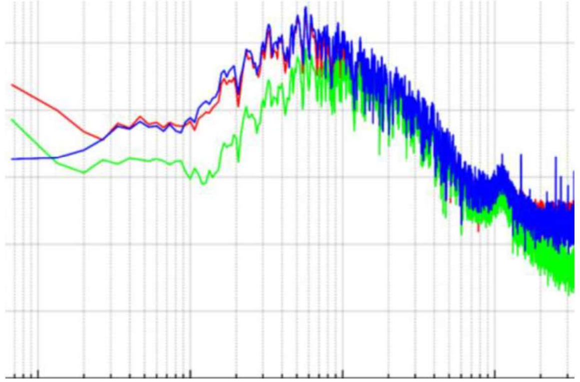
eliminarea zgomotului din semnale



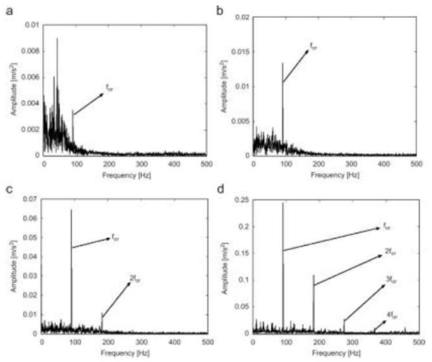
compresie audio / video / imagini



identificarea sursei semnalului



identificarea defectelor în fază incipientă



• • •

Transformata Fourier

Energia unui semnal

Introducere

Spectrograma

Introducere

Energia unui semnal

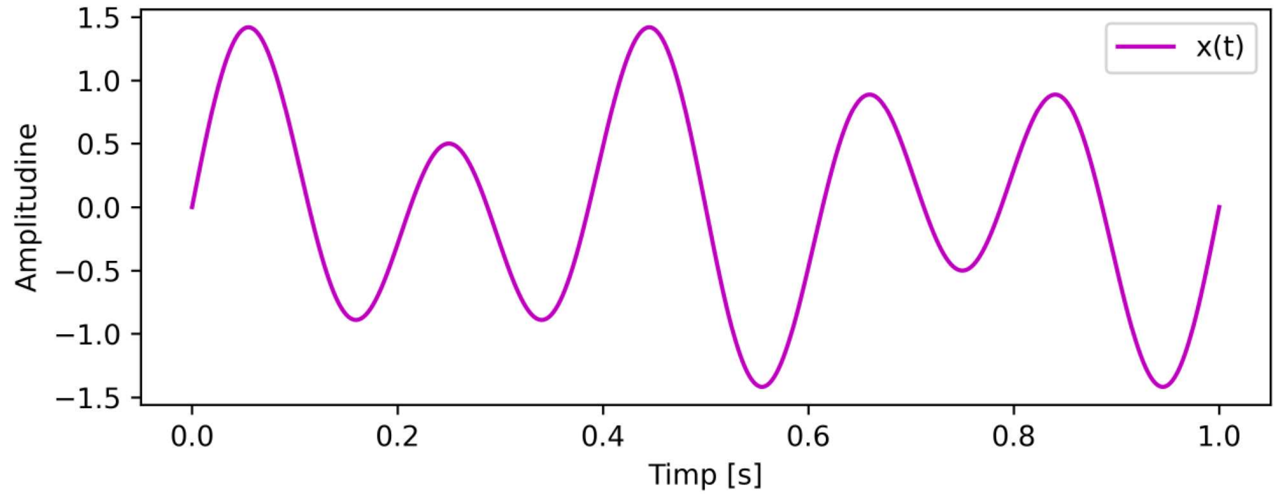
Transformata Fourier

Spectrograma

Definiție

Transformata Fourier transformă semnalul continuu $x(t)$ din domeniul timpului în semnalul continuu $\hat{x}(\omega)$ în domeniul frecvențelor.

$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t) \; + \; 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t)$$

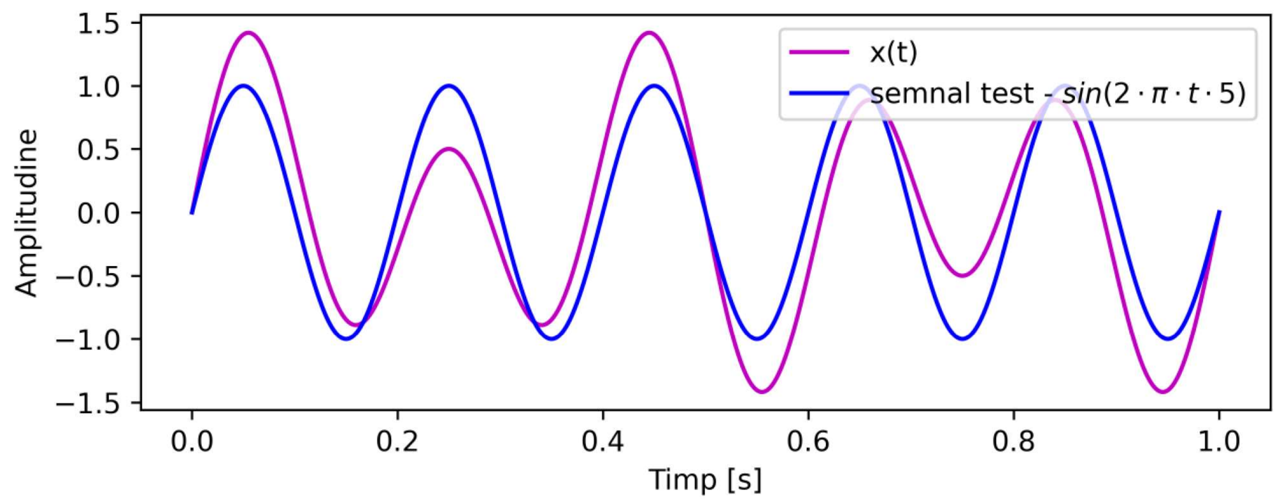


Definiție

Transformata Fourier transformă semnalul continuu $x(t)$ din domeniul timpului în semnalul continuu $\hat{x}(\omega)$ în domeniul frecvențelor.

$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t) + 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t)$$

Cum putem testa dacă avem componenta de 5 Hz în semnalul nostru?



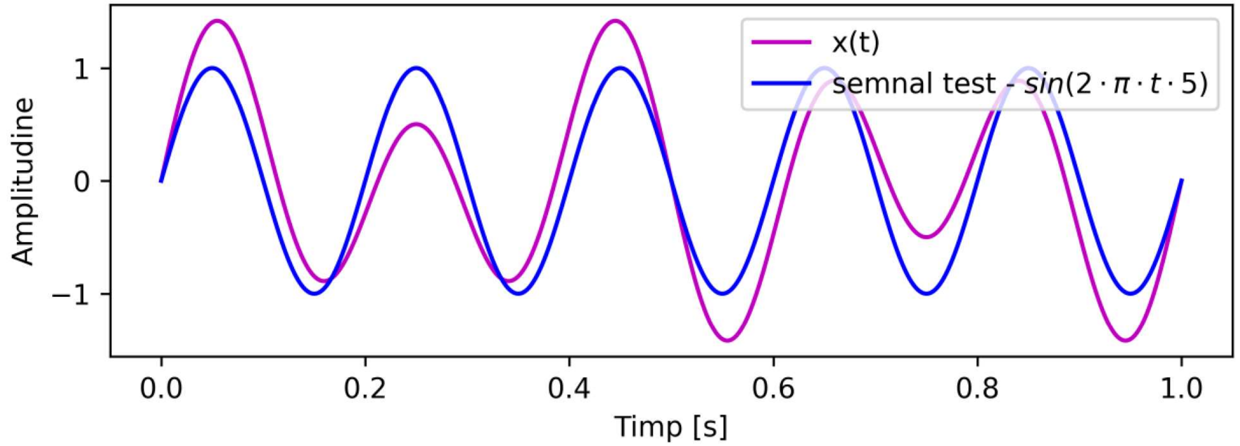
Definiție

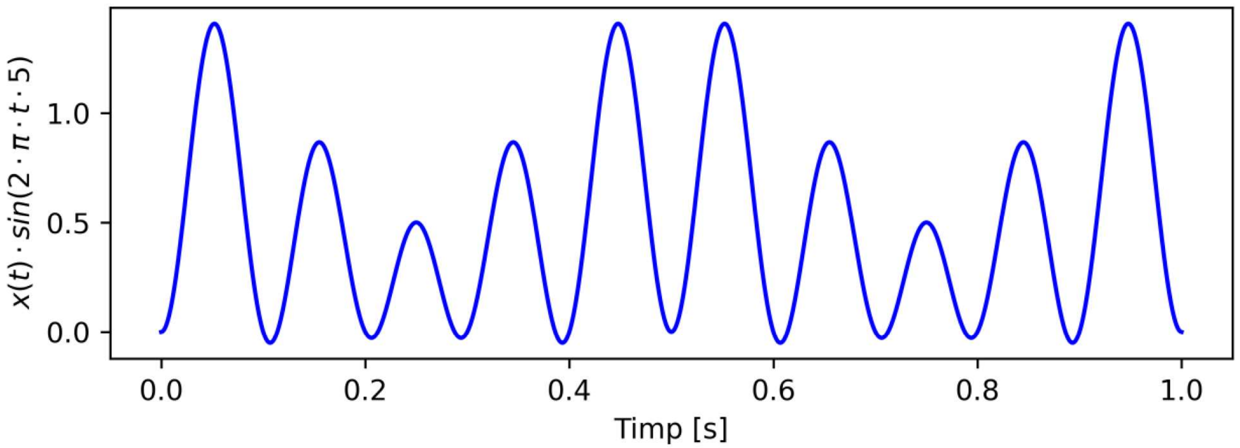
Transformata Fourier transformă semnalul continuu $x(t)$ din domeniul timpului în semnalul continuu $\hat{x}(\omega)$ în domeniul frecvențelor.

$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t) + 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t)$$

Putem înmulți semnalul cu o component de test având frecvența de 5 Hz!

Cum putem obține o singură valoare care să măsoare cât de prezentă este această componentă?



