

Trasformata di fourier

```
DT=.1; % periodo di campionamento [s]
```

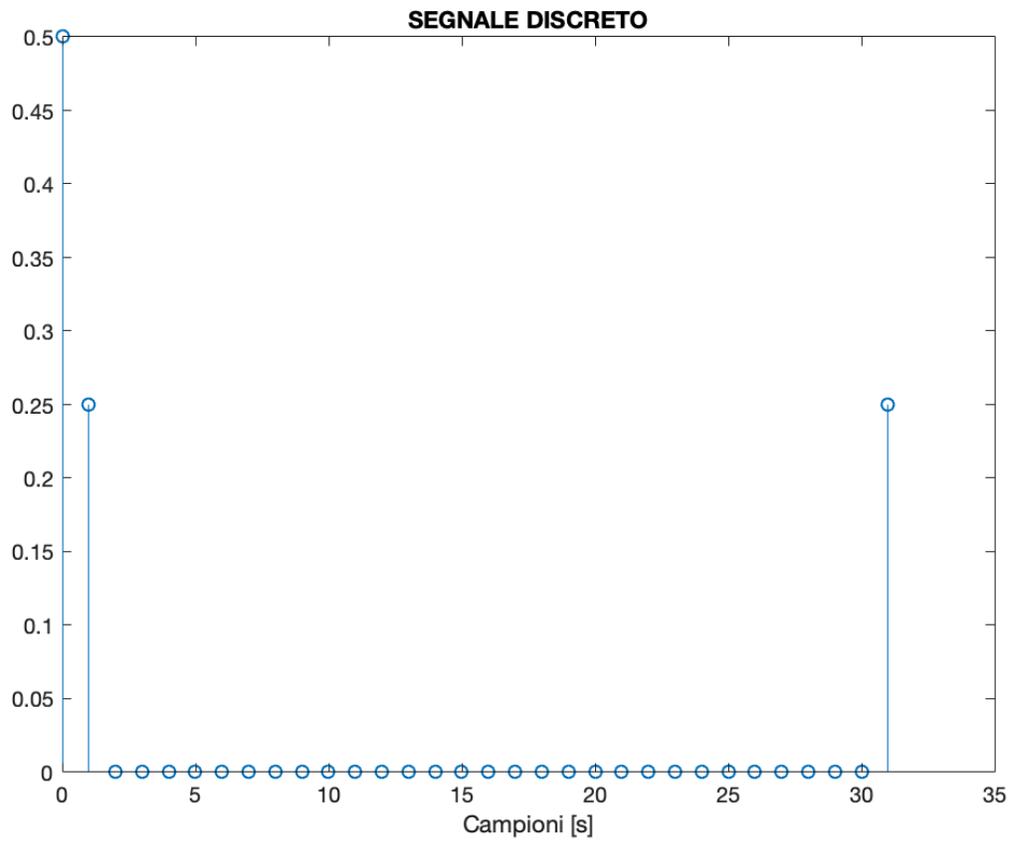
```
Nd=32; % numero di campioni  
nd=0:Nd-1; % variabile d'appoggio per ascisse plot  
  
Td=Nd*DT; % lunghezza segnale in secondi  
Fd=1/Td; % passo di campionamento in frequenza
```

```
Nc=1000; % numero di campioni  
nc=0:Nc-1; % variabile d'appoggio per ascisse plot  
  
Tc=Nc*DT; % lunghezza segnale in secondi  
Fc=1/Tc; % passo di campionamento in frequenza
```

Sequenza [1/4 1/2 1/4]

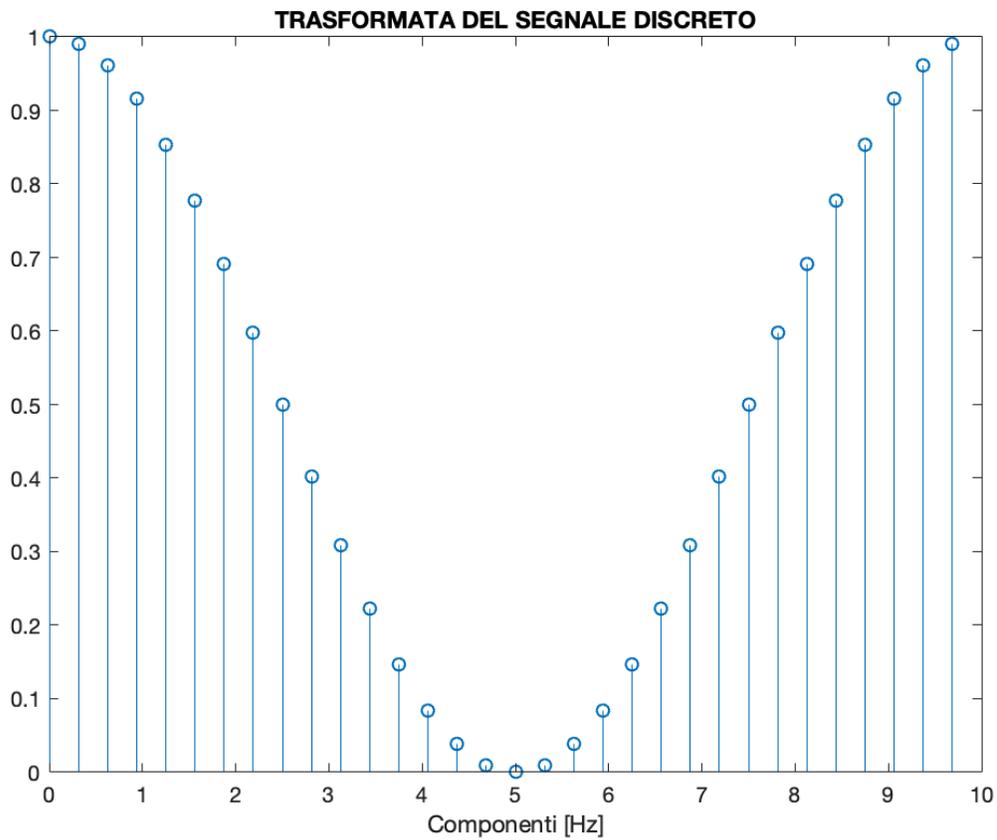
Simulazione segnale discreto (numero di campioni basso)

```
xd=zeros(Nd,1);  
xd(1)=1/2;  
xd(2)=1/4;  
xd(Nd)=1/4;  
  
figure(1); stem(nd,xd);  
title('SEGNALE DISCRETO')  
xlabel('Campioni [s]')
```



