Laboratorio di Elaborazione di Immagini

Esercitazione 2:

TRASFORMATA DI FOURIER

Silvia Obertino

22 marzo 2017

Fourier

- Jean Baptiste Joseph Fourier (1768 - 1830)
- Matematico francese
- Ha praticamente fondato l'elaborazione dei segnali (senza saperlo)

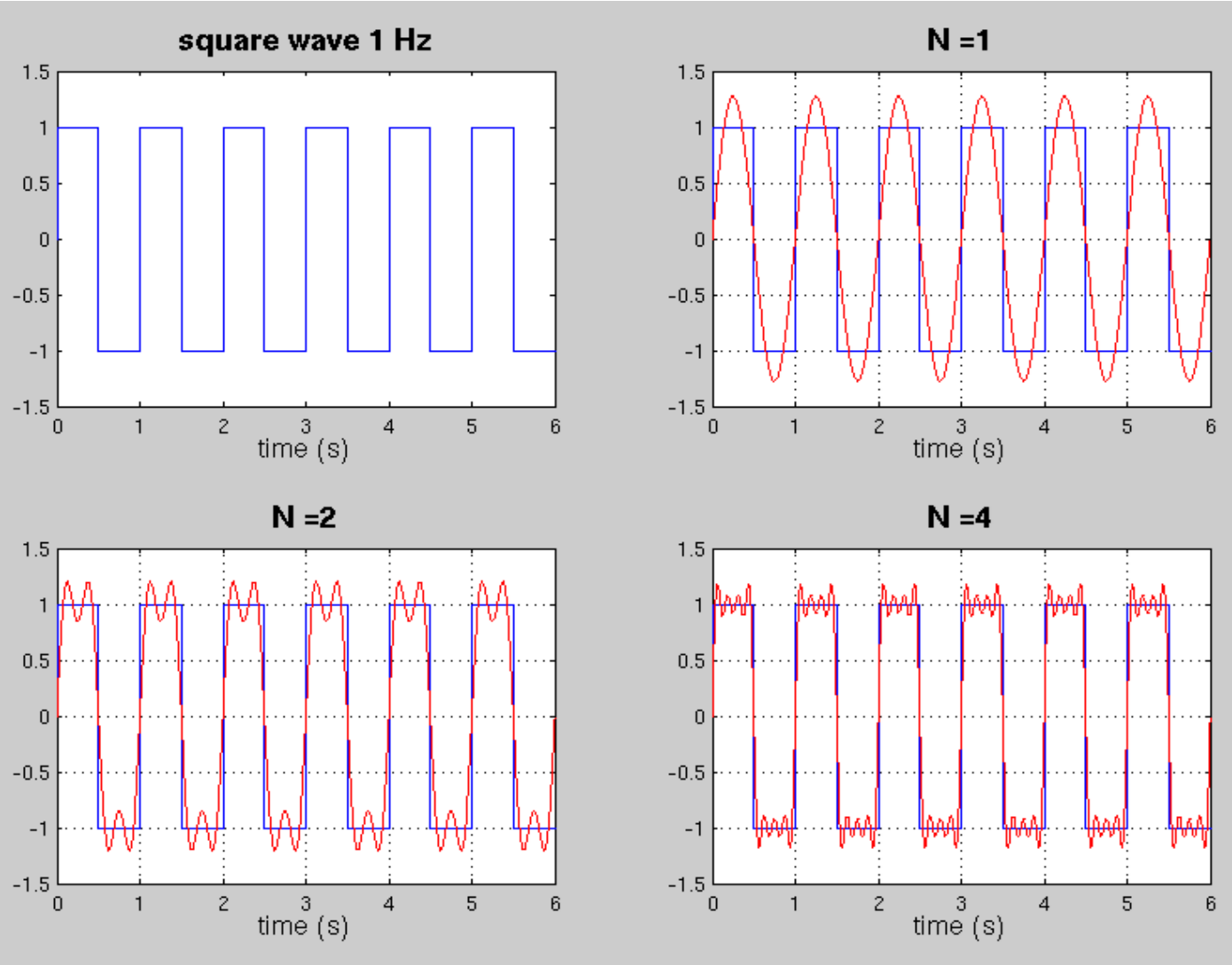
Serie di Fourier

- **Fourier** ebbe la brillante idea di rappresentare funzioni continue e periodiche come una sommatoria di funzioni trigonometriche (seni e coseni)