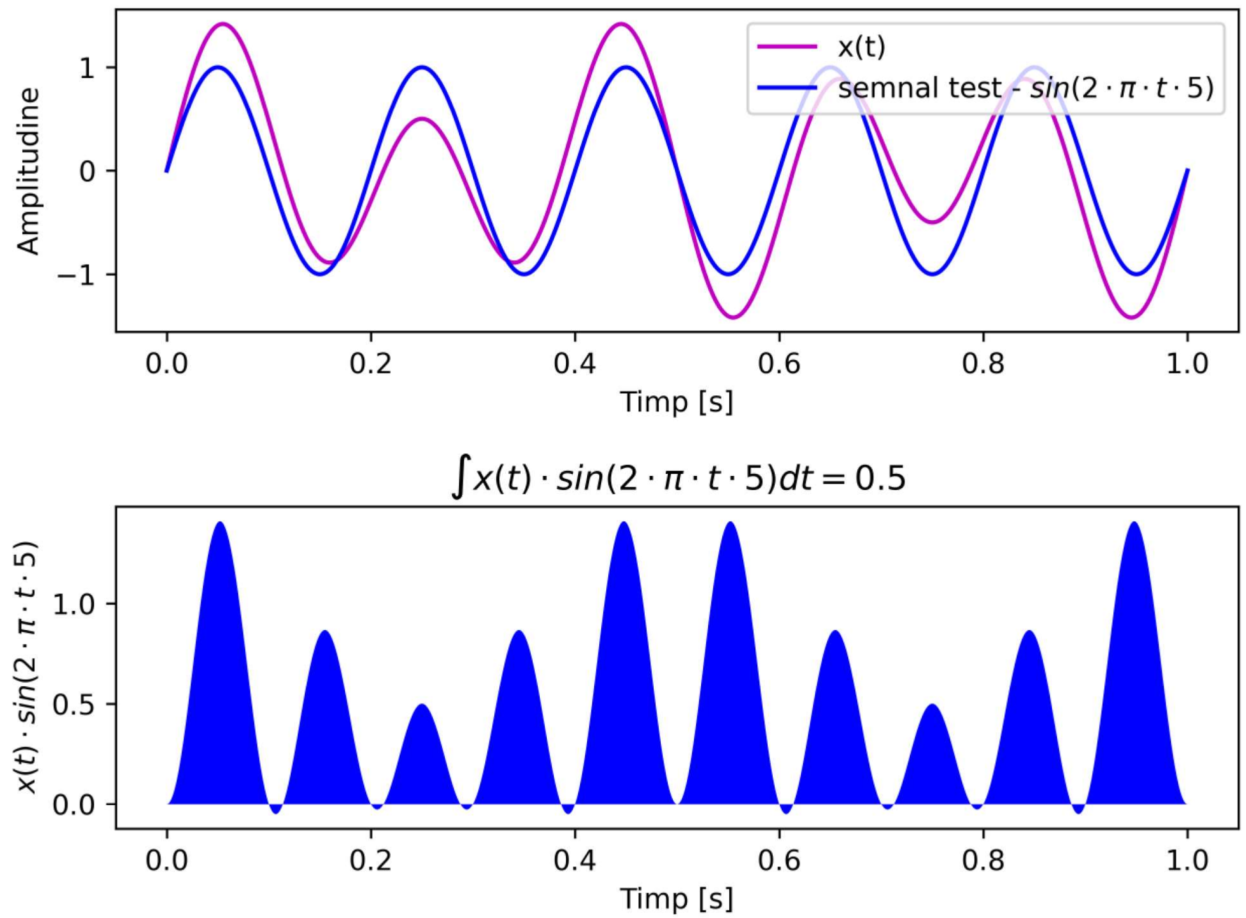


Definiție

Transformata Fourier transformă semnalul continuu $x(t)$ din domeniul timpului în semnalul continuu $\hat{x}(\omega)$ în domeniul frecvențelor.

$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t) + 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t)$$

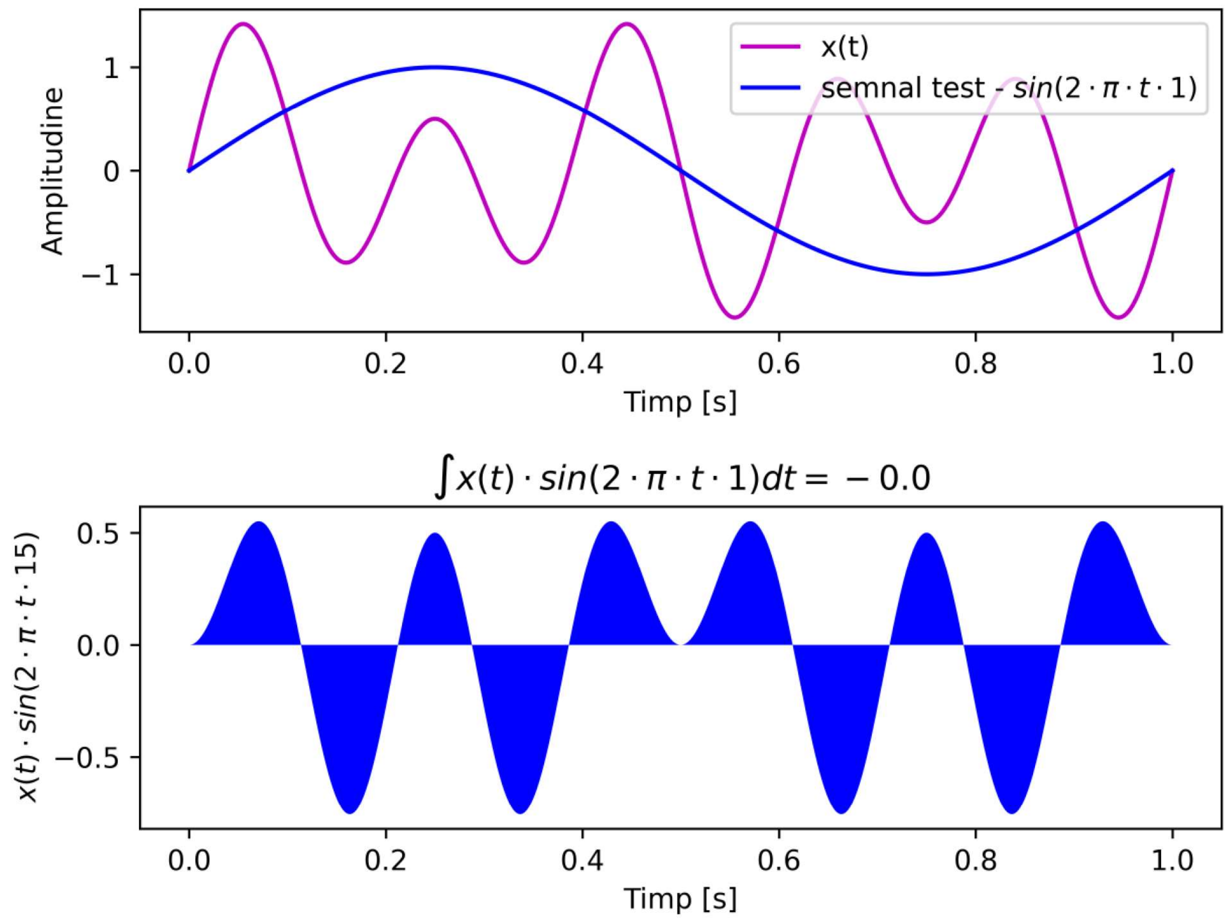
Calculăm aria dintre graficul semnalului obținut în urma înmulțirii, adică integrala acestuia.



Definiție

Transformata Fourier transformă semnalul continuu $x(t)$ din domeniul timpului în semnalul continuu $\hat{x}(\omega)$ în domeniul frecvențelor.

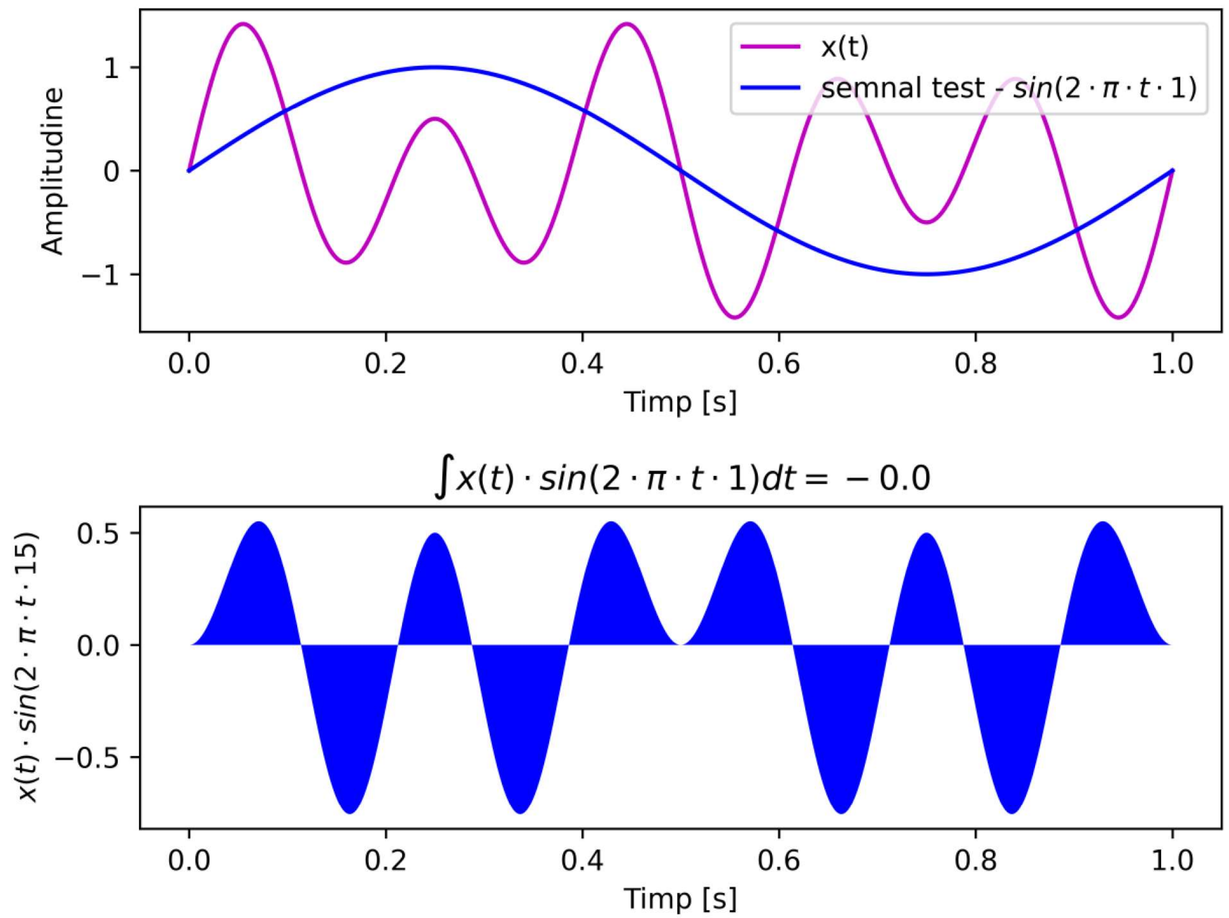
$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t) + 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t)$$