Trasformata del segnale discreto

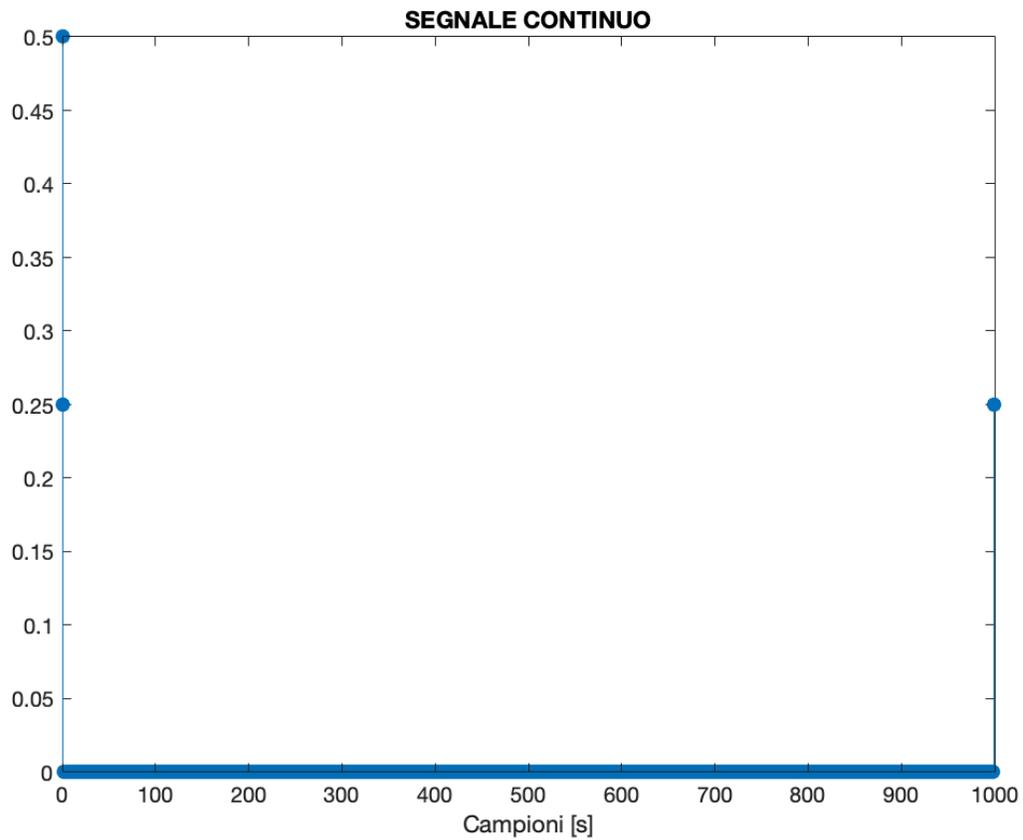
```
Xd=fft(xd);  
  
figure(2); stem(Fd*nd,real(Xd))  
title('TRASFORMATA DEL SEGNALE DISCRETO')  
xlabel('Componenti [Hz]')
```



