Università degli Studi di Perugia

Dipartimento di Matematica e Informatica

Report di Laboratorio – Tecniche di Acquisizione Dati

Trasformata e Anti trasformata

Studenti

Nicolò Tittarelli

Leonardo Nicoletta

Filippo Notari

Anno Accademico 2023-2024

Sommario

Introduzione

La relazione in esame si concentra completamente sulla teoria dei segnali e in particolare sullo studio di un generico segnale periodico attraverso le equazioni di sintesi e di analisi, successivamente si introduce il concetto di sviluppo in serie di Fourier per segnali periodici. L'obiettivo di questa esperienza è quello di fornire una dimostrazione dello studio di diversi segnali periodici evidenziando quindi le differenze tra uno studio nel dominio del tempo e uno nel dominio delle frequenze e come avviene il passaggio dall'uno all'altro. La relazione è composta da due capitoli: il primo fornisce alcuni concetti di base della teoria dei segnali e successivamente si passa alla spiegazione dello sviluppo in serie di Fourier per segnali periodici; il secondo capitolo si occupa dello studio di tre tipi di segnali concentrandosi poi sulla corrispondente trasformata di Fourier. Per quanto riguarda l'aspetto pratico di tutto ciò che viene trattato è stato scelto come linguaggio di programmazione Python mentre tutte le librerie verranno menzionate all'interno di ogni capitolo.

Capitolo 1. Equazione di sintesi di un segnale periodico

1.1 Teoria dei segnali e sviluppo in serie di Fourier

La seguente relazione si concentrerà sullo studio di un tipo particolare di segnali chiamati segnali periodici e continui, possiamo quindi dare una definizione di segnale e successivamente di segnale periodico: un segnale è una grandezza fisica variabile nel tempo a cui è associata un'informazione di interesse, facciamo presente che si distingue dal rumore in quanto quest'ultimo non contiene nessun dato utile. Possiamo passare alla definizione di segnale periodico: un segnale si dice periodico se i suoi valori si ripetono ciclicamente ad intervalli di tempo regolari e tali intervalli sono multipli del periodo fondamentale del segnale. Un segnale periodico ha la seguente forma:

$$x(t) = x(t + T)\ \forall\ t$$

Un esempio di segnale periodico è la sinusoide:

$$x(t) = A\cos(w_0 t + \varphi)$$

Gli elementi principali che caratterizzano i segnali periodici sono i seguenti: il periodo ossia il tempo, espresso in secondi, impiegato dal segnale a compiere un'oscillazione completa quindi il tempo durante il quale il segnale si ripete, viene poi definita l'ampiezza ovvero la differenza tra il punto massimo e quello minimo sull'asse delle ordinate e infine si ha la frequenza che indica il numero di periodi ripetuti in un intervallo temporale e viene misurata in Hertz. Tutti questi elementi sono presenti nella formula mostrata sopra e li possiamo visualizzare in questo elenco:

- A è l'ampiezza
- $w_0 = 2\pi f = \frac{2\pi}{T}$ si tratta della velocità angolare
- dalla formula precedente riusciamo ad ottenere la frequenza come $f = \frac{w}{2\pi} = \frac{1}{T}$
- t è il tempo
- φ è la fase

Lo sviluppo in serie di Fourier permette di rappresentare funzioni, quindi segnali, reali e periodici, come combinazione lineare di funzioni sinusoidali; ognuna di queste sinusoidi prende il nome di armonica e l'insieme delle armoniche viene detto spettro del segnale. L'obiettivo dello sviluppo in serie di Fourier è anche quello di facilitare lo studio dei segnali dato che questi nelle applicazioni pratiche non si presentano mai come sinusoidi o altri segnali elementari. Per capire meglio questo concetto è bene fare riferimento immediatamente alla forma polare della serie di Fourier:

$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k \cos(2\pi k f_0 t + \theta_k)$$

questa formula evidenzia come il segnale in esame x(t) venga espresso come sovrapposizione di infinite armoniche anche se poi nelle applicazioni è necessario porre un limite a tale sommatoria data la finitezza del calcolatore.

A partire da tale formula e grazie all'utilizzo della formula di Eulero

$$\cos(x) = \frac{e^{ix} + e^{-ix}}{2}$$

riusciamo ad ottenere la forma complessa dello sviluppo in serie di Fourier chiamata anche equazione di sintesi:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{i2\pi k f_0 t}$$

Tale equazione permette di ricostruire il segnale a partire dalle sue armoniche, conoscendo a priori i coefficienti di Fourier ossia i termini denominati X_k.

A questo punto possiamo procedere con lo studio di un segnale periodico, in particolare si è scelto di analizzare il segnale onda quadra. L'obiettivo di questa analisi è quello di comprendere la qualità con la quale viene ricostruito il segnale scegliendo in modo arbitrario il numero di coefficienti di Fourier da utilizzare; naturalmente tale qualità aumenta all'aumentare del numero di coefficienti utilizzati.