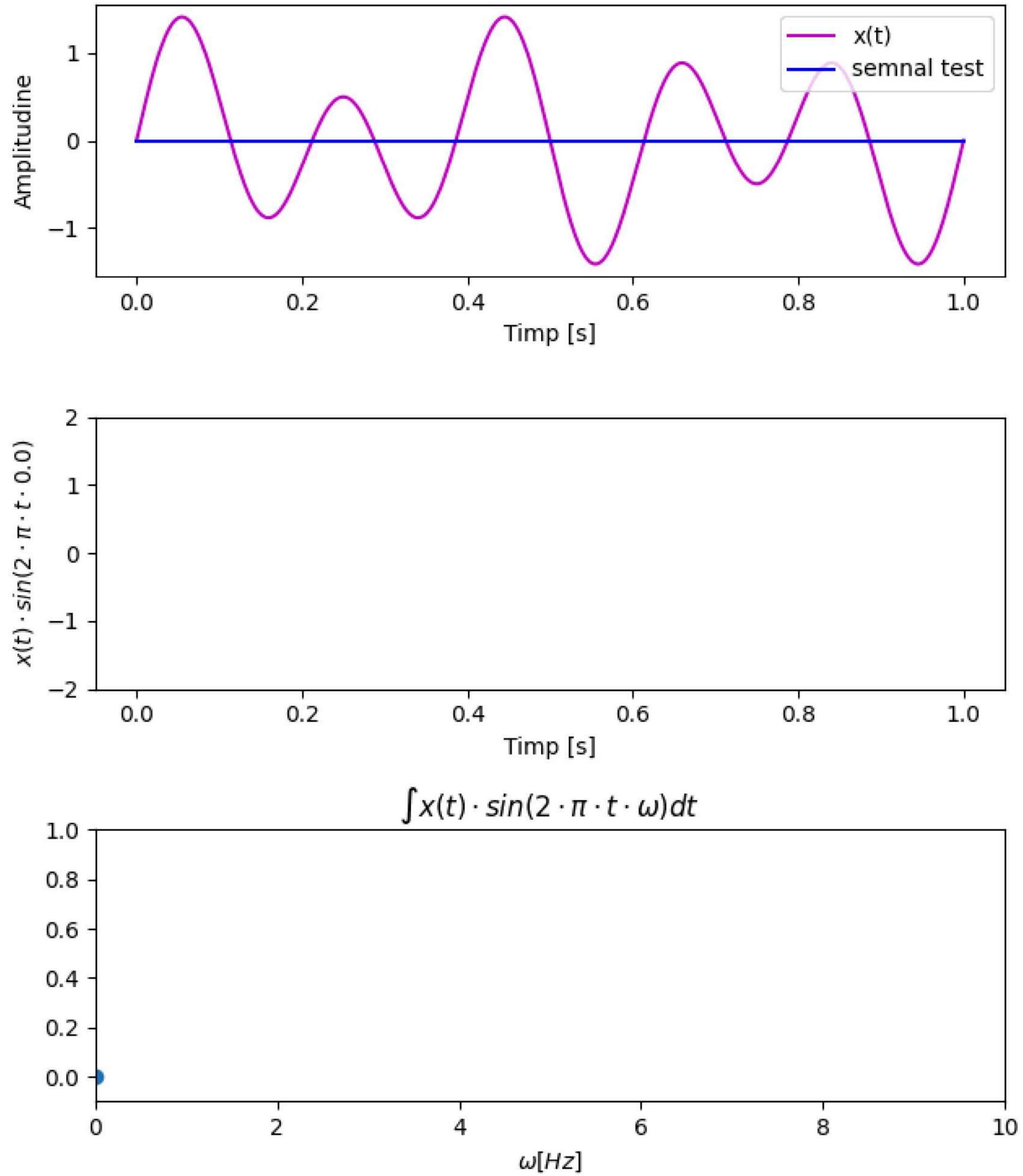
Definiție

Transformata Fourier transformă semnalul continuu $x(t)$ din domeniul timpului în semnalul continuu $\hat{x}(\omega)$ în domeniul frecvențelor.

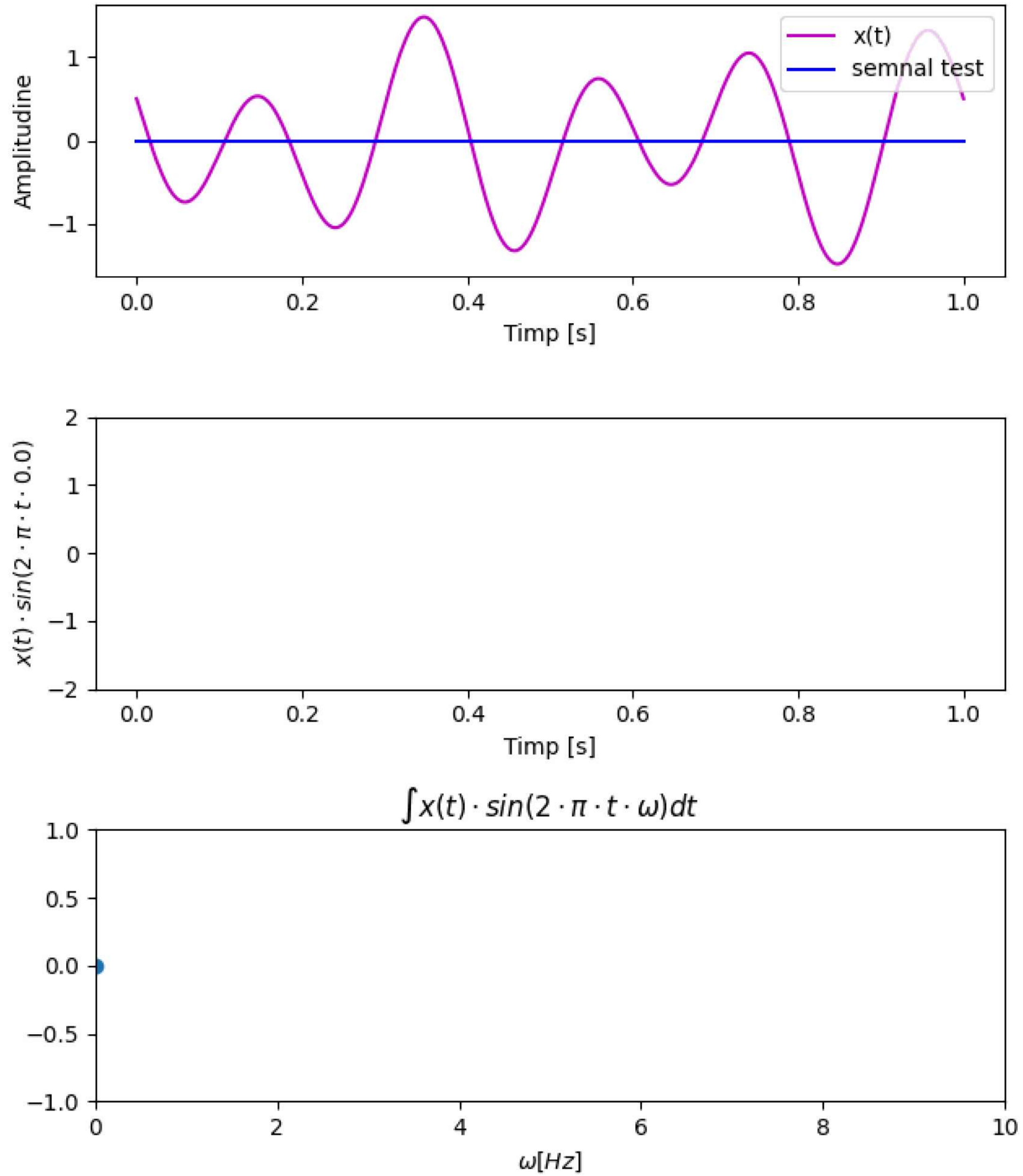
$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t) + 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t)$$



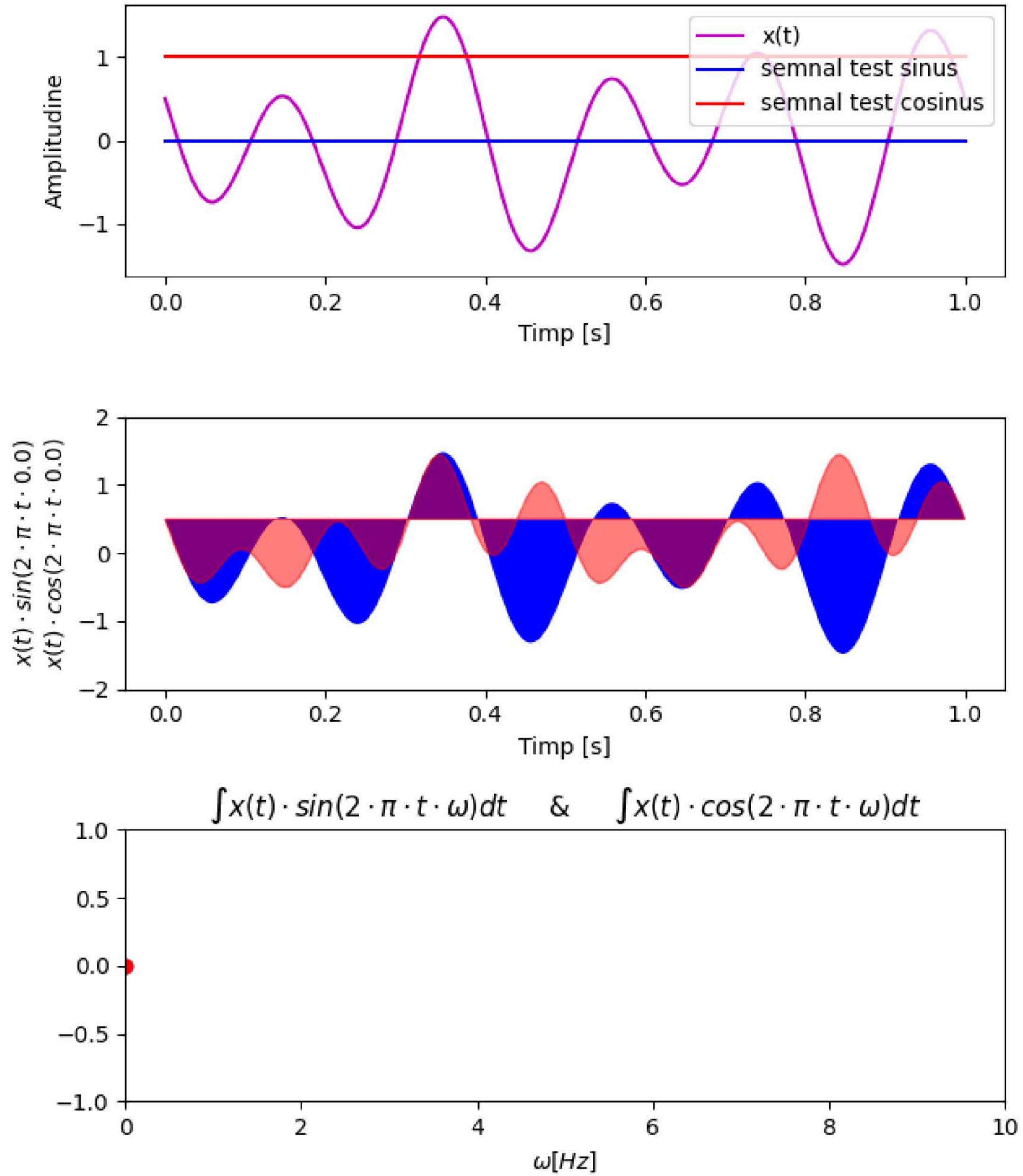
$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t) + 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t)$$



