

```

%% primo esercizio
% Costruire un segnale y ottenuto come somma di
% a. La somma di un segnale seno (con ampiezza 2) e un segnale coseno (con
% ampiezza 1) aventi lo stesso periodo di 1 s, usando un passo di campionamento
% pari a 1 campione ogni 10 ms (frequenza di campionamento: 100 Hz)
% b. Un segnale seno con frequenza dimezzata rispetto ai primi due (con ampiezza
% 4)
% Si rappresenti il segnale per un tempo pari a 5 periodi del segnale al punto b.

```

```
clear;
```

```

Fcamp = 100;
Tcamp = 0.01;
%costruisco asse temporale
Tfine = 5*2; %5 periodi di B, ma un periodo di B è doppio del periodo di A che è 1 secondo
t = 0:Tcamp:Tfine;

```

```

%costruisco i 3 segnali
y1 = 2*sin(2*pi*1*t); %se T=1s F=1Hz
y2 = 1*cos(2*pi*1*t);
y3 = 4*sin(2*pi*0.5*t); %frequenza dimezzata = 0.5

```

```
%costruisco segnale e lo rappresento
```

```

y = y1 + y2 + y3;
figure;
subplot(3,1,1)
plot(t, y);
title('segnale y');
xlabel('tempo');
ylabel('ampiezza');
xlim([0 Tfine]);

```

```

% 2) Si rappresenti la densità spettrale di potenza del segnale y calcolato al punto
% precedente in modo che lo spettro sia centrato (a) sulla frequenza massima
% rappresentabile e (b) sullo zero.
% Inoltre, si scalino opportunamente l'asse orizzontale (in modo che corrisponda all'asse
% delle frequenze) e l'asse verticale (in modo che corrisponda alla densità spettrale di
% potenza).

```

```

%per la densità spettrale di potenza uso periodogram
%essa restituisce già l'asse delle frequenze che uso nel plot
[densitaSpettralePotenza, f] = periodogram(y);
subplot(3,1,2);
plot(f, fftshift(densitaSpettralePotenza));
title('densità spettrale di potenza centrata sul massimo');
xlabel('frequenza');

```

```

subplot(3,1,3);
plot(f, densitaSpettralePotenza);
title('densità spettrale di potenza centrata sullo zero');
xlabel('frequenza');

```

```
%% esercizio 1 punto 3
```

```

% 3) Costruire un segnale composto da una sinusoide a 2 Hz e una a 2.5 Hz (si utilizzi la
% frequenza di campionamento come al punto 1). Se ne calcoli la trasformata di Fourier.
% Si utilizzi lo Zero Padding per modificare la dimensione dei frequency bins e poter
% meglio identificare visivamente le due sinusoidi nel dominio delle frequenze.

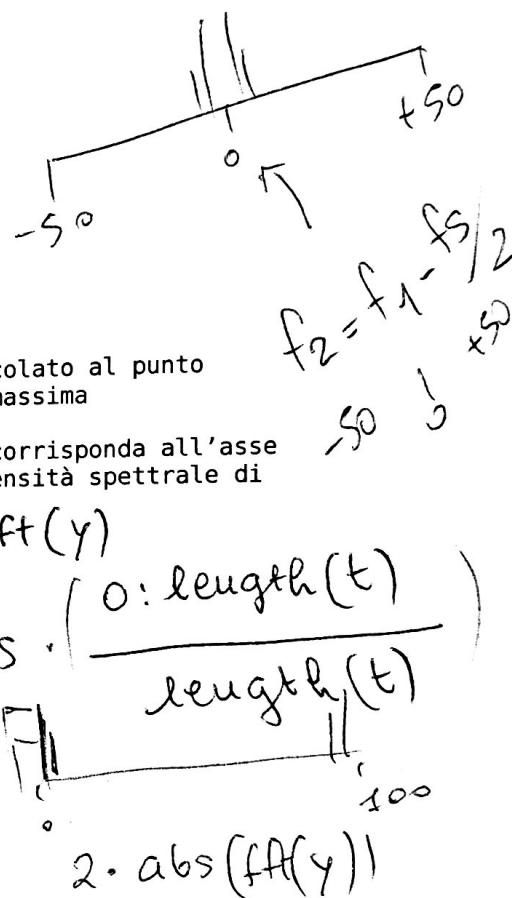
```

```
clear;
```

```

Fcamp = 100;
Tcamp = 0.01;
%costruisco asse temporale
Tfine = 5*2; %uso lo stesso asse di prima
t = 0:Tcamp:Tfine;
%costruisco segnale
s = sin(2*pi*2*t) + sin(2*pi*2.5*t);
%rappresento
figure;
subplot(3,1,1)
plot(t, s);
title('segnale s');