Nello specifico abbiamo a disposizione la seguente equazione di sintesi:

$$x(t) = \frac{1}{2} + \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{1}{k} (-1)^{\frac{k-1}{2}} \cos(2\pi k f_0 t)$$

devono poi essere determinati una serie di parametri che sono:

- frequenza di campionamento
- tempo di osservazione del segnale
- frequenza del segnale
- numero di coefficienti

una volta stabiliti questi parametri possiamo passare al calcolo della sommatoria in determinati istanti di tempo.

1.2 Studio nel dominio temporale di un segnale

Questa sezione è dedicata all'analisi del codice scritto in Python per realizzare ciò che è stato descritto nella sezione precedente.

Le librerie utilizzate sono:

- numpy, per le diverse funzioni matematiche che offre

- matplotlib, per la gestione dei grafici

Come è possibile vedere dal codice sottostante è stato costruito un vettore tempo contenente tanti punti quanti sono quelli della frequenza di campionamento e compresi in un intervallo temporale scelto in modo arbitrario, in questo caso un secondo. La funzione denominata nel codie calcola prende come unico parametro l'istante di tempo in cui si studia il segnale, calcola il valore della successione dei termini di Fourier in accordo al numero di armoniche specificato e restituisce il risultato.

La porzione di codice è la seguente:

```
sampleR = 1000       # Frequenza di campionamento
durata0 = 1          # Tempo di osservazione del segnale
f0 = 5       # Frequenza
t = np.linspace(0,durata0,sampleR)       # Vettore tempo
xt = []                  # Array contenente i termini di Fourier
numeroCoeff = 10                  # Numero di coefficienti da utilizzare nella
sommatoria

# Funzione che calcola i termini di Fourier
def calcola(t0):

    somma = 0

    for k in range(1,numeroCoeff):
        # Vogliamo solo i termini dispari
        if k % 2 == 1:
            somma = somma + (1/k*pow(-1, (k-1)/2)*np.cos(2*np.pi*k*f0*t0))

    return 1/2+2/np.pi*somma

for i in t:
    xt.append(calcola(i))
```

Il segnale ricostruito per mezzo delle sue armoniche può essere visualizzato costruendo un grafico e di seguito si riporta la porzione di codice che lo realizza:

```
plt.figure()
plt.plot(t,xt)
plt.title("Segnale onda quadra")
plt.ylabel("x(t)")
plt.xlabel("Tempo (s)")
plt.show()
```