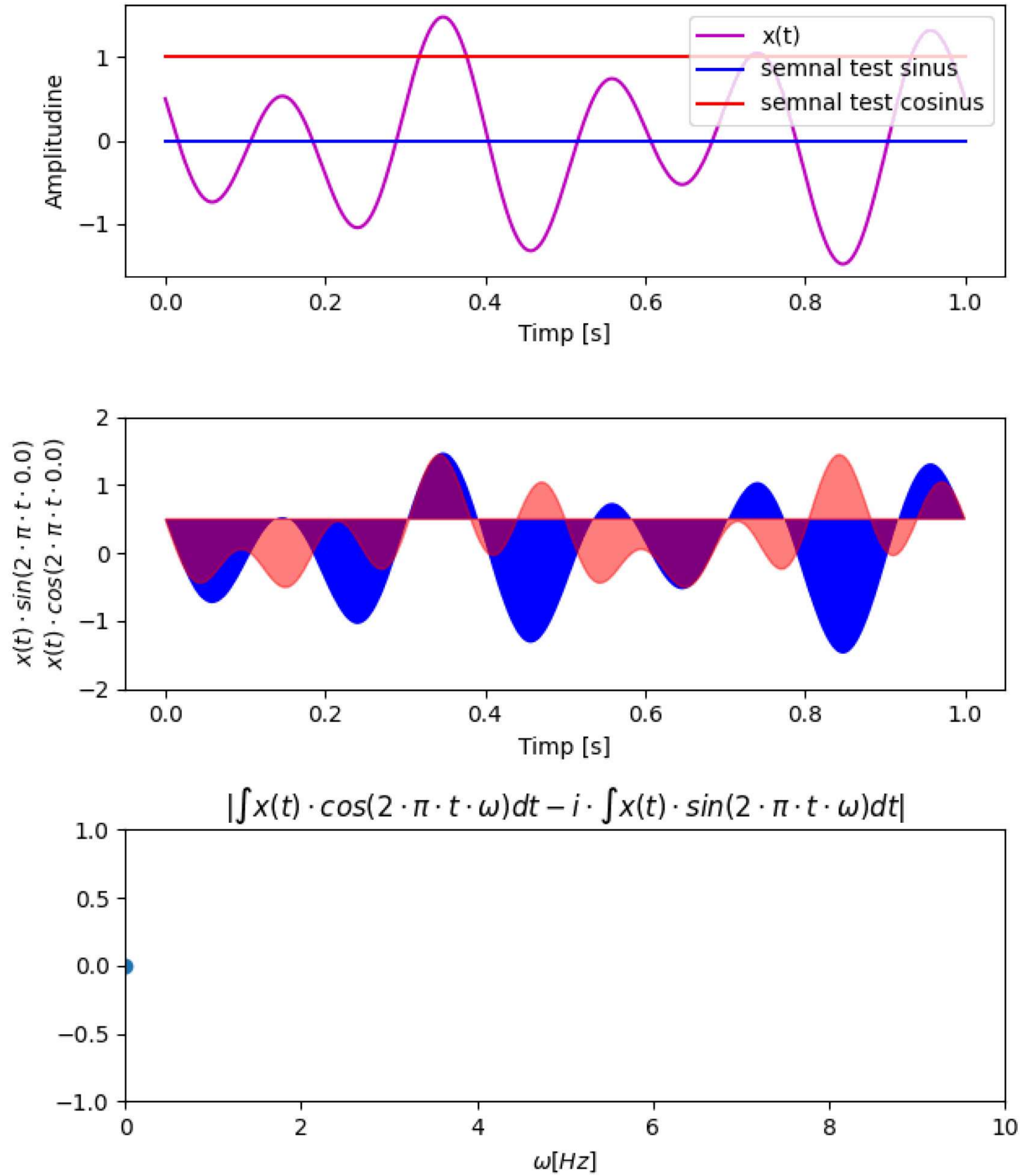
$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t + \pi) \; + \; 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t + \pi/2)$$



$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t + \pi) \ + \ 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t + \pi/2)$$



$$x(t) = sin(2 \cdot \pi \cdot 5 \cdot t + \pi) + 0.5 \cdot sin(2 \cdot \pi \cdot 3 \cdot t + \pi/2)$$



$$\hat{x}(t) = \mathcal{F}\big(x(t)\big) = \int x(t) \cdot cos(2 \cdot \pi \cdot t \cdot \omega)dt \; - i \cdot \int x(t) \cdot sin(2 \cdot \pi \cdot t \cdot \omega)dt$$

Rearanjăm puțin termenii

$$\hat{x}(t) = \mathcal{F}\big(x(t)\big) = \int x(t) \cdot [cos(2 \cdot \pi \cdot t \cdot \omega) \; - i \cdot sin(2 \cdot \pi \cdot t \cdot \omega)]\mathrm{dt}$$

Ne aducem aminte de identitatea lui Euler: $e^{-i \cdot x} = cos(x) \; - i \cdot sin(x)$

$$\boxed{\hat{x}(t) = \mathcal{F}\big(x(t)\big) = \int x(t) \cdot \mathrm{e}^{-i \cdot 2\pi \cdot \omega \cdot t}\mathrm{dt}}$$

TRANSFORMATA FOURIER

- Transformata Fourier $\hat{x}(\omega)$ reprezintă o funcție complexă $\hat{x}(\omega) = Re\{\hat{x}(\omega)\} + i \cdot Im\{\hat{x}(\omega)\}$, ce poate fi scrisă în funcție de modul și fază: $\hat{x}(\omega) = |\hat{x}(\omega)| \cdot e^{i \cdot \varphi(\omega)}$
- Astfel, modulul $|\hat{x}(\omega)|$ reprezintă amplitudinea, iar $\varphi(\omega)$ faza sinusoidei de frecvență ω ce face parte din descompunerea spectrală a semnalului $x(t)$
- Valorile amplitudinilor alcătuiesc **spectrul de amplitudini** al semnalului, iar valorile fazelor **spectrul de faze**.
- Relația inversă de transformare care permite recuperarea unui semnal din spectrul lui este:

$$\boxed{x(t) = \mathcal{F}^{-1}\big(\hat{x}(\omega)\big) = \frac{1}{2\pi} \cdot \int \hat{x}(\omega) \cdot e^{i \cdot 2\pi \cdot \omega \cdot t} d\omega}$$

TRANSFORMATA FOURIER INVERSĂ

$$\hat{x}(t) = \mathcal{F}\big(x(t)\big) = \int x(t) \cdot \mathrm{e}^{-i\cdot 2\pi\cdot\omega\cdot t}\mathrm{dt}$$

TRANSFORMATA FOURIER

$$x(t) = \mathcal{F}^{-1}\big(\hat{x}(\omega)\big) = \frac{1}{2\pi} \cdot \int \hat{x}(\omega) \cdot e^{i\cdot 2\pi\cdot\omega\cdot t} d\omega$$

TRANSFORMATA FOURIER INVERSĂ

- **Transformata Fourier Discretă** este varianta eșantionată a transformatei Fourier continue.
- Pentru un semnal eșantionat $x[n]$ format din N eșantioane:

$$\hat{x}[k] = \mathcal{F}(x[n]) = \sum_{n=0}^{N-1} x[n] \cdot e^{-i\cdot\frac{2\pi}{N}\cdot k\cdot n}, \qquad k = \overline{0, N-1}$$

TRANSFORMATA FOURIER DISCRETĂ

Fiecărui element $\hat{x}[k]$ corespunzându-i o frecvență $\omega_k = k \cdot \frac{f_s}{N}$

$$x[n] = \mathcal{F}^{-1}(\hat{x}[k]) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}[k] \cdot e^{i\cdot\frac{2\pi}{N}\cdot k\cdot n}, \qquad n = \overline{0, N-1}$$

TRANSFORMATA FOURIER DISCRETĂ INVERSĂ

$$\hat{x}[k] = \mathcal{F}(x[n]) = \sum_{n=0}^{N-1} x[n] \cdot e^{-i \cdot \frac{2\pi}{N} \cdot k \cdot n}, \qquad k = \overline{0, N-1}$$