Simulazione segnale continuo (numero alto di campioni)

```
xc=zeros(Nc,1);
xc(1)=1/2;
xc(2)=1/4;
xc(Nc)=1/4;

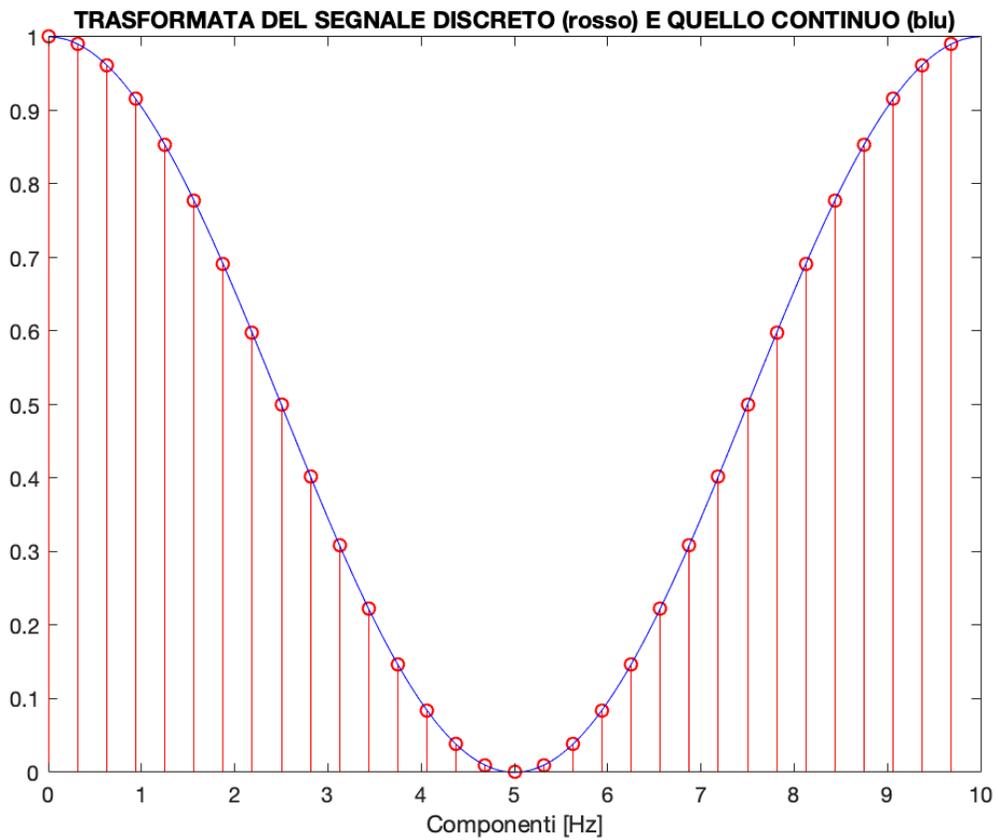
figure(3); stem(nc,xc,'filled');
title('SEGNALE CONTINUO')
xlabel('Campioni [s]')
```



Trasformata del segnale continuo

```
Xc=fft(xc);

figure(4);
stem(Fd*nd,real(Xd),'r'); hold on;
plot(Fc*nc,real(Xc),'b');
title('TRASFORMATA DEL SEGNALE DISCRETO (rosso) E QUELLO CONTINUO (blu)')
xlabel('Componenti [Hz]')
```



Sequenza $-1/4 \ 1/2 \ -1/4$

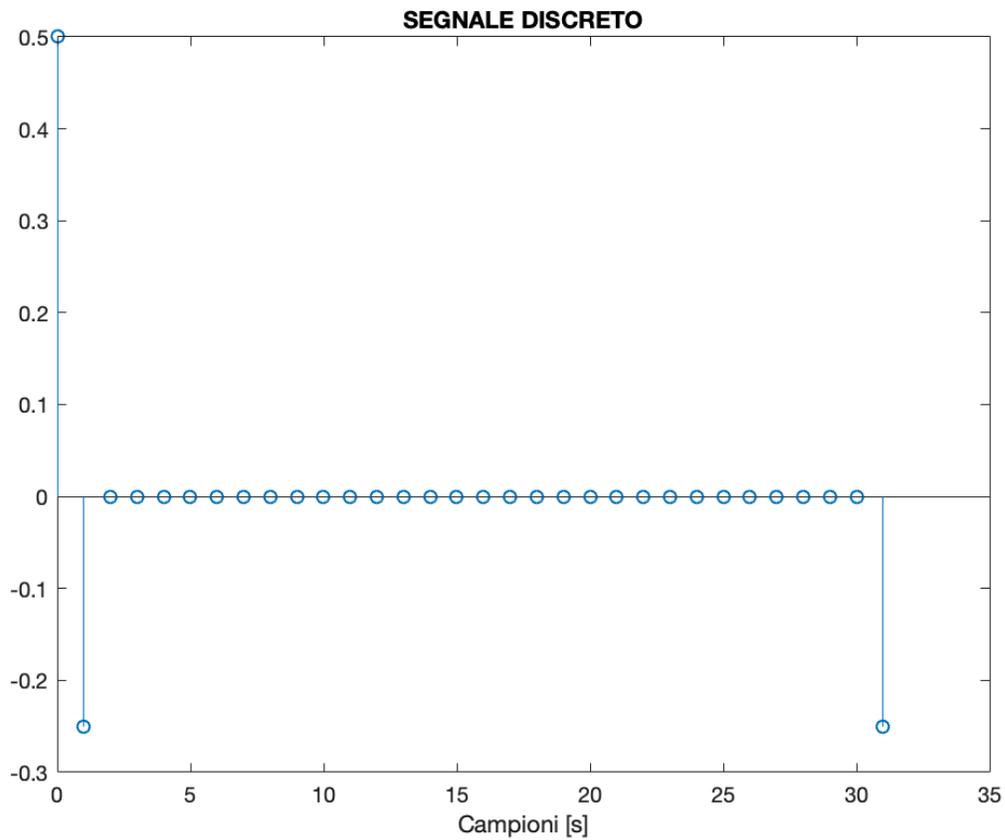
Simulazione segnale discreto

```

xd=zeros(Nd,1);
xd(1)=1/2;
xd(2)=-1/4;
xd(Nd)=-1/4;

figure(5); stem(nd,xd);
title('SEGNALE DISCRETO')
xlabel('Campioni [s]')