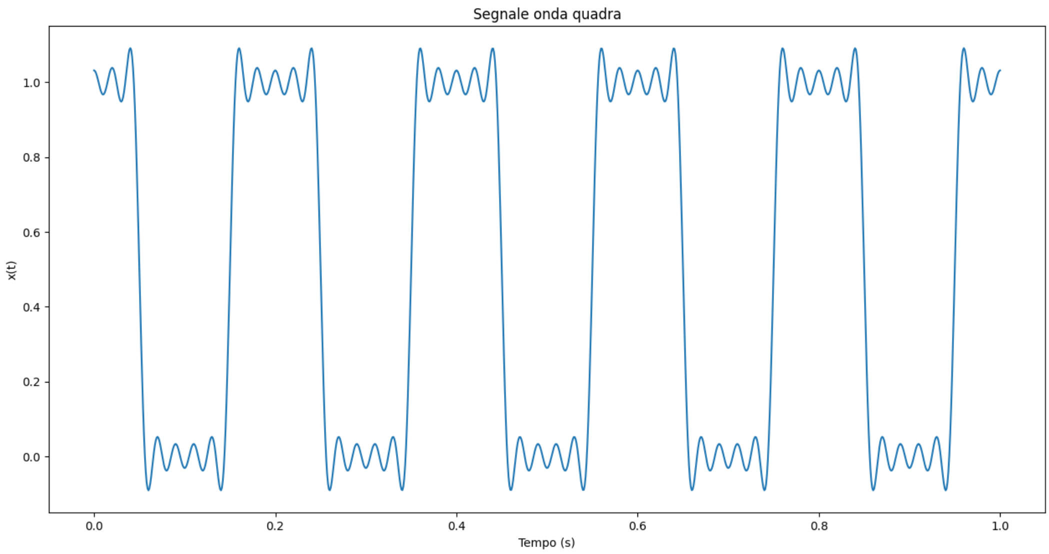


Figura 1-1 Segnale onda sinusoidale con un numero di coefficienti pari a 10

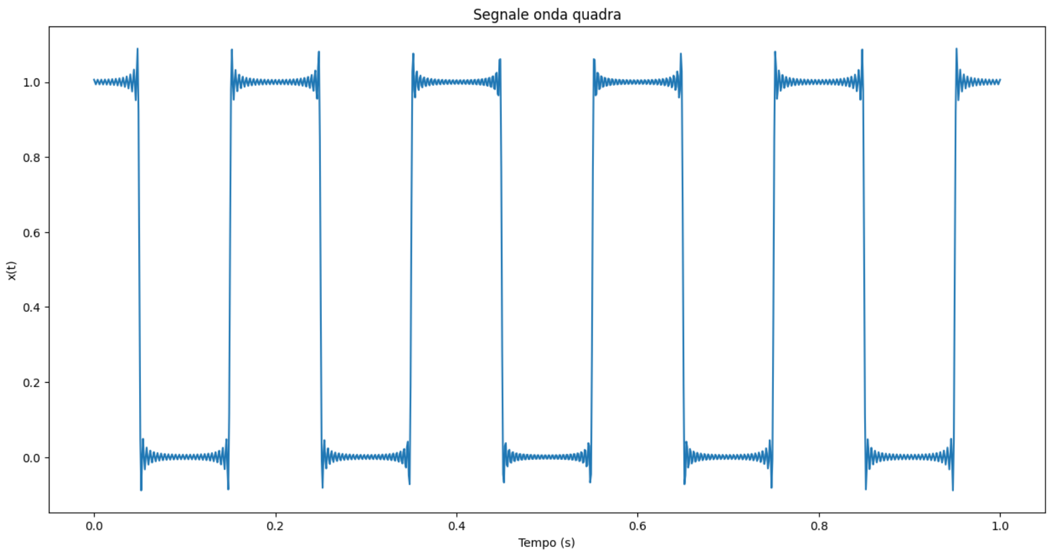
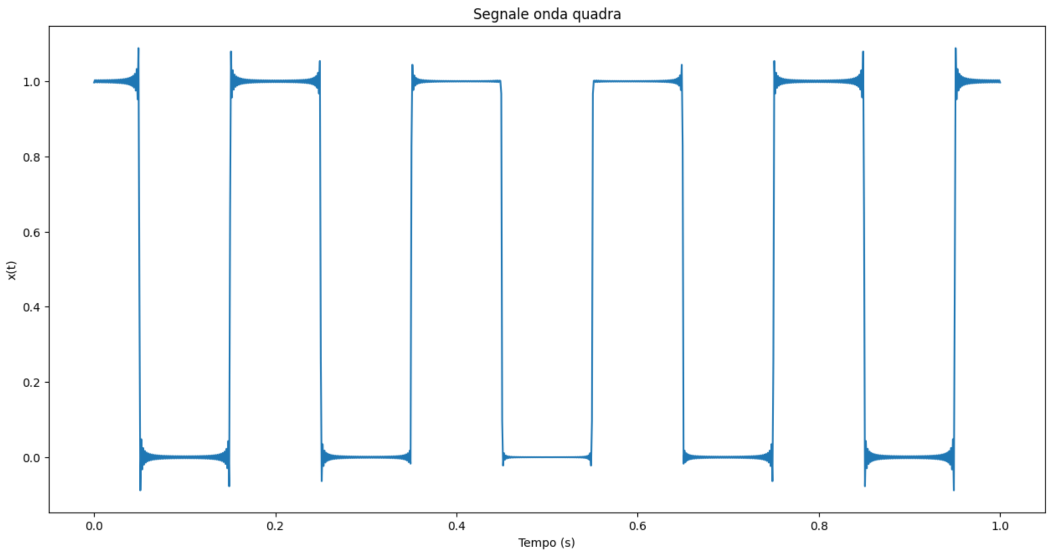


Figura 1-2 Segnale onda sinusoidale con un numero di coefficienti pari a 50


Segnale onda quadra
1.0
0.8
0.6
0.4
0.2
0.0
x(t)
0.0
0.2
0.4
0.6
0.8
1.0
Tempo (s)

Capitolo 2. Equazione di analisi

2.1 Analisi di segnali periodici nel dominio temporale

In questa sezione andremo ad analizzare i segnali periodici onda sinusoidale, onda triangolare e onda quadra alle frequenze 100 Hz, 200 Hz e 440 Hz prima nel dominio temporale e poi successivamente nel dominio delle frequenze.

Le librerie utilizzate per generare le onde sono:

- numpy
- scipy

Di seguito si riporta la porzione di codice che oltre alla generazione delle onde comprende anche l'impostazione della frequenza di campionamento, la durata di osservazione del segnale, la creazione del vettore tempo utilizzando la stessa funzione vista nel Capitolo 1 e infine una lista contenente le frequenze.

```
sampleRate1 = 1000  # Frequenza di campionamento
durata1 = 1  # Tempo di osservazione del segnale

t1 = np.linspace(0,durata1,sampleRate1,endpoint=False) # Vettore tempo
f = [100,200,440]  # Insieme delle frequenze

# Per ogni frequenza in f si generano i segnali sinusoide, onda triangolare e onda
quadra
# Inserisco ogni segnale all'interno della corrispettiva lista

ondasinusoidale = []
for freq in f:
    ondasinusoidale.append(np.sin(2*np.pi*freq*t1))

ondaquadra = []
for freq in f:
    ondaquadra.append(signal.square(2*np.pi*freq*t1))

ondatraingolare = []
for freq in f:
    ondatraingolare.append(signal.sawtooth(2*np.pi*freq*t1,0.5))  # il parametro
0.5 è stato inserito per ottenere una simmetria nella forma dell'onda
```