TRANSFORMATA FOURIER DISCRETĂ

- Implementarea software a acestei relații va avea un timp de rulare direct proporțional cu numărul de termeni de tipul $x[n] \cdot e^{-i \cdot \frac{2\pi}{N} \cdot k \cdot n}$ calculați. Câți astfel de termeni se calculează?
- În total se calculează N^2 astfel de termeni (pentru $k = \overline{0, N-1}$ se calculează $\hat{x}[k]$ ca sumă de N termeni).
- Rezultă că timpul de rulare este direct proporțional cu N^2, astfel, timpul de calcul al Transformatei Fourier Discrete crește exponențial cu numărul de eșantioane.
- Exemplu:
  - 1 secundă de semnal audio => 48000 eșantioane => aproximativ $2.3 \cdot 10^9$ termeni de calculat
  - 2 secunde de semnal audio => 96000 eșantioane => aproximativ $9.2 \cdot 10^9$ termeni de calculat
- **Pentru un semnal cu un număr mare de eșantioane, calcularea Transformatei Fourier Discrete nu este fezabilă din punct de vedere computațional.**

$$\hat{x}[k] = \mathcal{F}(x[n]) = \sum_{n=0}^{N-1} x[n] \cdot e^{-i \cdot \frac{2\pi}{N} \cdot k \cdot n}, \qquad k = \overline{0, N-1}$$

TRANSFORMATA FOURIER DISCRETĂ

- **Transformata Fourier Rapidă** este o tehnică ce ne permite să calculăm mai rapid spectrul unui semnal.
- Există mai mulți algoritmi pentru a implementa Transformata Fourier Rapidă.
- Unul din cei mai cunoscuți este **algoritmul cu decimare în frecvență**.
- Este un algoritm de tip **divide et impera** care împarte Transformata Fourier Discretă a unui semnal într-o secvență de Transformate Fourier Discrete ale unor porțiuni de semnal

Fie semnalul cu N eșantioane $x = \{x[0],\ x[1],\ x[2],\ x[3],\ \dots,\ x[N-1]\}$, a cărui reprezentare în domeniul frecvențelor o notăm cu $\hat{x} = \mathcal{F}(x)$.

$$\hat{x}[k] = \mathcal{F}(x[n]) = \sum_{n=0}^{N-1} x[n] \cdot e^{-i \cdot \frac{2\pi}{N} \cdot k \cdot n}, \qquad k = \overline{0, N-1}$$

TRANSFORMATA FOURIER DISCRETĂ

Fie semnalul cu N eșantioane $x = \{x[0],\ x[1],\ x[2],\ x[3],\ \dots,\ x[N-1]\}$, a cărui reprezentare în domeniul frecvențelor o notăm cu $\hat{x} = \mathcal{F}(x)$.

Vom împărți eșantioanele semnalului în două șiruri de eșantioane:

- Eșantioanele pare: $x_1 = \left\{x_1[0],\ x_1[1],\ x_1[2],\ \dots,\ x_1\left[\frac{N}{2}-1\right]\right\} = \{x[0],\ x[2],\ x[4],\ \dots.\}$ cu $\hat{x}_1 = \mathcal{F}(x_1)$.
- Eșantioanele impare: $x_2 = \left\{x_2[0],\ x_2[1],\ x_2[2],\ \dots,\ x_2\left[\frac{N}{2}-1\right]\right\} = \{x[1],\ x[3],\ x[5],\ \dots.\}$ cu $\hat{x}_2 = \mathcal{F}(x_2)$.

Plecând de la Transformata Fourier Discretă:

$$\hat{x}[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i \cdot \frac{2\pi}{N} \cdot k \cdot n} = \sum_{m=0}^{N/2-1} \underbrace{x[2m]}_{x[2m] = x_1[m]} \cdot e^{-i \cdot \frac{2\pi}{N} \cdot k \cdot 2m} + \sum_{m=0}^{N/2-1} \underbrace{x[2m+1]}_{x[2m+1] = x_2[m]} \cdot e^{-i \cdot \frac{2\pi}{N} \cdot k \cdot (2m+1)}$$

$$\hat{x}[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i\cdot\frac{2\pi}{N}\cdot k\cdot n} = \sum_{m=0}^{N/2-1} x[2m] \cdot e^{-i\cdot\frac{2\pi}{N}\cdot k\cdot 2m} + \sum_{m=0}^{N/2-1} x[2m+1] \cdot e^{-i\cdot\frac{2\pi}{N}\cdot k\cdot(2m+1)}$$

$x[2m] = x_1[m]$ $\quad$ $x[2m+1] = x_2[m]$

$$\Rightarrow \quad \hat{x}[k] = \sum_{m=0}^{N/2-1} x_1[m] \cdot e^{-i\cdot\frac{2\pi}{N}\cdot k\cdot 2m} + \sum_{m=0}^{N/2-1} x_2[m] \cdot e^{-i\cdot\frac{2\pi}{N}\cdot k\cdot(2m+1)}$$

$$\Rightarrow \quad \hat{x}[k] = \sum_{m=0}^{N/2-1} x_1[m] \cdot e^{-i\cdot\frac{2\pi}{N/2}\cdot k\cdot m} + \sum_{m=0}^{N/2-1} x_2[m] \cdot e^{-i\cdot\frac{2\pi}{N/2}\cdot k\cdot m - i\cdot\frac{2\pi}{N}\cdot k}$$

$$\Rightarrow \quad \hat{x}[k] = \sum_{m=0}^{N/2-1} x_1[m] \cdot e^{-i\cdot\frac{2\pi}{N/2}\cdot k\cdot m} + \sum_{m=0}^{N/2-1} x_2[m] \cdot e^{-i\cdot\frac{2\pi}{N/2}\cdot k\cdot m} \cdot \overbrace{e^{-i\cdot\frac{2\pi}{N}\cdot k}}^{\text{constant}}$$

Știm că Transformata Fourier Discretă este periodică

$$\Rightarrow \quad \hat{x}[k] = \hat{x}_1\left[k \bmod \frac{N}{2}\right] + e^{-i\cdot\frac{2\pi}{N}\cdot k} \cdot \hat{x}_2\left[k \bmod \frac{N}{2}\right]$$

$$\Rightarrow \quad \hat{x}[k] = \hat{x}_1\left[k \bmod \frac{N}{2}\right] + e^{-i\cdot\frac{2\pi}{N}\cdot k} \cdot \hat{x}_2\left[k \bmod \frac{N}{2}\right]$$

- Se observă că Transformata Fourier Discretă a unui semnal se poate calcula din Transformatele Fourier Discrete ale eșantioanelor pare și ale eșantioanelor impare.
- Astfel, rezultă reducerea numărului de termeni calculați de la N^2 la $2 \cdot \frac{N^2}{4} = \frac{N^2}{2}$, adică la jumătate.
- Dacă pentru a calcula Transformata Fourier Discretă a eșantioanelor pare și a eșantioanelor impare folosim aceeași metodă, și tot așa, până ajungem la un singur eșantion, vom obține o reducere a numărului de termeni calculați de la N^2 la $N \cdot \log_2 N$.
- Acest algoritm funcționează doar pentru un număr de eșantioane N putere a lui 2. Dacă N nu este putere a lui 2 putem fie să eliminăm eșantioane până ajungem la o putere a lui 2 sau să adăugăm eșantioane cu valoarea 0 până ajungem la o putere a lui 2.
- Aplicând același procedeu și pentru **Transformata Fourier Discretă Inversă**, vom obține algoritmul pentru **Transformata Fourier Rapidă Inversă.**

- Pentru a calcula în ***Python*** **Transformata Fourier Rapidă** vom folosi funcția `fft` din sub-modulul `fft` al librăriei **SciPy**.
- Vom determina **valorile spectrului de amplitudini** calculând modulul eșantioanelor complexe ale Transformatei Fourier folosind funcția abs din librăria **NumPy**.
- Vom determina **valorile spectrului de faze** calculând argumentul eșantioanelor complexe ale Transformatei Fourier folosind funcția `angle` din librăria **NumPy**.
- Cele N valori ale spectrului de amplitudini și cele N valori ale spectrului de faze vor corespunde unor frecvențe $\omega = \frac{k \cdot f_s}{N}$ pentru $k = \overline{0 \ldots N-1}$

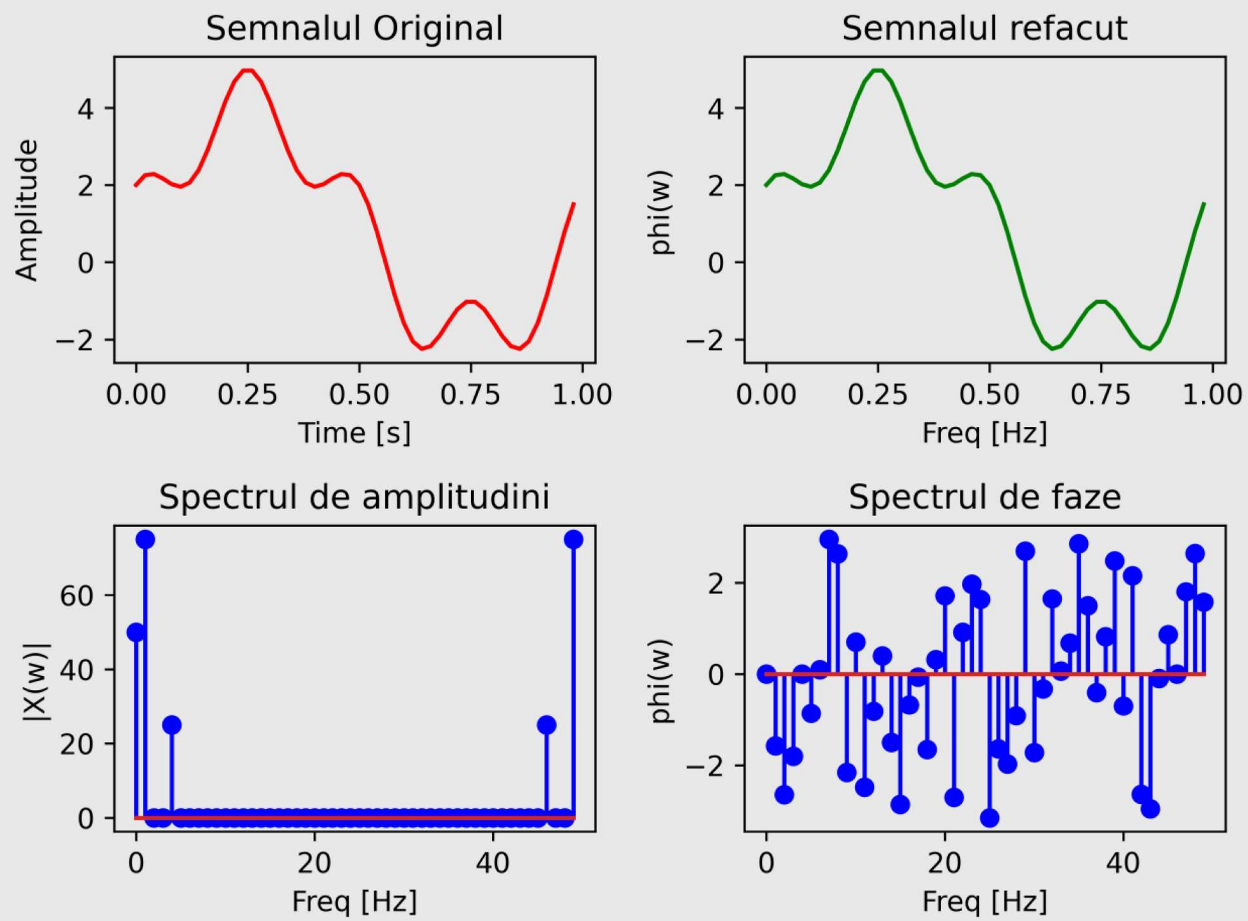
- Pentru a calcula în ***Python*** **Transformata Fourier Rapidă Inversă** vom folosi funcția `ifft` din sub-modulul `fft` al librăriei **SciPy**.
- Atunci când calculăm Transformata Fourier Inversă, trebuie să știm că eșantioanele rezultate vor fi tot numere complexe (dar cu partea imaginară 0), deci va trebui să păstrăm doar partea reală. Pentru aceasta vom folosi funcția `real` din librăria **NumPy**.