```



Trasformata del segnale discreto

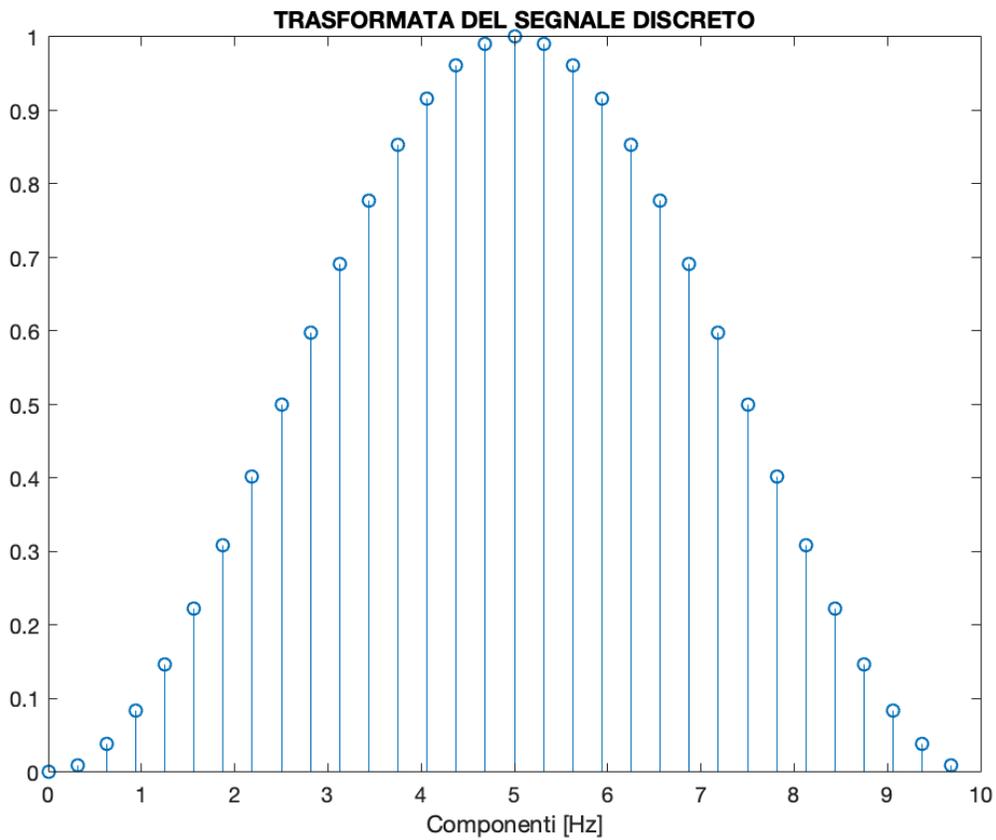
```

Td=Nd*DT;           % lunghezza segnale in secondi
Fd=1/Td;           % passo di campionamento in frequenza

Xd=fft(xd);

figure(6); stem(Fd*nd,real(Xd))
title('TRASFORMATA DEL SEGNALE DISCRETO')
xlabel('Componenti [Hz]')

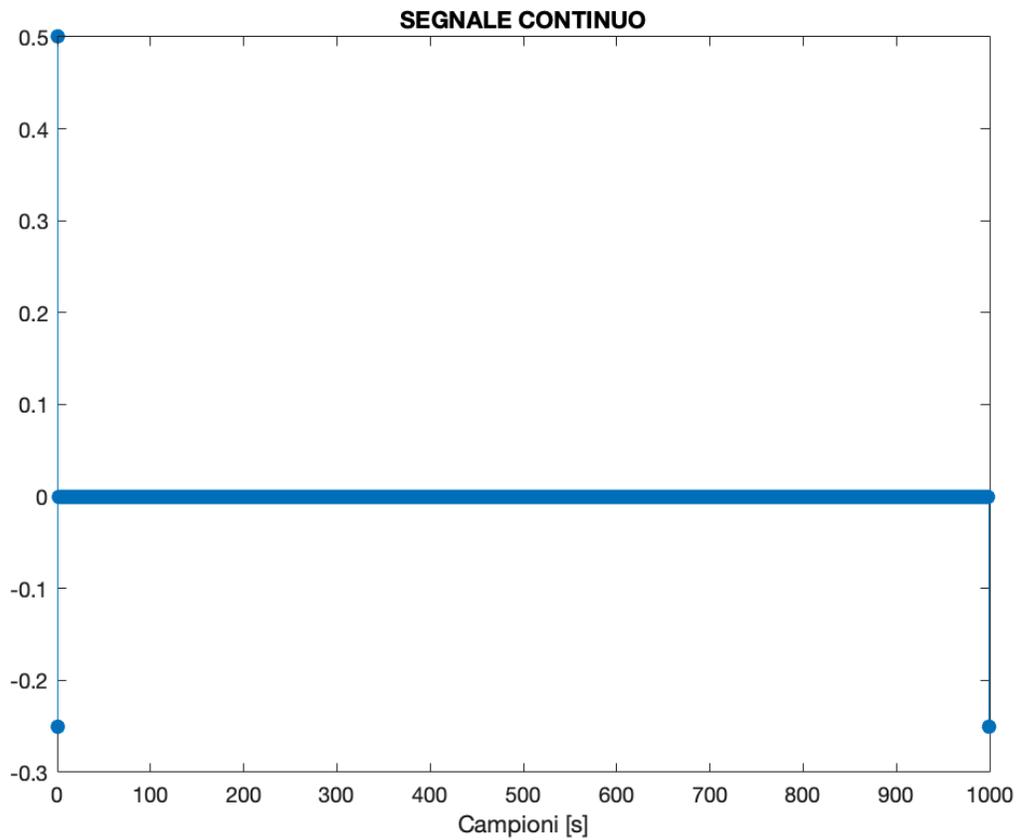
```