Dopo aver generato le onde alle varie frequenze non rimane altro che creare i grafici corrispondenti.

```
# Plot onda sinusoidale

plt.figure()
plt.xlabel("Tempo (s)")
plt.ylabel("x(t)")
plt.plot(t1, ondasinusoidale[0])
plt.title("Segnale onda sinusoidale")

# Plot onda triangolare

plt.figure()
plt.xlabel("Tempo (s)")
plt.ylabel("x(t)")
plt.plot(t1, ondatraingolare[0])
plt.title("Segnale onda triangolare")

# Plot onda quadra

plt.figure()
plt.xlabel("Tempo (s)")
plt.ylabel("x(t)")
plt.plot(t1, ondaquadra[0])
plt.title("Segnale onda quadra")

plt.show()
```

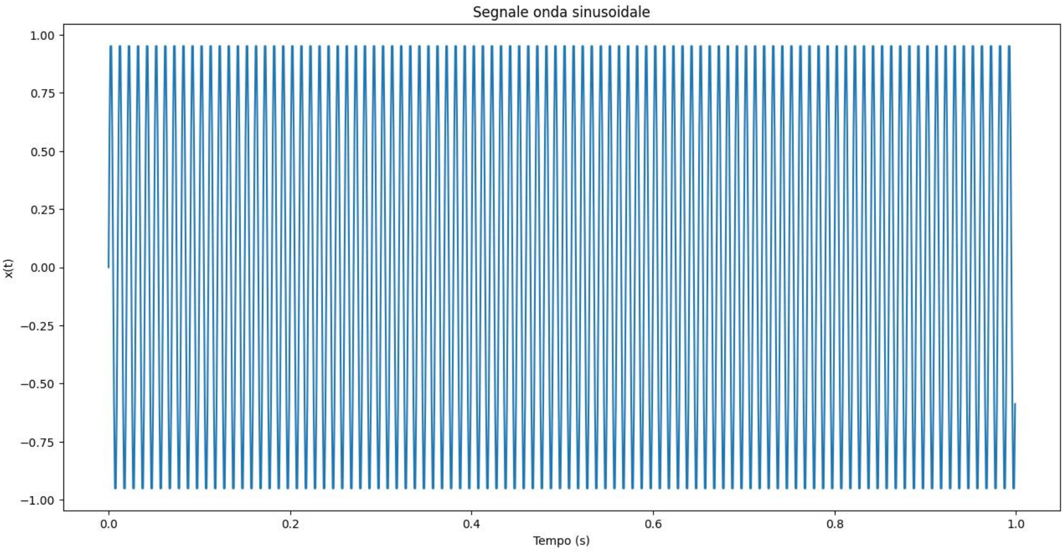


Figura 2-1 Segnale onda sinusoidale alla frequenza di 100 Hz

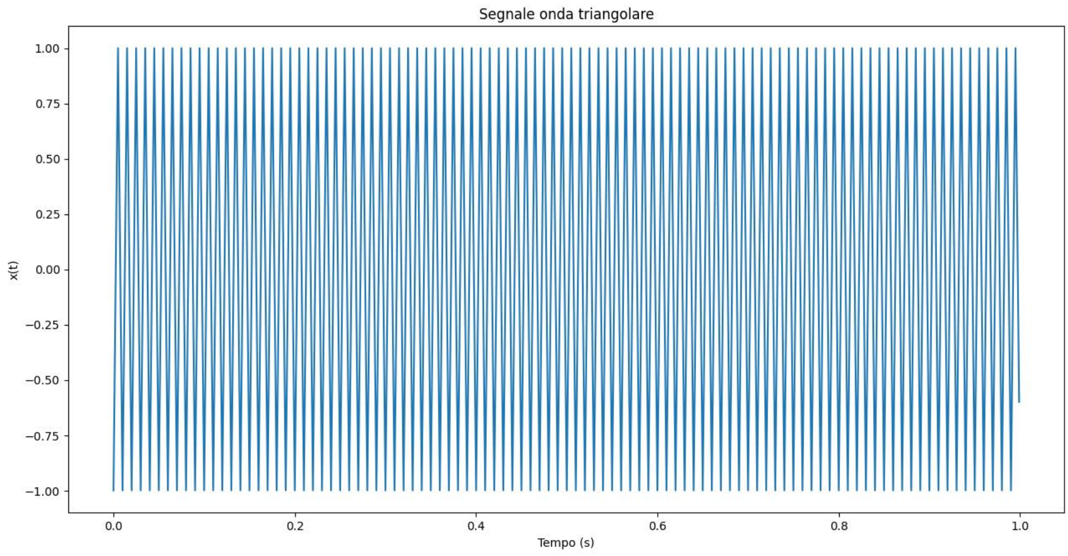


Figura 2-2 Segnale onda triangolare alla frequenza di 100 Hz

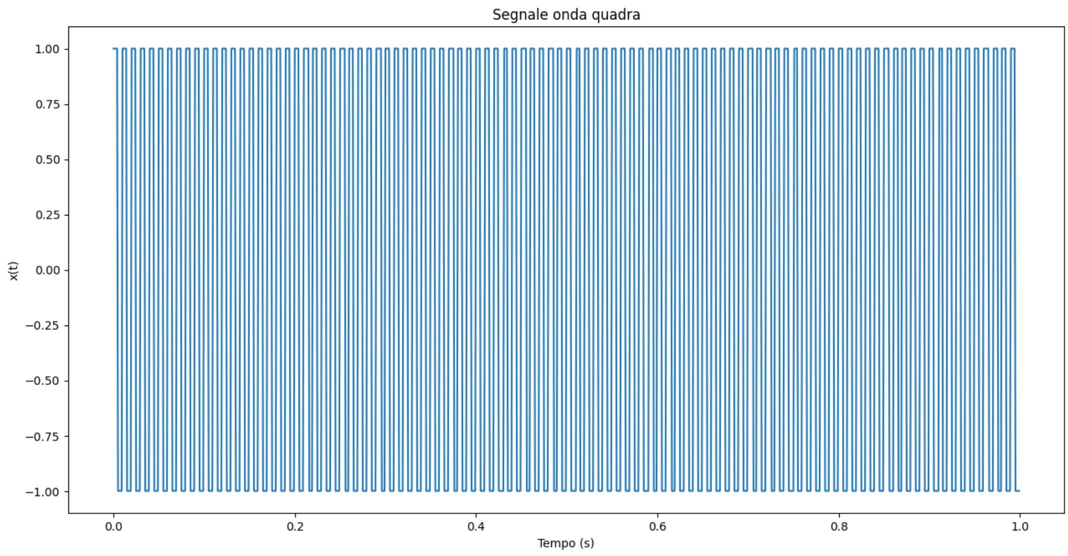


Figura 2-3 Segnale onda quadra alla frequenza di 100 Hz

2.2 Analisi di segnali periodici nel dominio delle frequenze

In questa sezione si affronta lo studio dei segnali nel dominio delle frequenze e si utilizzerà l'equazione di analisi grazie alla quale è possibile ottenere i coefficienti di Fourier e quindi visualizzare lo spettro in frequenza del segnale. L'equazione di analisi si presenta in questa forma:

$$X_k = \frac{1}{T_0}\int_{-T_0/_2}^{T_0/_2} x(t)e^{-j2\pi k f_0 t}\, dt$$

quindi in questo caso l'obiettivo è quello di determinare, a partire da un certo segnale *x(t)*, un insieme che rappresenti la successione dei coefficienti di Fourier e che ci permette di analizzare lo spettro del segnale e in particolare la parte reale e la parate immaginaria dei coefficienti.

Per quanto riguarda il lato computazionale di questa operazione possiamo fare riferimento ad un algoritmo chiamato Fast Fourier Transform, il quale presenta vantaggi importanti in termini di velocità e complessità rispetto ad un approccio manuale nel calcolo dei coefficienti di Fourier. Tale algoritmo viene messo a disposizione dalla libreria scipy e permette di calcolare la trasformata di Fourier discreta. Di seguito si presenta la porzione di codice che di fatto realizza la trasformata di Fourier per tutti i segnali considerati.