Dorim să aplicăm Transformata Fourier Rapidă semnalului x(t)=1+3·sin(2·π·1·t)+sin(2·π·4·t+π/2) eșantionat cu o frecvență $f_s = 50\ Hz$ pe o perioadă de o secundă. Vom încerca apoi să refacem semnalul prin folosirea Transformatei Fourier Rapidă Inversă

main.py

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

tf, fs = 1, 50
t = np.arange(0, tf, 1 / fs)
N = len(t)
x = 1 + 3 * np.sin(2*np.pi*1*t) + np.sin(2*np.pi*4*t+np.pi/2)

plt.subplot(2, 2, 1), plt.plot(t, x, 'r')
plt.xlabel('Time [s]'), plt.ylabel('Amplitude')
plt.title('Semnalul Original')

xhat = scipy.fft.fft(x)
w = np.arange(0, N) * fs / N

plt.subplot(2, 2, 3), plt.stem(w, np.abs(xhat), 'b')
plt.xlabel('Freq [Hz]'), plt.ylabel('|X(w)|')
plt.title('Spectrul de amplitudini')
plt.subplot(2, 2, 4), plt.stem(w, np.angle(xhat), 'b')
plt.xlabel('Freq [Hz]'), plt.ylabel('phi(w)')
plt.title('Spectrul de faze')

plt.subplot(2, 2, 2), plt.plot(t, np.real(scipy.fft.ifft(xhat)), 'g')
plt.xlabel('Freq [Hz]'), plt.ylabel('phi(w)')
plt.title('Semnalul refacut')

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

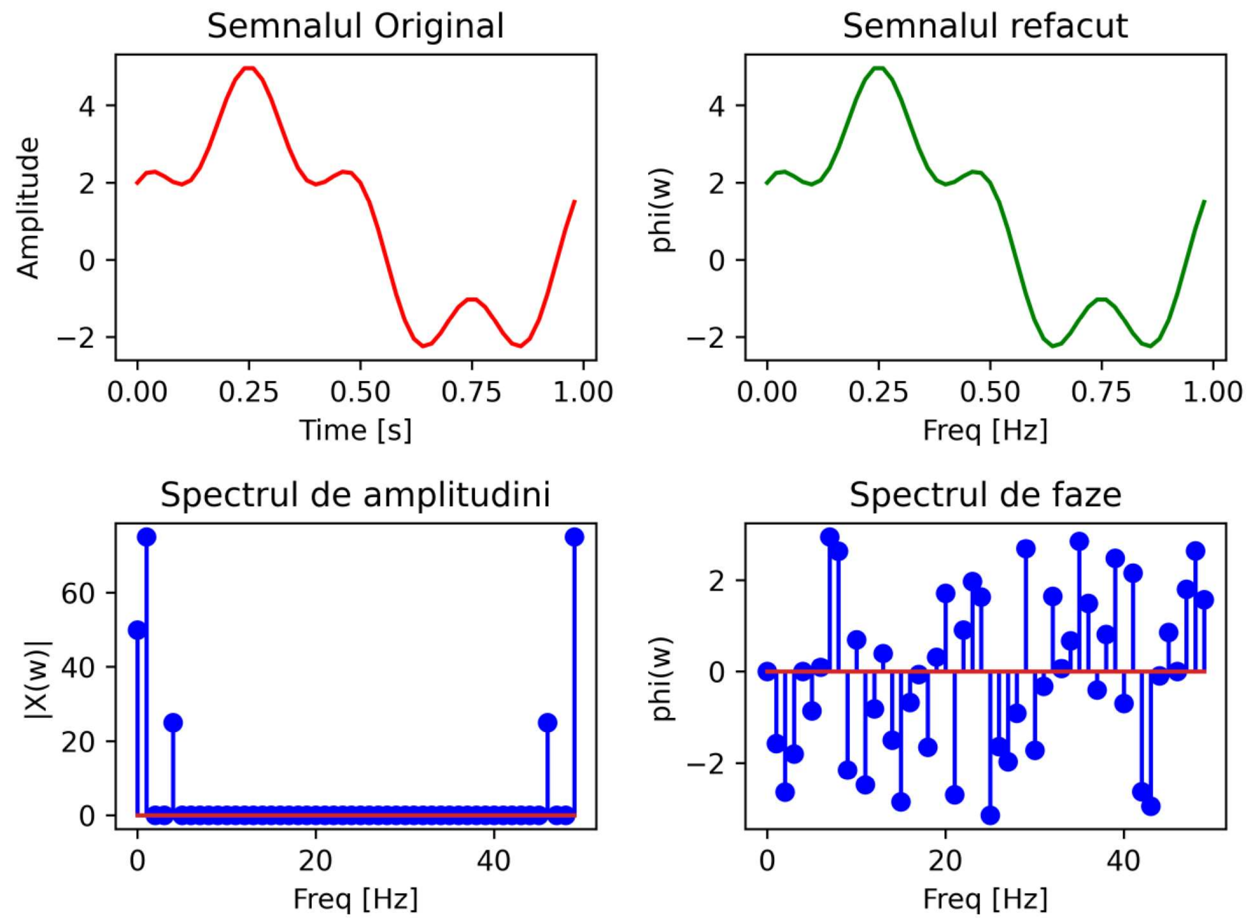
rezultat



Observăm:

- spectrul de amplitudini, exceptând componenta de $0\ Hz$ este simetric față de frecvența Nyquist $\frac{f_s}{2}$. Pentru afișarea grafică, putem ascunde această jumătate.
- amplitudinea componentei continue ($0\ Hz$) are o valoare de N ori mai mare decât amplitudinea componentei continue a semnalului în timp, în timp ce amplitudinile celorlalte componente sunt de N/2 ori mai mari decât amplitudinile sinusoidelor din semnalul original. Astfel, dacă dorim să reprezentăm grafic valorile exacte ale componentelor, se impune o normalizare a amplitudinilor.
- defazajul arată foarte zgomotos (din cauza erorilor de calcul pentru componentele cu amplitudine 0). Va trebui să triem defazajele și să le stabilim 0 pentru frecvențele ale căror amplitudini sunt mai mici de un anumit prag (spre exemplu mai mici de 0.01).
- pentru a afișa corect defazajele sinusoidelor, la defazajele obținute trebuie să adăugăm ${}^{\pi}/_{2}$ pentru $k > 0$.

$$x(t) = 1 + 3 \cdot \sin(2 \cdot \pi \cdot 1 \cdot t) + \sin(2 \cdot \pi \cdot 4 \cdot t + \pi/2)$$

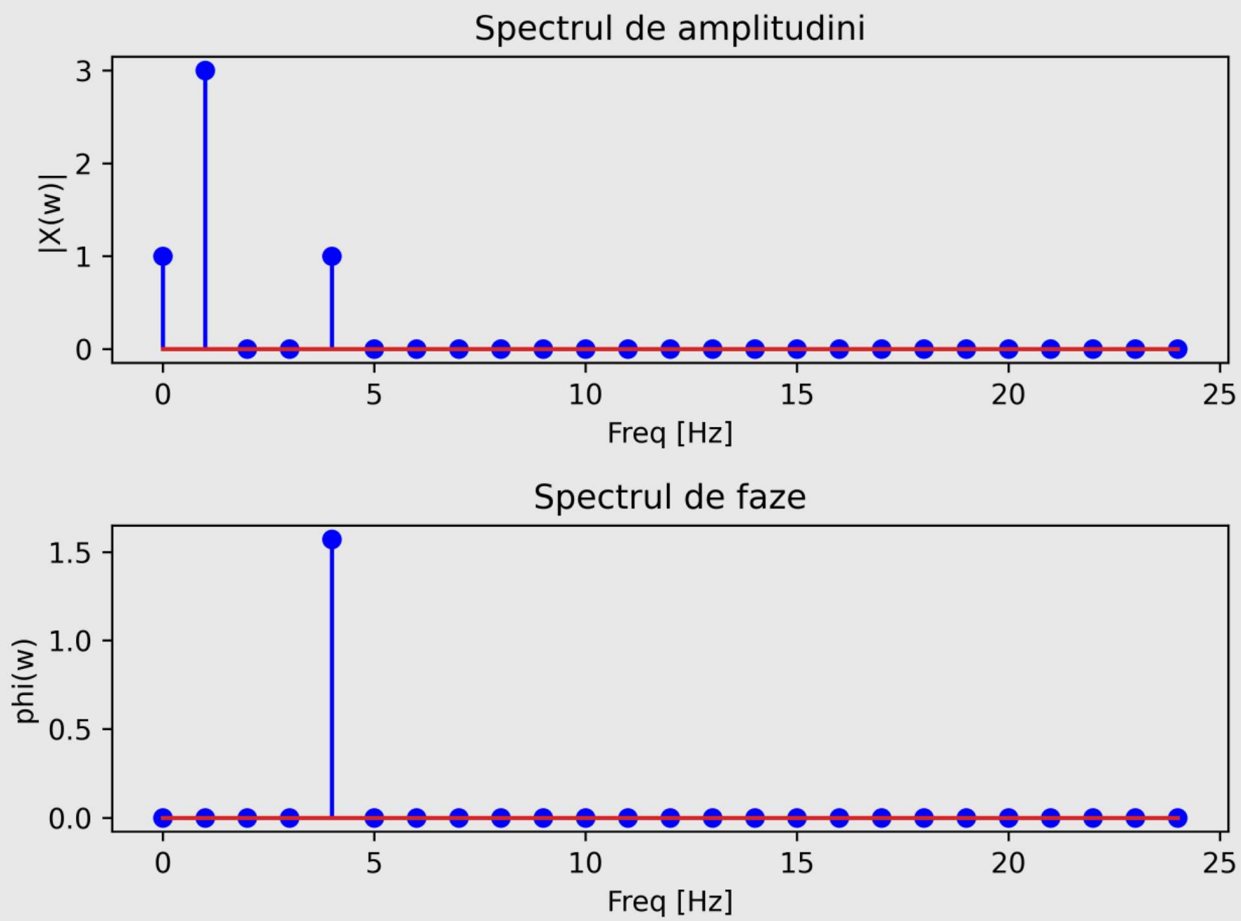


Dorim să aplicăm Transformata Fourier Rapidă semnalului x(t)=1+3·sin(2·π·1·t)+sin(2·π·4·t+π/2) eșantionat cu o frecvență $f_s = 50$ Hz pe o perioadă de o secundă.