Simulazione segnale continuo

```
xc=zeros(Nc,1);
xc(1)=1/2;
xc(2)=-1/4;
xc(Nc)=-1/4;

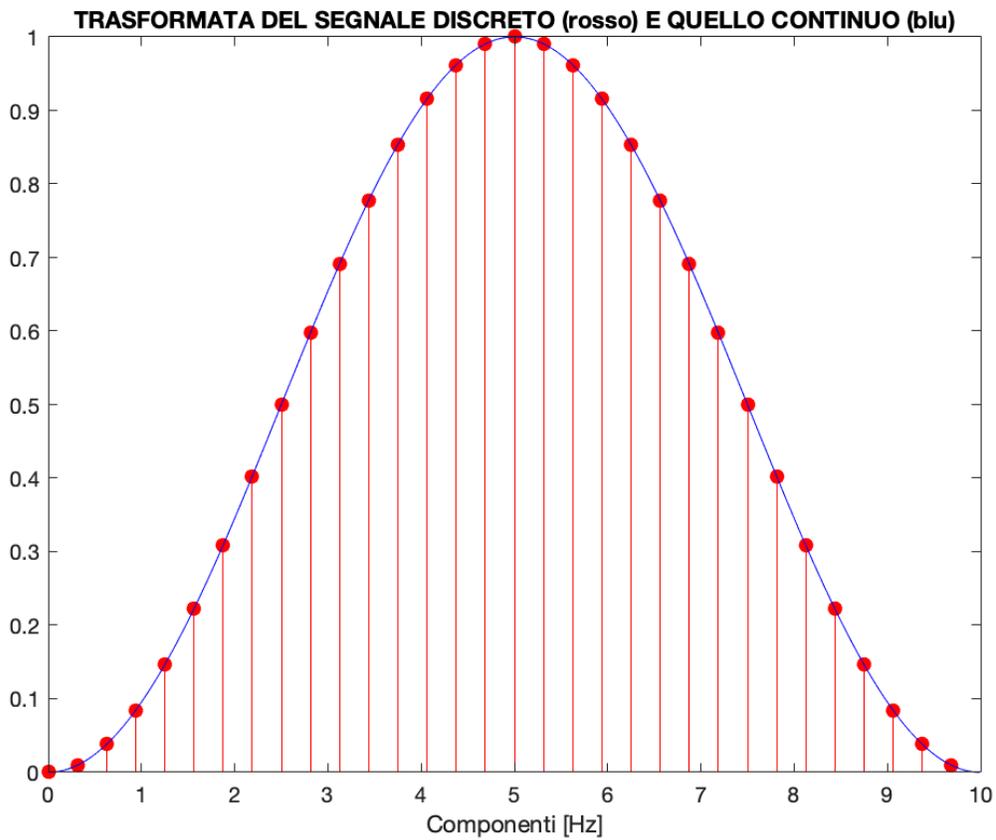
figure(7); stem(nc,xc,'filled');
title('SEGNALE CONTINUO')
xlabel('Campioni [s]')
```



Trasformata del segnale continuo

```
Xc=fft(xc);

figure(8)
stem(Fd*nd,real(Xd),'r','filled'); hold on;
plot(Fc*nc,real(Xc),'b');
title('TRASFORMATA DEL SEGNALE DISCRETO (rosso) E QUELLO CONTINUO (blu)')
xlabel('Componenti [Hz]')
```



Coseno (numero intero di cicli)