```
# Effettuo la trasformata di Fourier di ogni onda e per ogni frequenza
# Calcolo le trasformate e le inserisco nella lista corrispondente

fftsinusoide = []
for onda in ondasinusoidale:
    fftsinusoide.append(fft(onda, norm='forward'))   # norm='forward' serve per la
normalizzazione

fftquadra = []
for onda in ondaquadra:
    fftquadra.append(fft(onda, norm='forward'))

ffttraingolare = []
for onda in ondatraingolare:
    ffttraingolare.append(fft(onda, norm='forward'))
```

2.3 Grafici di spettro di potenza, parte reale e parte immaginaria

A causa del passaggio dal dominio temporale a quello delle frequenze è necessario eliminare il vettore tempo costruito negli esempi precedenti e sostituirlo con un nuovo vettore composto da frequenze. La creazione di questo vettore avviene grazie alla funzione fftfreq della libreria scipy.

```
# Creazione vettore delle frequenze
freq = fftfreq(len(ondasinusoidale[0]), 1/sampleRate1)
```

Di seguito si presenta la porzione di codice il cui obiettivo è quello di fornire un'analisi spettrale del segnale, in particolare permette di visualizzare in un grafico lo spettro di potenza e inoltre le due componenti che formano il segnale reale ossia la parte immaginaria e la parte reale.

```
# Plot spettro di potenza dell'onda sinusoidale

plt.figure()
plt.plot(freq, abs(fftsinusoide[0])**2)
plt.ylabel("$|X|_{k}$")
plt.xlabel("Frequenza (Hz)")
```

```
plt.title("Onda sinusoidale: potenza")

# Plot parte immaginaria dell'onda sinusoidale

plt.figure()
plt.plot(freq, fftsinusoide[0].imag)
plt.ylabel("$|X|_{k}$")
plt.xlabel("Frequenza (Hz)")
plt.title("Onda sinusoidale: parte immaginaria")

# Plot parte reale dell'onda sinusoidale

plt.figure()
plt.plot(freq, fftsinusoide[0].real)
plt.ylabel("$|X|_{k}$")
plt.xlabel("Frequenza (Hz)")
plt.title("Onda sinusoidale: parte reale")

plt.show()
```