main.py

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

tf, fs = 1, 50
t = np.arange(0, tf, 1 / fs)
N = len(t)
x = 1 + 3 * np.sin(2*np.pi*1*t) + np.sin(2*np.pi*4*t+ np.pi/2)

xhat = scipy.fft.fft(x)

# folosim prima jumătate a spectrului
w = np.arange(0, int(N / 2)) * fs / N
xhat_amp = np.abs(xhat[:int(N / 2)])
xhat_phi = np.angle(xhat[:int(N / 2)])

xhat_amp *= 2 / N
xhat_amp[0] /= 2  # normalizăm valorile
plt.subplot(2, 1, 1), plt.stem(w, xhat_amp, 'b')
plt.xlabel('Freq [Hz]'), plt.ylabel('|X(w)|')
plt.title('Spectrul de amplitudini')

xhat_phi[1:] += np.pi / 2 # adaugam pi/2 pentru w > 0
xhat_phi[xhat_amp < 0.01] = 0 # triem defajazele

plt.subplot(2, 1, 2), plt.stem(w, xhat_phi, 'b')
plt.xlabel('Freq [Hz]'), plt.ylabel('phi(w)')
plt.title('Spectrul de faze')

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

rezultat

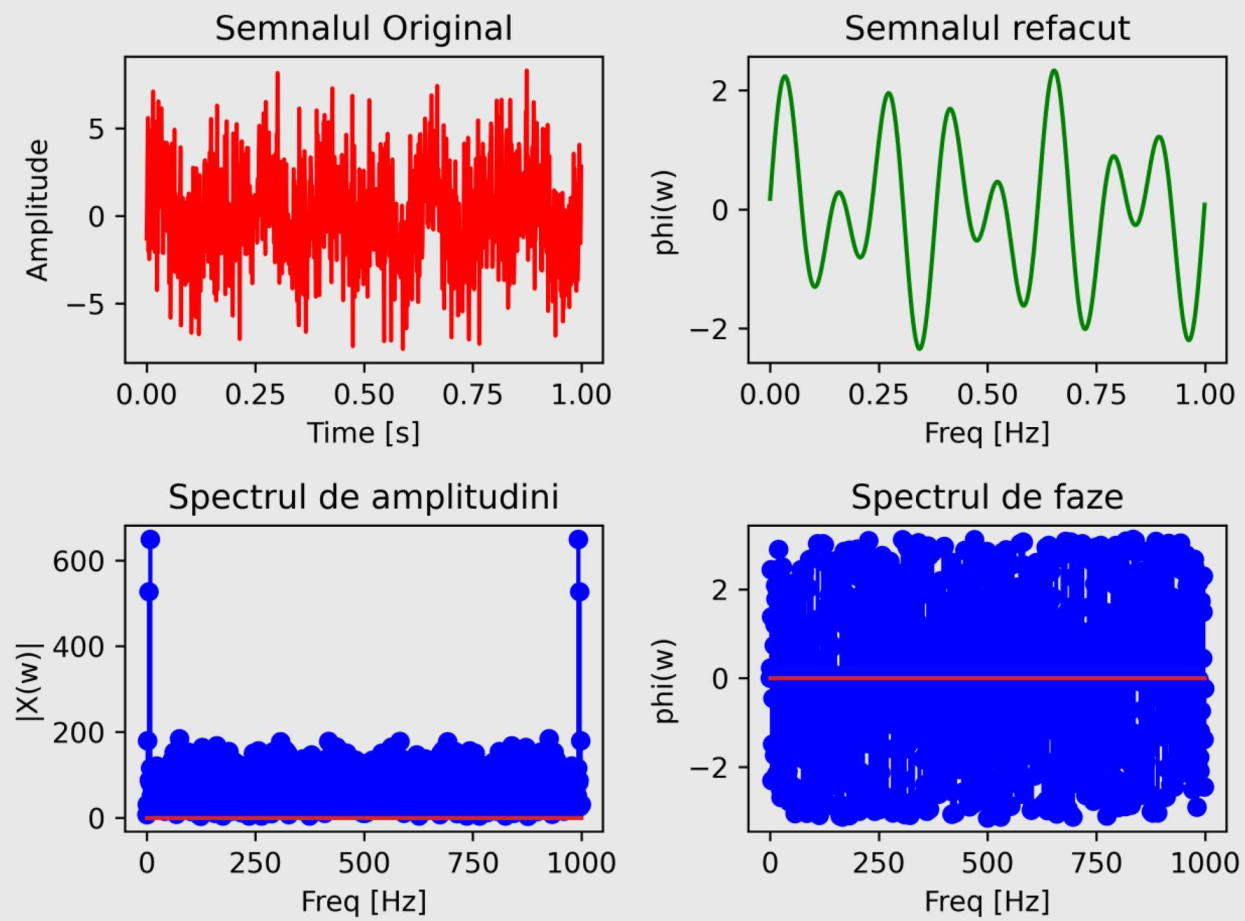


Exemplu de utilizare a Transformatei Fourier pentru eliminarea zgomotului alb dintr-un semnal

main.py

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

tf, fs = 1, 1000
t = np.arange(0, tf, 1 / fs)
N = len(t)
x = np.sin(2*np.pi*5*t) + np.sin(2*np.pi*8*t) + 2.5 * np.random.randn(N)

plt.subplot(2, 2, 1), plt.plot(t, x, 'r')
plt.xlabel('Time [s]'), plt.ylabel('Amplitude')
plt.title('Semnalul Original')

xhat = scipy.fft.fft(x)
w = np.arange(0, N) * fs / N

plt.subplot(2, 2, 3), plt.stem(w, np.abs(xhat), 'b')
plt.xlabel('Freq [Hz]'), plt.ylabel('|X(w)|')
plt.title('Spectrul de amplitudini')
plt.subplot(2, 2, 4), plt.stem(w, np.angle(xhat), 'b')
plt.xlabel('Freq [Hz]'), plt.ylabel('phi(w)')
plt.title('Spectrul de faze')

# eliminare frecvente de amplitudine mica - eliminare zgomot alb
xhat[np.abs(xhat) < 300] = 0
plt.subplot(2, 2, 2), plt.plot(t, np.real(scipy.fft.ifft(xhat)), 'g')
plt.xlabel('Freq [Hz]'), plt.ylabel('phi(w)')
plt.title('Semnalul refacut')

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

rezultat



Introducere

Energia unui semnal

Transformata Fourier

Spectrograma

Introducere

Transformata Fourier

Energia unui semnal

Spectrograma

Definiție

Energia unui semnal continuu $f(t)$ se definește ca aria de sub pătratul magnitudinii semnalului: $E_s = \int_{-\infty}^{+\infty} |f(t)|^2 dt$. Pentru că lucrăm cu semnale discrete în timp cu un număr finit de eșantioane, formula matematică a **energiei unui semnal discret** devine o sumă finită: $E_s = \sum_{n=0}^{N-1} |f[n]|^2$.

Teoremă

Teorema lui Parceval: Trecerea din domeniul timp în domeniul frecvențelor utilizând Transformata Fourier nu modifică energia semnalului: $\sum_{n=0}^{N-1} |f[n]|^2 = \sum_{k=0}^{N-1} \left|\hat{f}[k]\right|^2$

- Această teoremă are implicații majore în compresia semnalelor (audio, imagine, video).

- Știind că energia semnalului în domeniul timp este egală cu energia semnalului în domeniul frecvențelor deducem că, dacă aducem o modificare minoră energiei în domeniul frecvențelor, ea se va manifesta ca o modificare minoră în domeniul timp.

Teoremă

Teorema lui Parceval: Trecerea din domeniul timp în domeniul frecvențelor utilizând Transformata Fourier nu modifică energia semnalului: $\sum_{n=0}^{N-1}|f[n]|^2 = \sum_{k=0}^{N-1}\left|\hat{f}[k]\right|^2$

- Această teoremă are implicații majore în compresia semnalelor (audio, imagine, video).

- Știind că energia semnalului în domeniul timp este egală cu energia semnalului în domeniul frecvențelor deducem că, dacă aducem o modificare minoră energiei în domeniul frecvențelor, ea se va manifesta ca o modificare minoră în domeniul timp.

Exemplu

Un semnal audio de 10 secunde eșantionat cu o frecvență de eșantionare de 48 kHz necesită pentru stocare un spațiu de aproximativ 400 KB. Dacă aplicăm Transformata Fourier semnalului nostru, vom observa că nu toate componentele spectrale au o amplitudine considerabilă, deci nu toate contribuie în mod egal la energia semnalului. Astfel, putem renunța la 80% din componentele spectrale fără a pierde prea mult din energia semnalului, cu alte cuvinte fără a pierde din calitatea semnalului audio. Astfel, pentru stocarea fișierului audio putem stoca semnalul în domeniul frecvențelor (doar cele mai considerabile 20% componente), reducând astfel considerabil utilizarea spațiului de stocare. Acesta este algoritmul care stă la baza formatului MP3.