Simulazione segnale discreto

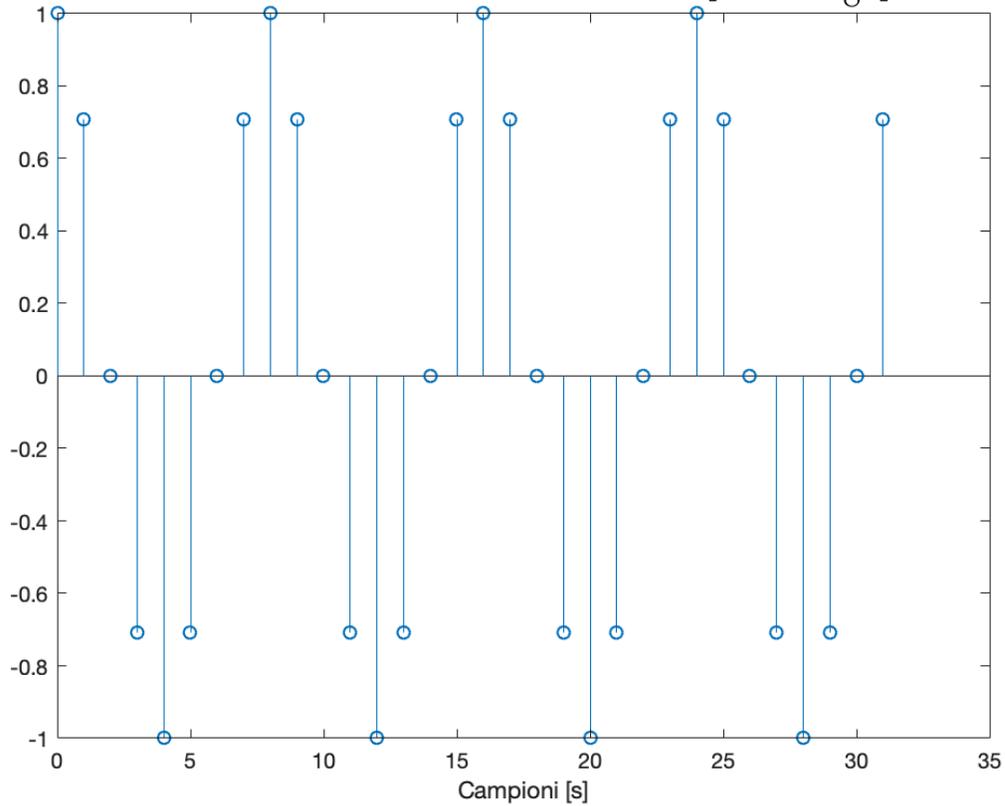
```

xd=cos(2*pi*nd/8);

figure(9);
stem(nd,xd);
xlabel('Campioni [s]')
title('COSENO DISCRETO [ $\omega=\frac{2\pi}{8}$ ]', 'interpreter', 'LaTeX', 'fontSize', 25)

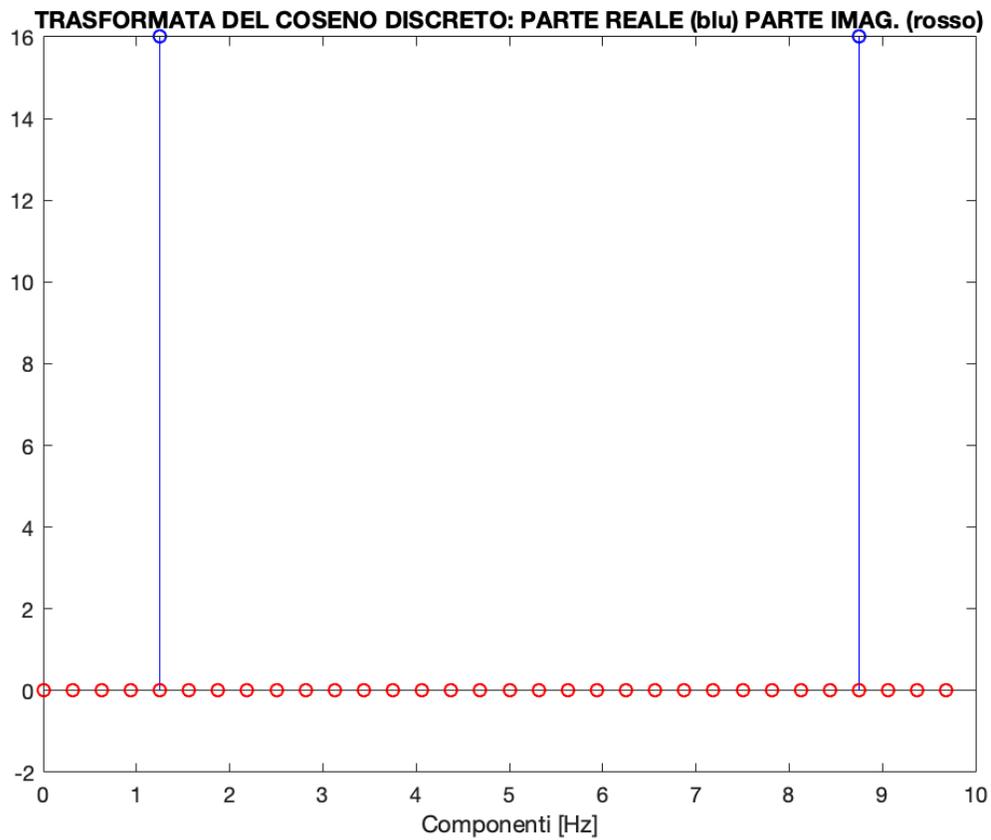
```

COSENO DISCRETO $[\omega = \frac{2\pi}{8}]$



Trasformata segnale discreto

```
Xd=fft(xd);  
  
figure(10);  
stem(Fd*nd,real(Xd),'b'); hold on;  
stem(Fd*nd,imag(Xd),'r');  
xlabel('Componenti [Hz]')  
title('TRASFORMATA DEL COSENO DISCRETO: PARTE REALE (blu) PARTE IMAG. (rosso)')
```



Coseno (numero non intero di cicli)