$$f(x) = \frac{a_0}{2} \sum_n \left[a_n cos(nx) + b_n sin(nx) \right]$$

- a_n e b_n sono i **coefficienti** della serie di Fourier

Serie di Fourier: esempio



Serie di Fourier

- Grazie alla Formula di Eulero:

$$e^{ix} = cos(x) + i\ sin(x)$$

- Possiamo scrivere la serie di Fourier in maniera più compatta

$$f(x) = \sum_{n} c_n e^{inx}$$

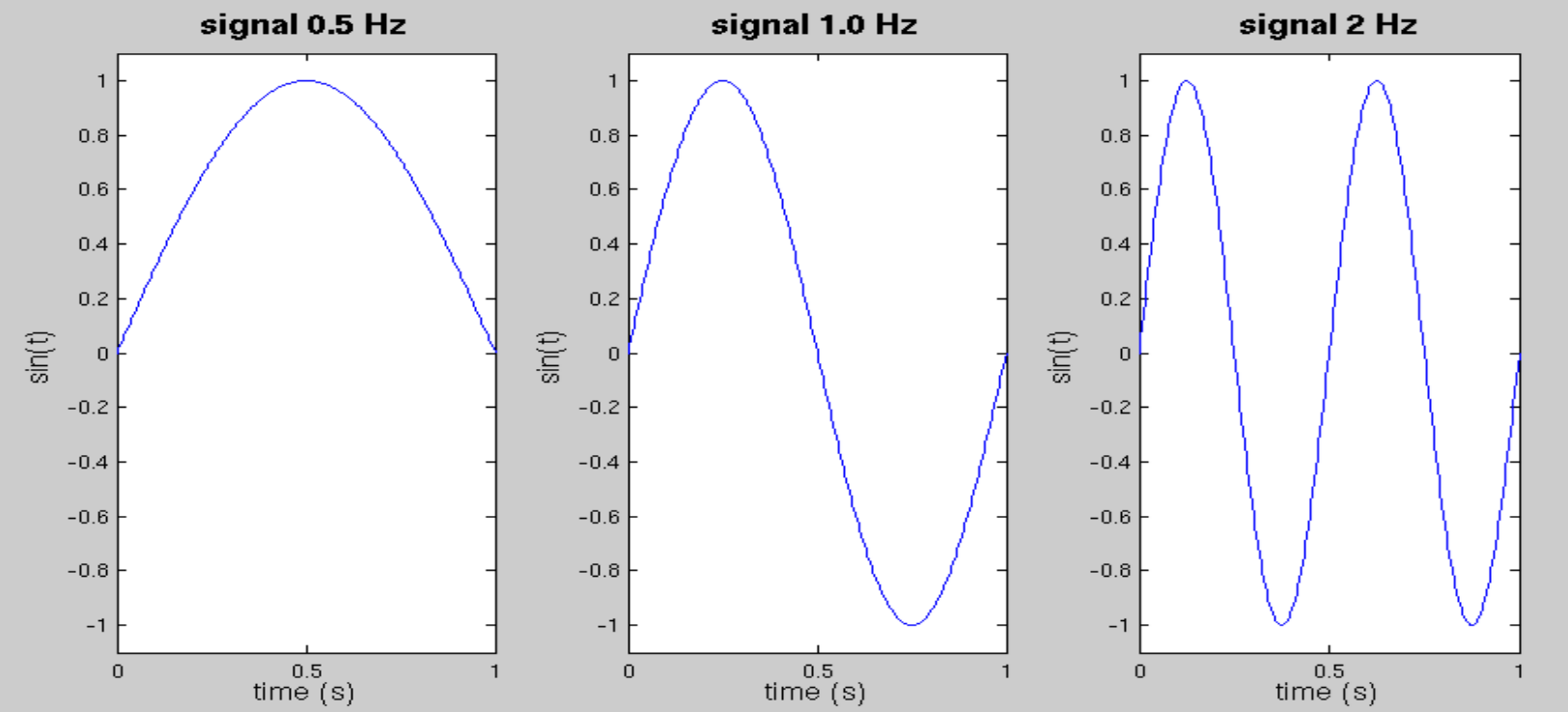
Serie di Fourier

- Per ottenere i coefficienti c_n basta calcolare il prodotto scalare tra la funzione della base di Fourier e la funzione considerata