Introducere

Transformata Fourier

Energia unui semnal

Spectrograma

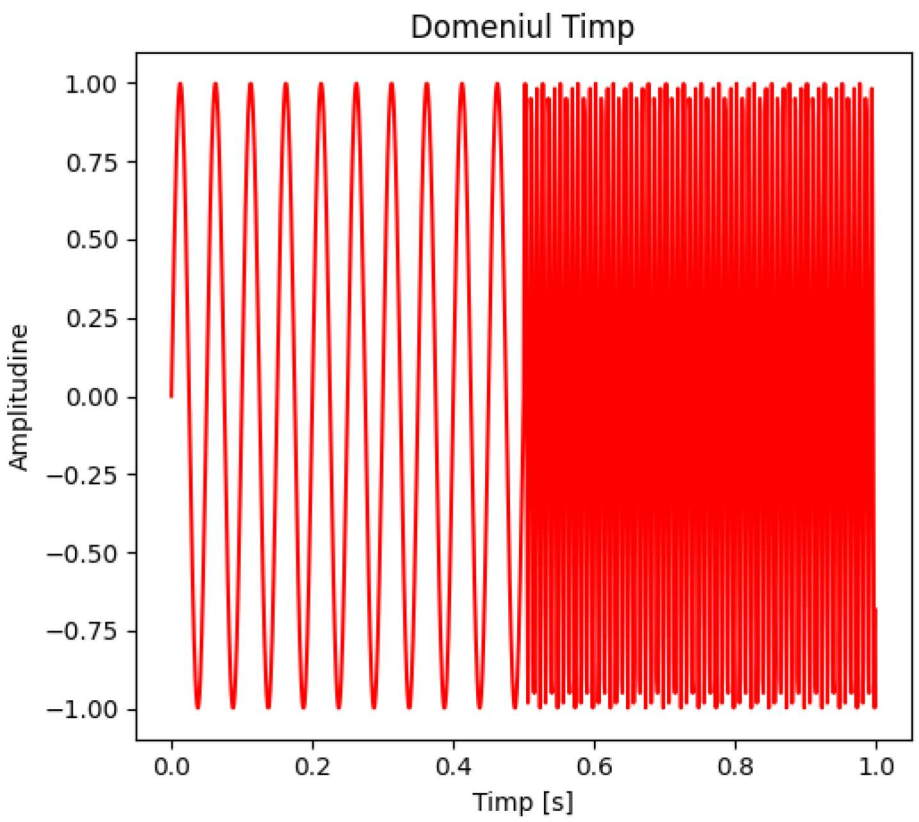
Introducere

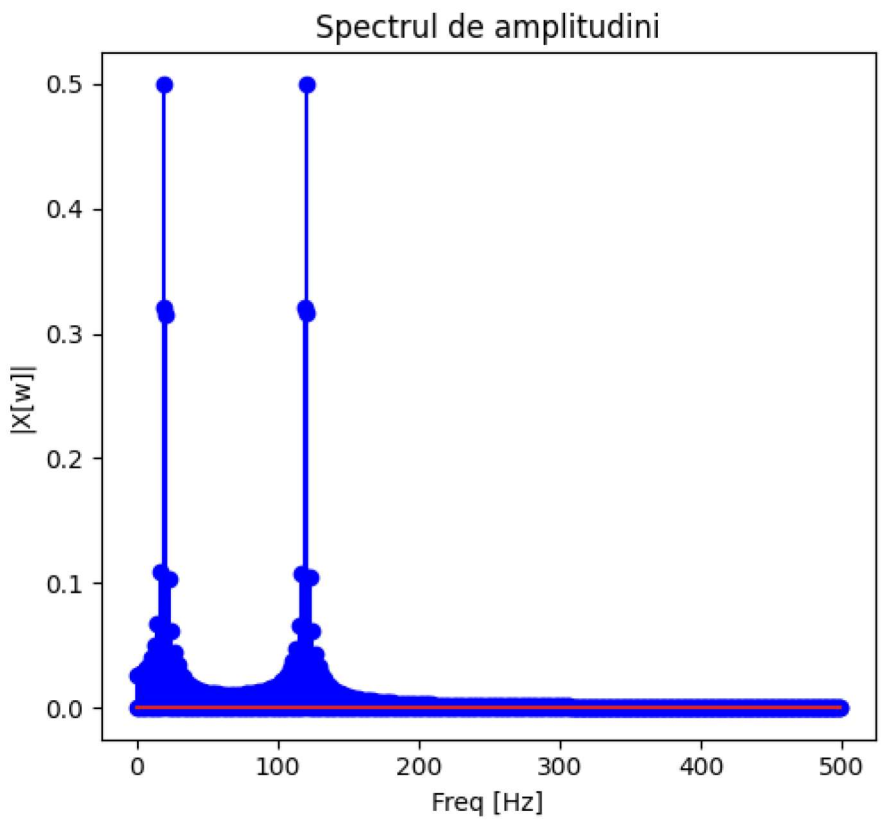
Transformata Fourier

Energia unui semnal

Spectrograma

- Fie un semnal sinusoidal cu amplitudinea 1 eșantionat cu frecvența de $f_s = 1\ kHz$ pe o perioadă de $1s$ a cărui frecvență este de $20\ Hz$ pentru prima jumătate de secundă și $120\ Hz$ pentru cea de-a doua jumătate. Reprezentarea grafică a semnalului în domeniul timp și în domeniul frecvențelor:





- Dacă analizăm spectrul de amplitudini se pot observa cele două componente spectrale (totuși cu amplitudinea înjumătățită datorită faptului ca fiecare din ele se manifestă doar pe jumătate din perioadă).
- Domeniul frecvențelor nu ne sugerează nimic în privința evoluției componentelor spectrale ale semnalului în timp.

Definiție

Spectrograma unui semnal este o reprezentare a spectrului de frecvențe al semnalului pe măsură ce semnalul evoluează în timp.

- Pentru a realiza spectrograma unui semnal, se folosește o fereastra de NFFT eșantioane care este plimbată în timp peste semnalul nostru.
- La fiecare moment de timp se calculează FFT pentru porțiunea de semnal de sub fereastra.
- În final rezultă o serie de spectre care vor fi apoi afișate într-o manieră grafică.
- Pentru reprezentarea spectrogramei în Python vom folosi funcția `specgram` din sub-modulul `pyplot` al librăriei **Matplotlib**, dând ca parametri: semnalul x, frecvența de eșantionare, numărul de eșantioane al fiecărei ferestre și încă doi parametri pentru afișarea amplitudinii: `mode='magnitude'`, `scale='linear'`.

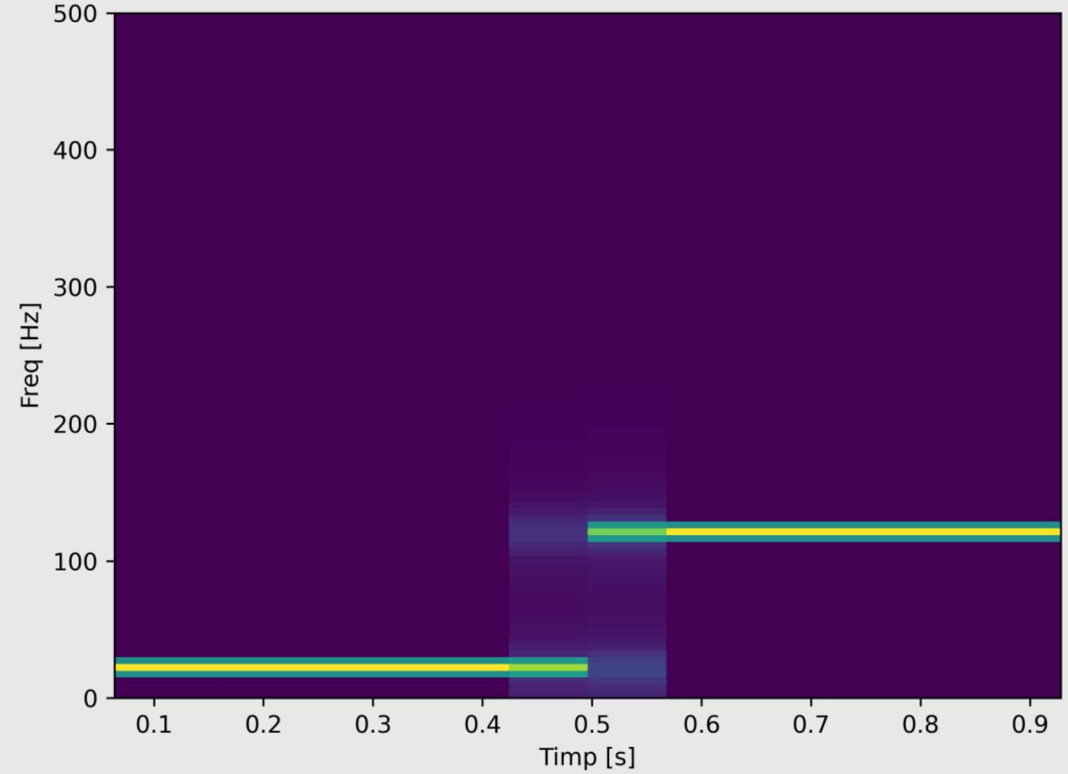
1. Introducere | 2. Transformata Fourier | 3. Energia unui semnal | 4. Spectrograma

EXEMPLU

main.py

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

tf, fs = 1, 1000
t = np.arange(0, tf, 1 / fs)
N = len(t)
x1 = np.sin(2 * np.pi * 20 * t[:int(N / 2)])
x2 = np.sin(2 * np.pi * 120 * t[int(N / 2):])
x = np.hstack((x1, x2))

# NFFT numărul de eșantioane al fiecărei ferestre
plt.specgram(x, Fs=fs, NFFT=int(fs / 5), mode='magnitude', scale='linear')
plt.xlabel('Timp [s]’)
plt.ylabel('Freq [Hz]')
plt.tight_layout()
plt.show()
```

rezultat



- Se observă clar cele două componente spectrale și trecerea de la o frecvență la alta la momentul $t = 0.5\ s$.

- În cazul spectrogramelor se aplică principiul incertitudinii: cu cât dorim să aflăm mai multe informații în domeniul timp, cu atât vom afla mai puține informații în domeniul frecvențelor, și viceversa. Adică, scăderea dimensiunii ferestrei (o granularitate mai mare, și implicit mai bună în timp) duce la o scădere a granularității în domeniul frecvențelor.

Introducere

Transformata Fourier

Energia unui semnal

Spectrograma

Introducere

Transformata Fourier

Energia unui semnal

Spectrograma