```



```

xlabel('tempo');
ylabel('ampiezza');
xlim([0 Tfine]);

% calcolo trasformata di fourier
S = fft(s);
%costruisco asse frequenze
l = length(t) - 1;
f = 0:Fcamp/l:Fcamp;
%grafico
subplot(3,1,2)
plot(f, S);
title('spettro fourier S');
xlabel('frequenza');
ylabel('ampiezza');
xlim([0 Fcamp/2]); %metto in evidenza solo la parte di sinistra

%aplico lo zero-padding e ricalcolo fourier

pad = 250;
Spad = fft(s, length(s)+pad);

subplot(3,1,3)
plot(abs(Spad));
title('spettro S con pad');
xlabel('frequenza');
ylabel('ampiezza');
xlim([0 Fcamp/2]);

%% esercizio 2


% Creare un'immagine di dimensioni 512x512 a 256 livelli di grigio e disegnarvi una croce
% nel centro corrispondente a livello di grigio 200 centrato su sfondo a livello di grigio
% 128 (per le misure si veda l'immagine sotto riportata).
% a. Calcolare la trasformata di Fourier e centrare lo spettro al centro
% dell'immagine. Visualizzare in una figura a subplot l'immagine di partenza, lo
% spettro di potenza [rappresentare solo la porzione (1:128,1:128)] e lo spettro
% di fase.

clear;

%costruisco immagine
cross(1:200, 1:200) = 200;
cross(1:90, 1:90) = 128;
cross(1:90, 110:200) = 128;
cross(110:200, 1:90) = 128;
cross(110:200, 110:200) = 128;

mat = padarray(cross, [156 156]);
mat(mat<199) = 128;

img = mat2gray(mat, [0 255]); %nil risultato visualizzato è in bianco e nero
MAT = fft2(img);
uint8(img)

figure;
subplot(2,2,1);
imshow(img, []);
title('Immagine');


% rappresento trasformata in ampiezza e fase
subplot(2,2,2);
imshow(log(abs(fftshift(MAT))), []); %scala logaritmica
title('Ampiezza spettro');
xlabel('scala logaritmica');

subplot(2,2,3);
imshow(log(abs(fftshift(MAT))), []);
title('Ampiezza spettro [1:128, 1:128]');
xlabel('scala logaritmica');
xlim([1 128]); %limito la porzione
ylim([1 128]);

```

```
subplot(2,2,4);
imshow(angle(fftshift(MAT)), []);
title('Fase spettro');
```

```
% b. Fare lo zero padding (raggiungere le dimensioni 612x612) tramite le apposite
% opzioni del comando per ottenere la trasformata di Fourier e ricalcolare la
% trasformata. Creare quindi una nuova figura a subplot in cui visualizzare la
% trasformata così ottenuta e l'immagine di partenza ricostruita a partire da:
% - La trasformata zero padded
% - Il solo contributo della fase
```

```
%faccio padding nella fft
MATpad = fft2(img, 612, 612);
```

```
figure;
subplot(3,1,1);
imshow(log(abs(fftshift(MATpad))), []);
title('ampiezza trasformata con pad');
xlabel('scala logaritmica');
```

```
%antitrasformo la zero padded
subplot(3,1,2);
imshow(iff2(MATpad), []);
title('immagine antitrasformata con pad');
```

```
%antitrasformo con solo contributo della fase
subplot(3,1,3);
imshow(iff2(angle(MATpad)), []);
title('immagine antitrasformata con pad, solo contributo fase');
```

$$\cos(\phi) + i \cdot \sin(\phi)$$

```
% 2) Aggiungere rumore gaussiano all'immagine di partenza (media = 0, varianza = 0.1).
% a. Filtrare l'immagine con tre diversi filtri gaussiani
% - Grandezza del filtro = 8; deviazione standard = 5
% - Grandezza del filtro = 8; deviazione standard = 1
% - Grandezza del filtro = 12; deviazione standard = 5
```

```
%creo immagine rumorosa
imgNoise = imnoise(img, 'gaussian', 0, 0.1);
%creo i 3 filtri
filtro1 = fspecial('gaussian', 8, 5);
filtro2 = fspecial('gaussian', 8, 1);
filtro3 = fspecial('gaussian', 12, 5);
%creo 3 immagini corrette
imgF1 = imfilter(imgNoise, filtro1);
imgF2 = imfilter(imgNoise, filtro2);
imgF3 = imfilter(imgNoise, filtro3);
%visualizzo le immagini
figure;
subplot(2,2,1);
imshow(imgNoise, []);
title('Immagine con rumore');
subplot(2,2,2);
imshow(imgF1, []);
title('Filtro 1');
subplot(2,2,3);
imshow(imgF2, []);
title('Filtro 2');
subplot(2,2,4);
imshow(imgF3, []);
title('Filtro 3');
```

```
% b. Decretare i valori migliori calcolando l'errore del filtraggio. Visualizzare in un
% plot l'immagine con rumore e i risultati dei 3 filtraggi.
```

```
%creo 3 immagini di errore
errCreato = abs(img - imgNoise);
err1 = abs(img - imgF1);
err2 = abs(img - imgF2);
err3 = abs(img - imgF3);
%visualizzo le immagini
```

```
figure;  
subplot(2,2,1);  
    imshow(errCreato, []);  
    title('Errore introdotto');  
subplot(2,2,2);  
    imshow(err1, []);  
    title('Errore Filtro 1');  
subplot(2,2,3);  
    imshow(err2, []);  
    title('Errore filtro 2');  
subplot(2,2,4);  
    imshow(err3, []);  
    title('Errore filtro 3');  
%tutti e 3 i filtri ricostruiscono bene l'immagine, ma il secondo ha  
%un'immagine di errore simile a rumore bianco, quindi ricostruisce meglio  
%l'immagine di partenza
```

valore errore? $\text{sum}(\text{sum}(\text{abs}(\text{imgN} - \text{imgF})))$