$$c_n = \frac{1}{2\pi} \int f(x) e^{-inx} dx$$

Trasformata di Fourier

- Visto che “n” è argomento di funzioni trigonometriche possiamo considerarlo come se fosse una frequenza
- Le frequenze ovviamente sono continue (es. ω=0.5 Hz)



Trasformata di Fourier

- Studiando i coefficienti della serie di Fourier possiamo vedere quindi quali “frequenze” sono predominanti nel nostro segnale

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{i\omega x} dx$$

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega) e^{-i\omega x} d\omega$$

Trasformata di Fourier DISCRETA

$$G[k] = \sum_{n=0}^{N-1} g[n] \exp(-j2\pi nk / N) \quad k = 0,1, \ldots, \text{N-1}$$

- Se definiamo una funzione di pesi

$$W_N = e^{\frac{-j2\pi}{N}}$$

- Possiamo riscrivere:

$$G[k] = \sum_{n=0}^{N-1} g[n] W_N^{kn}$$

Trasformata di Fourier DISCRETA

- La trasformata discreta è importante nell'analisi delle frequenze (spettro), perché prende un segnale discreto nel dominio del tempo e trasforma il segnale nella sua rappresentazione nel dominio discreto delle frequenze.

- Senza la discretizzazione della trasformata e del segnale non riusciremmo a computare la trasformata di Fourier con un microprocessore o con un sistema basato su Digital Signal Processing

Trasformata di Fourier in MATLAB

- In MATLAB è implementata la versione “veloce” della trasformata di Fourier discreta (per funzioni campionate)
  - Fast Fourier Transform (FFT)

FFT: esempio

```
>> x = [1 2 1];
>> xfft = fft(x)

>> xfft=
        4 + 0.0i
        -0.5 – 0.8660i
        -0.5 + 0.8660i
```

FFT: funzioni trigonometriche

- Se il nostro segnale è composto esclusivamente da funzioni trigonometriche, la nostra trasformata di Fourier sarà diversa da zero solo nelle frequenze corrispondenti quelle delle funzioni trigonometriche del segnale