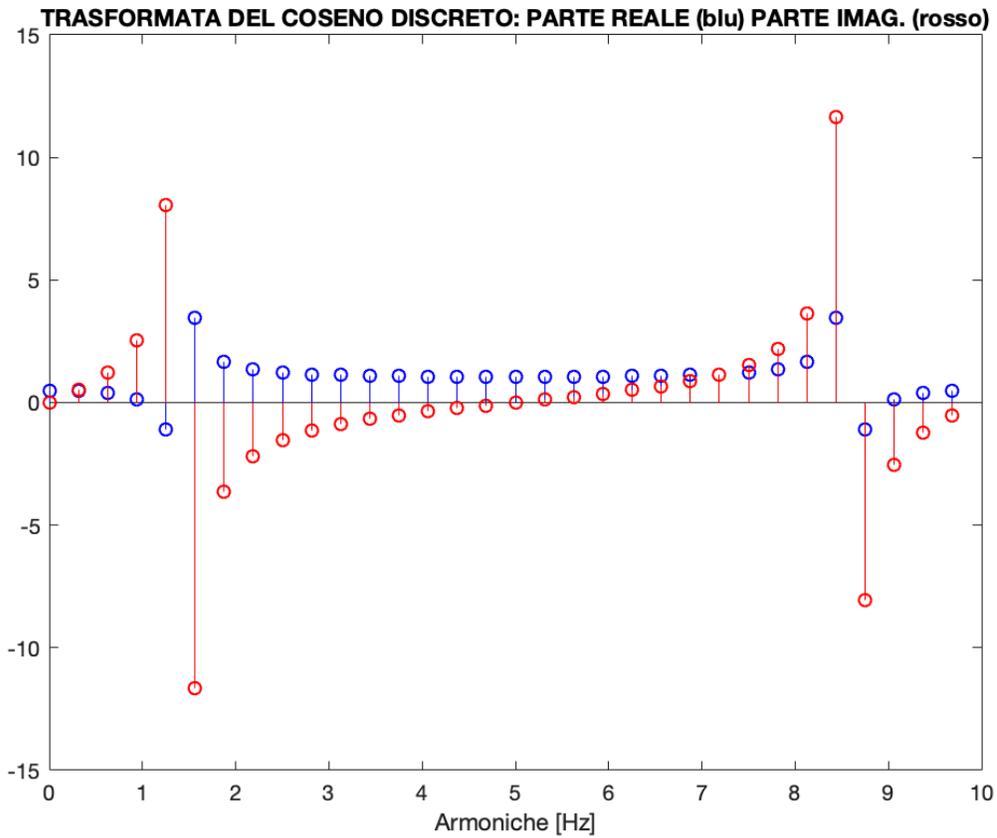
Simulazione coseno discreto

```

xd=cos(2*pi*nd/7);

figure(13);
stem(nd,xd,'filled');
xlabel('Campioni [s]')
title('COSENO DISCRETO [ $\omega=\frac{2\pi}{7}$ ]', 'interpreter','LaTeX','fontSize',25)