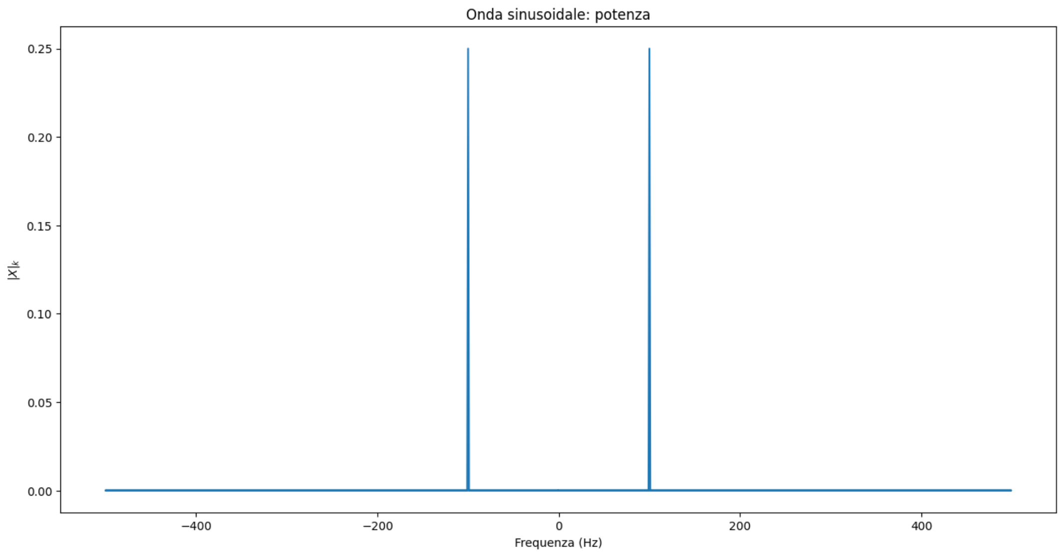


Figura 2-4 Segnale onda sinusoidale a 100 Hz nel dominio delle frequenze

Come si può notare dalla Figura 2-4 si hanno due picchi in corrispondenza delle frequenze 100 Hz e -100 Hz, il motivo di ciò risiede nel fatto che lo spettro di ampiezza per segnali reali è sempre simmetrico rispetto all’origine quindi rispetto alla frequenza nulla.

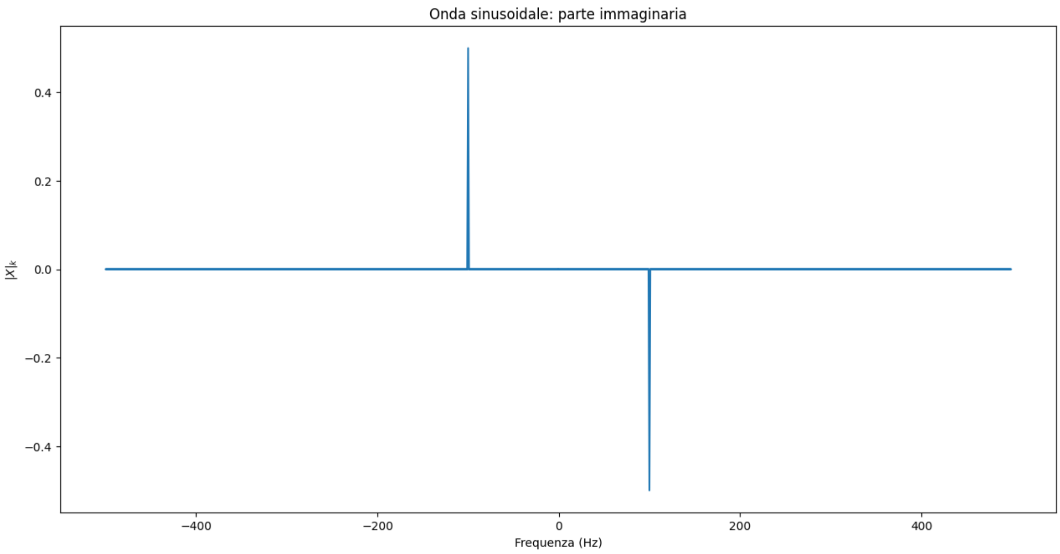


Figura 2-5 Parte immaginaria dei coefficienti di Fourier per segnale onda sinusoidale a 100 Hz