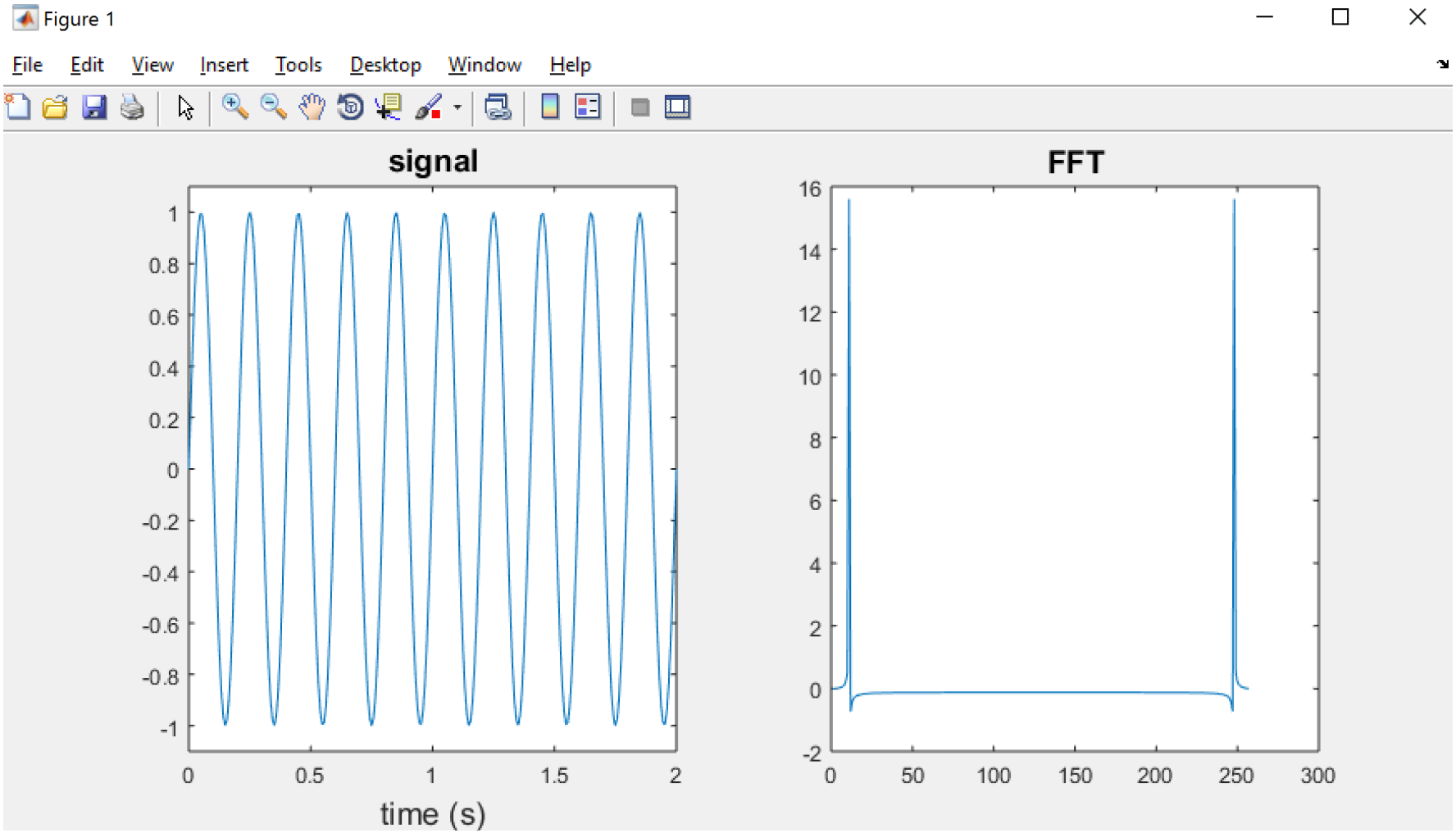
FFT: funzioni trigonometriche

```
>> Fs = 128;                % Sampling frequency
>> T = 1/Fs;                % Sampling period
>> t = 0:T:2;       % Time vector

>> s = sin(2*pi*5*t);
>> S = real(fft(s))

>> figure
>> subplot(1,2,1) plot(t,s);
>> xlabel('time (s)','FontSize',14)
>> title('signal','FontSize',14,'fontweight','bold')
>> ylim([-1.1,1.1])
>> subplot(1,2,2) plot(S);
>> title('FFT','FontSize',14,'fontweight','bold')
```

FFT: funzioni trigonometriche



FFT: funzioni trigonometriche

- Per motivi di efficienza la FFT è ordinata in maniera un po' particolare:
  - Omega = [0:n/2-1 0 -n/2+1:-1]
  - n/2 = massima frequenza raggiungibile per il **Teorema di Nyquist** (frequenza campionamento /2)

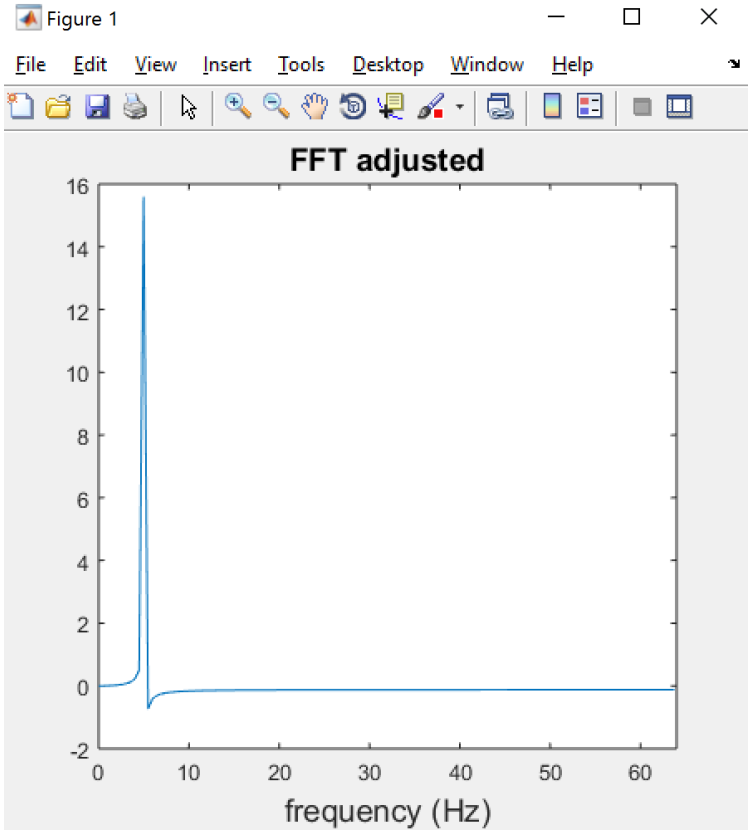
- Vengono incluse anche le **frequenze negative**, quindi lo spettro viene 'duplicato'

FFT: funzioni trigonometriche

- Il grafico corretto si ottiene considerando solo la prima metà dello spettro (frequenze positive) e sistemando opportunamente l'asse delle x

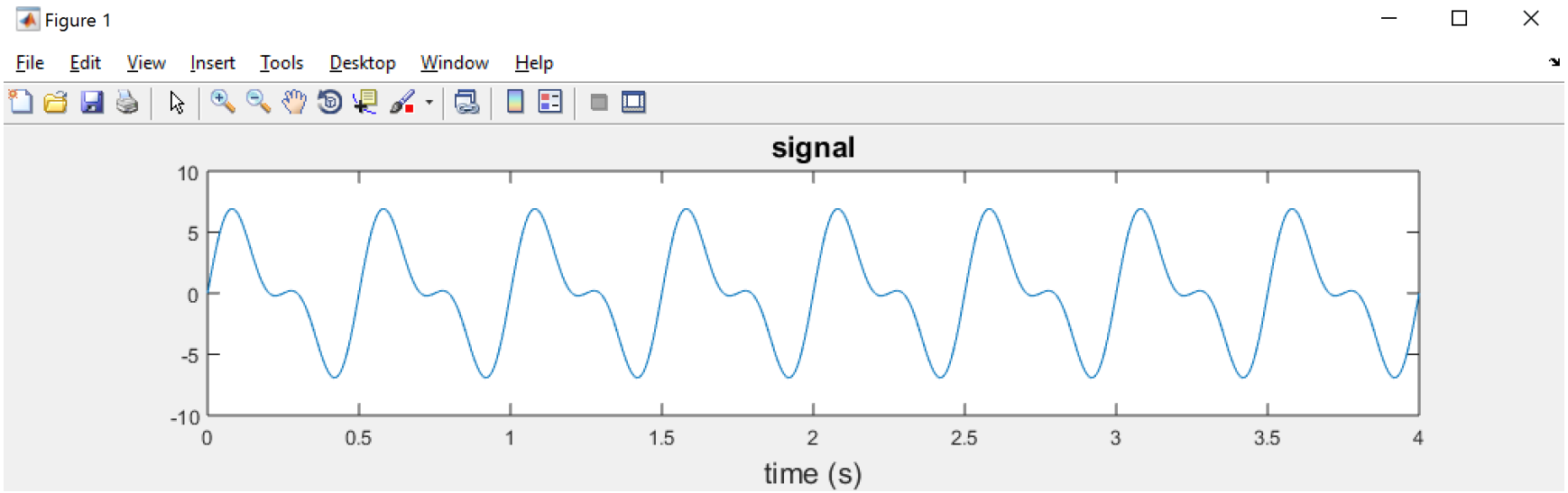
```
>> l = length(t);
>> N = floor(l/2)+1;
>> S = S(1:N);
>> f_max = Fs/2;
>> f_sampling = Fs/l;
>> f = 0:f_sampling:f_max;


>> plot(f,S);
>> title('FFT adjust','FontSize',14,'fontweight','bold')
>> xlabel('frequency (Hz)','FontSize',14)
>> xlim([0,f_max])
```