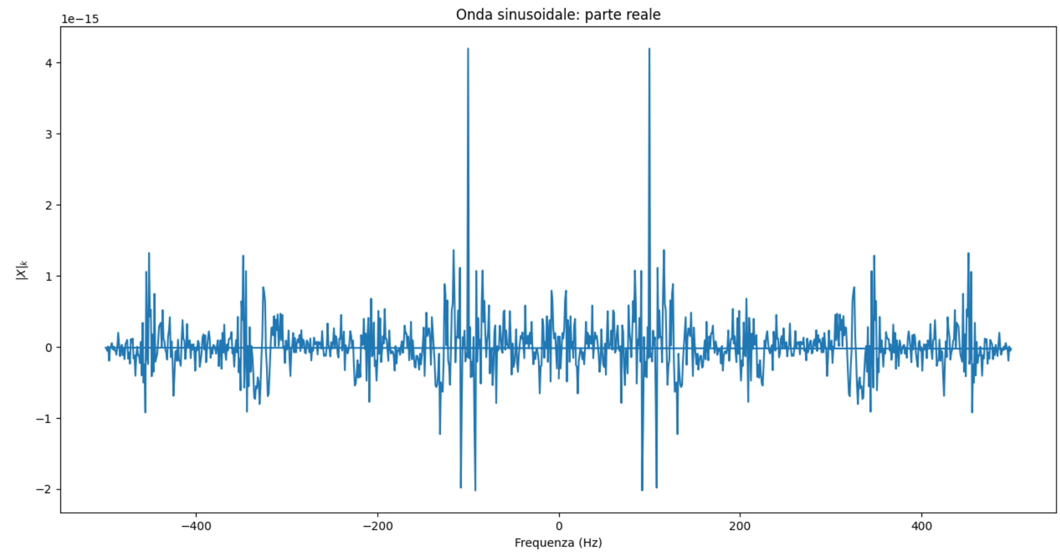


Figura 2-6 Parte reale dei coefficienti di Fourier per segnale onda sinusoidale a 100 Hz

2.4 Studio in frequenze del segnale somma

In questa sezione il lavoro da svolgere è essenziale uguale a quello della sezione precedente, l'unico aspetto che cambia è il segnale che andremo ad analizzare in quanto questo si ottiene dalla somma delle onde sinusoidali alle frequenze 100 Hz, 200 Hz e 440 Hz. Una volta costruito il segnale si effettua la trasformata di Fourier utilizzando la funzione fft della libreria scipy.

```
segnaleSomma = ondasinusoidale[0] + ondasinusoidale[1] + ondasinusoidale[2]

fftSommaSinusoide = fft(segnaleSomma)
```

Di seguito si riporta la porzione di codice che si occupa di generare i grafici seguendo lo stesso metodo visto nella sezione precedente.

```
plt.figure()
plt.plot(freq, abs(fftSommaSinusoide)**2)
plt.ylabel("$|X|_{k}$")
plt.xlabel("Frequenza (Hz)")
plt.title("Ssegnale somma: potenza")

plt.figure()
plt.plot(freq, fftSommaSinusoide.imag)
plt.ylabel("$|X|_{k}$")
plt.xlabel("Frequenza (Hz)")
plt.title("Segnale somma: parte immaginaria")

plt.figure()
plt.plot(freq, fftSommaSinusoide.real)
plt.ylabel("$|X|_{k}$")
plt.xlabel("Frequenza (Hz)")
plt.title("Segnale somma: parte reale")
plt.show()
```