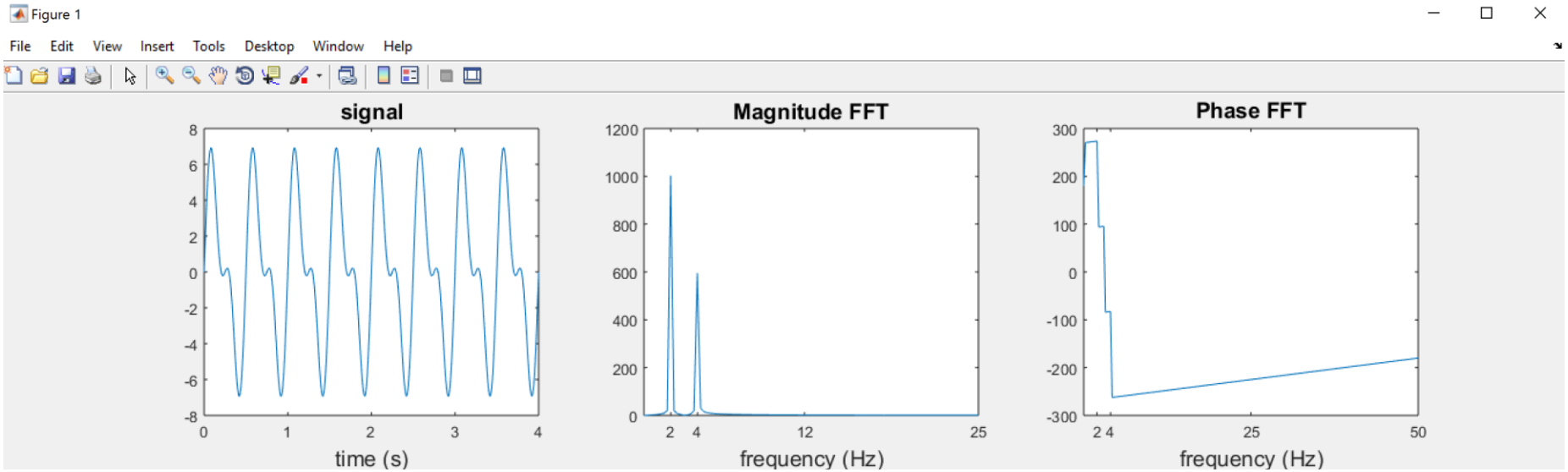
Esercizio: funzioni trigonometriche

- Supponiamo di campionare un segnale ogni 0.01 secondi per la durata di 4 secondi
- Il segnale è dato dalla somma di due sinusoidi di ampiezza 3 e 5 e frequenza 4 e 2 ($\omega=2\pi f$) rispettivamente
- Generare il grafico tempo – ampiezza



Esercizio: rappresentazioni di spettro e fase

- Rappresentare in tre subplot
  - Segnale precedente
  - Lo spettro della trasformata
  - La fase della trasformata

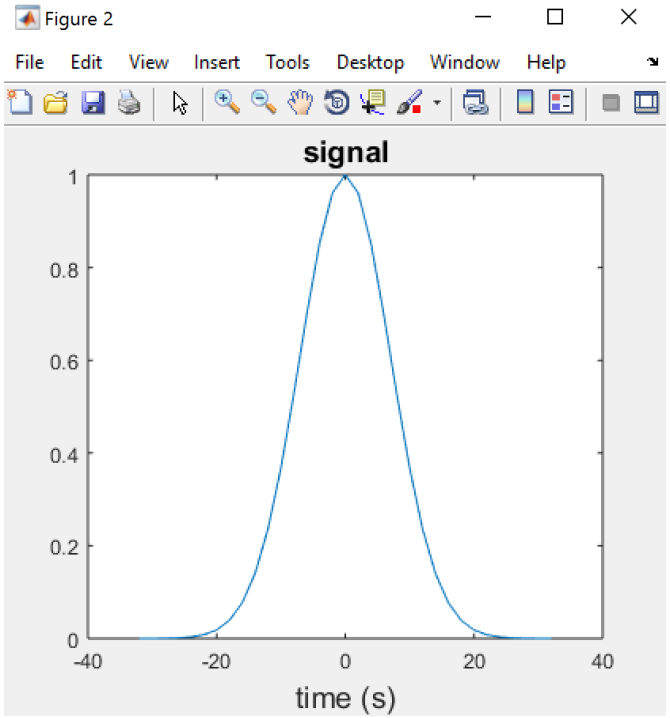


Funzione gaussiana

- Funzione Gaussiana definita come

$$f(x) = e^{-\alpha x^2}$$

```
>> Fs = 0.5;
>> T = 1/Fs;
>> t = -32:T:32;
>> l = length(t);
>> alpha = 0.01;
>> g = exp(-alpha*t.^2 );
>> figure; plot(t,g)
>> xlabel('time (s)','FontSize',14)
>> title('signal','FontSize',14,'fontweight','bold')
```



FFTSHIFT

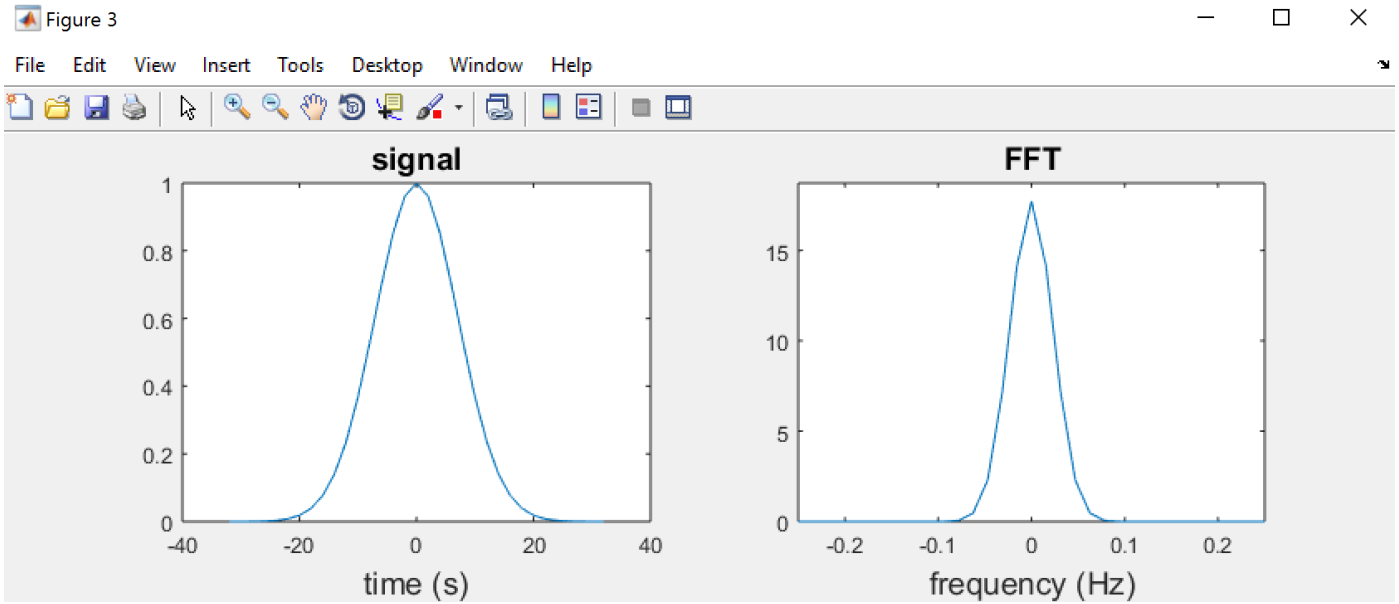
- Per funzioni simmetriche bisogna usare un riordinamento delle frequenze

```
>> help fftshift
```