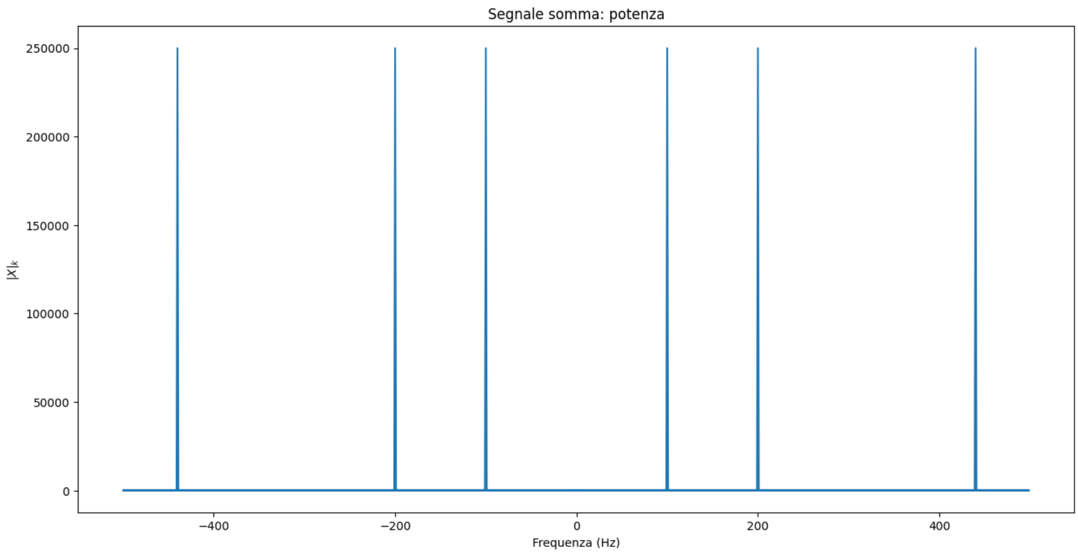


Figura 2-7 Segnale somma onde sinusoidali nel dominio delle frequenze

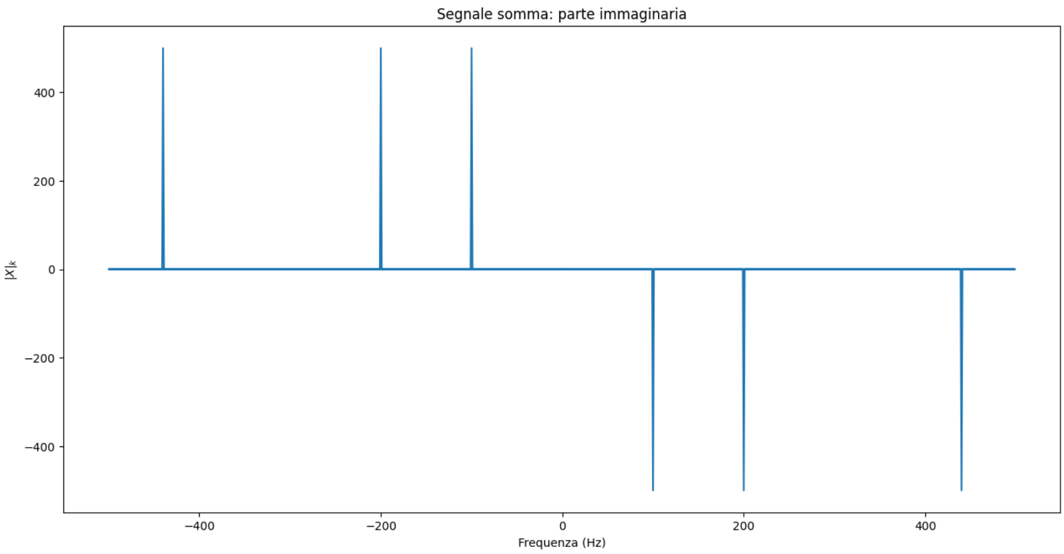


Figura 2-8 Parte immaginaria dei coefficienti di Fourier per segnale somma onde sinusoidali

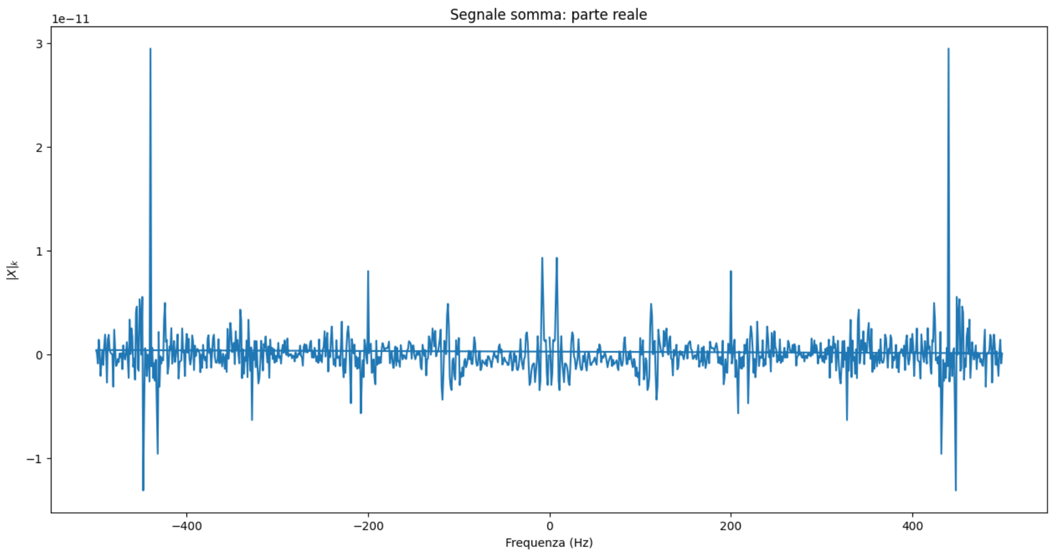


Figura 2-9 Parte reale dei coefficienti di Fourier per segnale somma onde sinusoidali

Conclusioni

In questa relazione sono stati introdotti alcuni concetti di base della teoria dei segnali, necessari a comprendere il lavoro che è stato svolto successivamente. Nella prima parte l’attenzione è stat posta sull’equazione di sintesi di un segnale onda quadra, introducendo prima lo sviluppo in serie di Fourier e poi si è dimostrato come la ricostruzione di tale segnale vari a seconda del numero di coefficienti di Fourier e quindi di armoniche utilizzate. La seconda parte della relazione ha mostrato un aspetto più pratico e in particolare abbiamo visto come può essere utilizzata l’equazione di analisi per realizzare uno studio spettrale di tre tipi di segnali e come possono essere calcolati i coefficienti di Fourier attraverso un apposito algoritmo chiamato Fast Fourier Transform implementato nel linguaggio Python.