- Figura con segnale originale trasformata e trasformata riordinata

FFTSHIFT su gaussiana

```
>> G = fftshift(fft(g));
```



Zero Padding

- Sapete che se volete aumentare il numero di campioni della trasformata, dovete usare lo Zero Padding

```
>> help padarray
```

Zero Padding

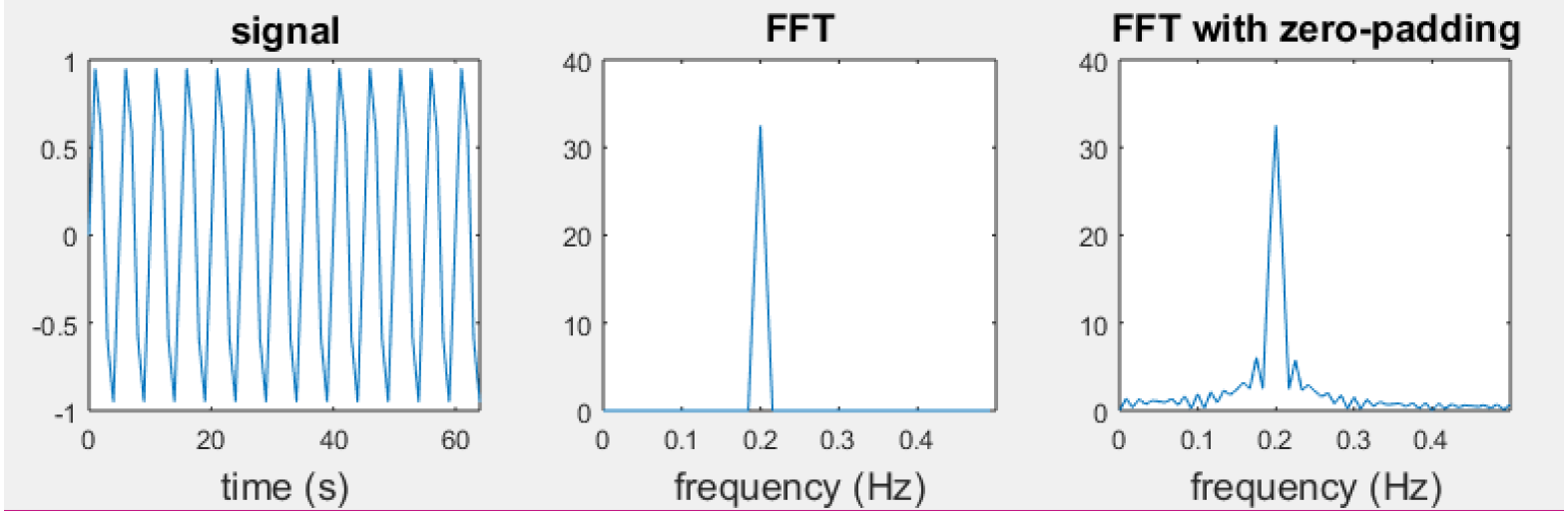
- In MATLAB la funzione fft prevede già lo zero-padding automatico nel caso venga esplicitato il numero di campioni in output

```
>> fft(X,n)
```

Dove n è il numero di campioni che vogliamo ottenere

Esercizio: Zero Padding

- Supponiamo di campionare un segnale ogni secondo per la durata di 64 secondi
- Il segnale è dato sinusoide di ampiezza unitaria e frequenza 0.2 ($\omega=2\pi f$)
- Plottare segnale e spettro della trasformata di Fourier e spettro della trasformata utilizzando lo Zero-Padding



Esercizio: Zero Padding

- Controllare il segnale da cui deriva la trasformata con zero-padding plottando il segnale completo