```

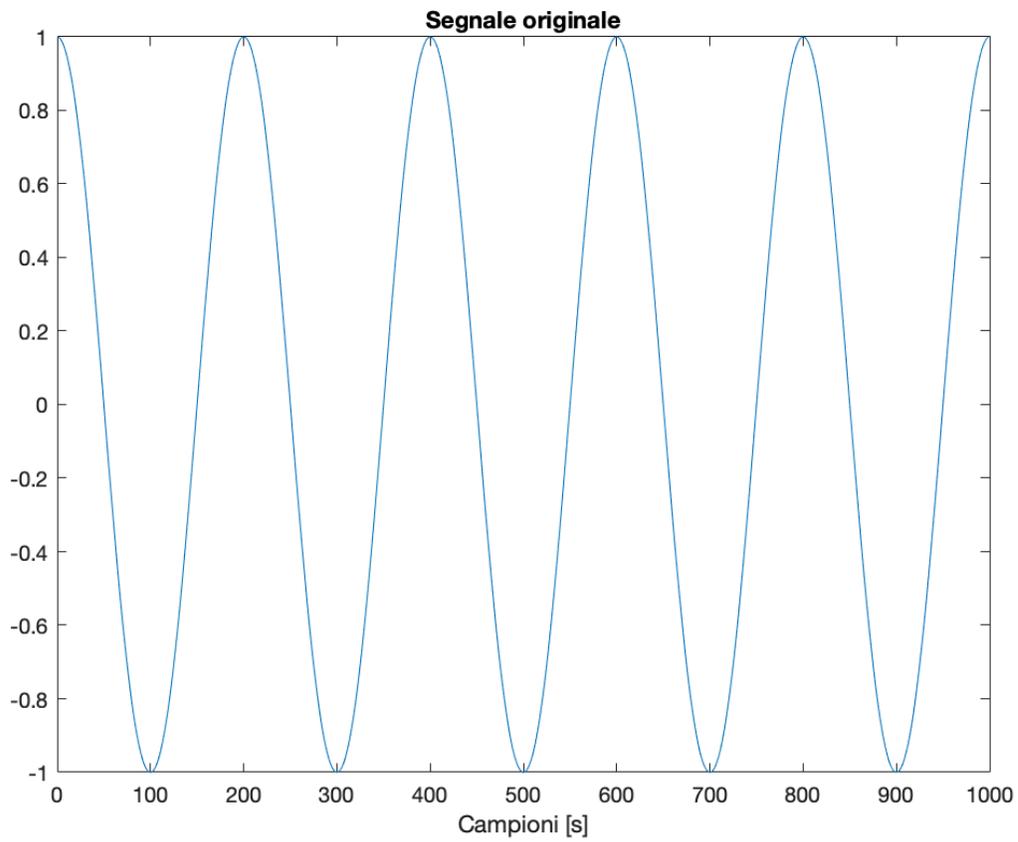



Filtraggio

Simulazione di un segnale sinusoidale

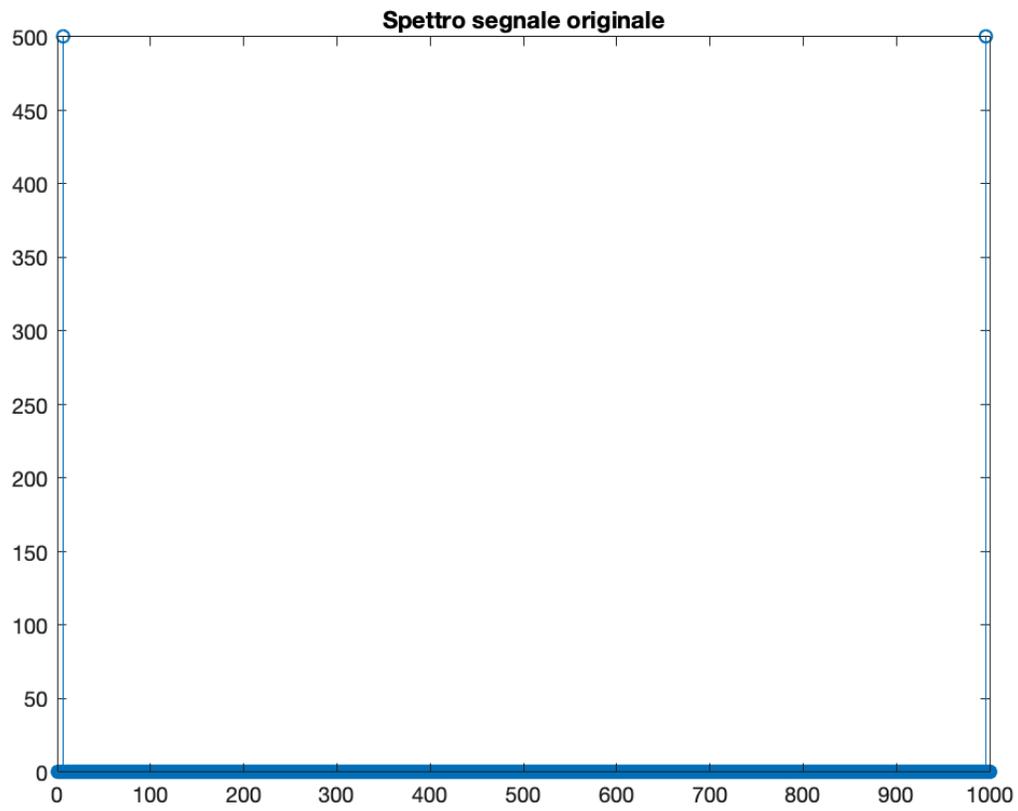
```
T=200;           %periodo
omega=2*pi/T;   %pulsazione
N=1000;         %numero di campioni
n=0:N-1;        %tempo [s]

y=sin(omega*n+(pi/2));
figure; plot(n,y)
xlabel('Campioni [s]')
title('Segnale originale')
```



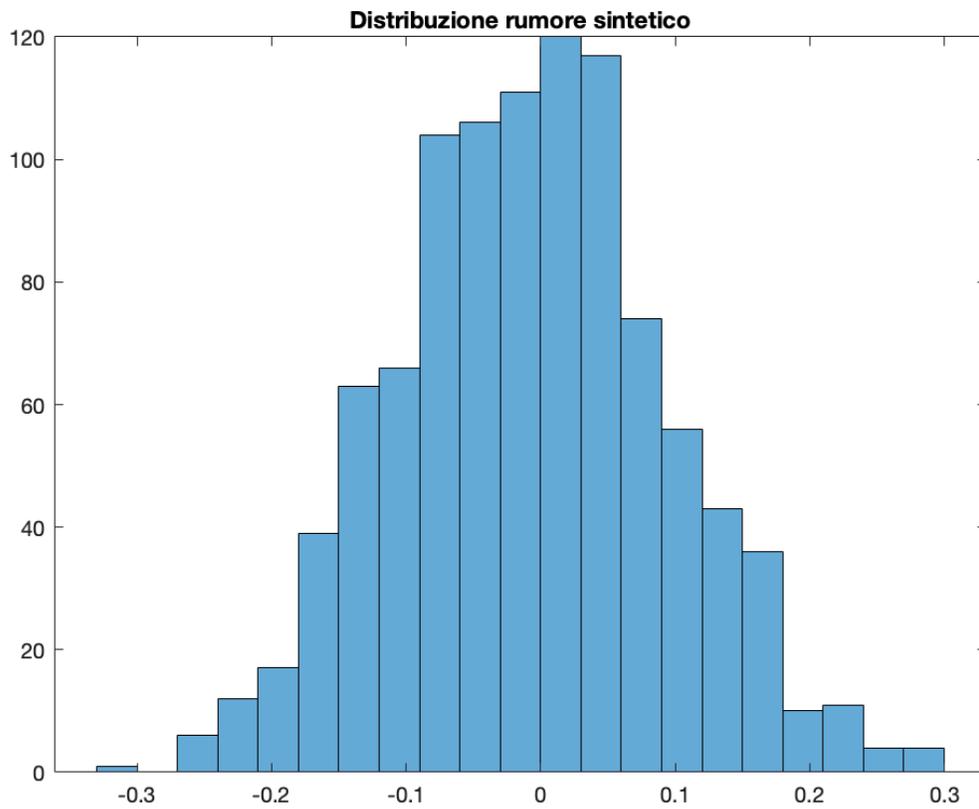
Spettro segnale

```
Hy=fft(y);  
figure; stem(abs(Hy));  
title('Spettro segnale originale')
```



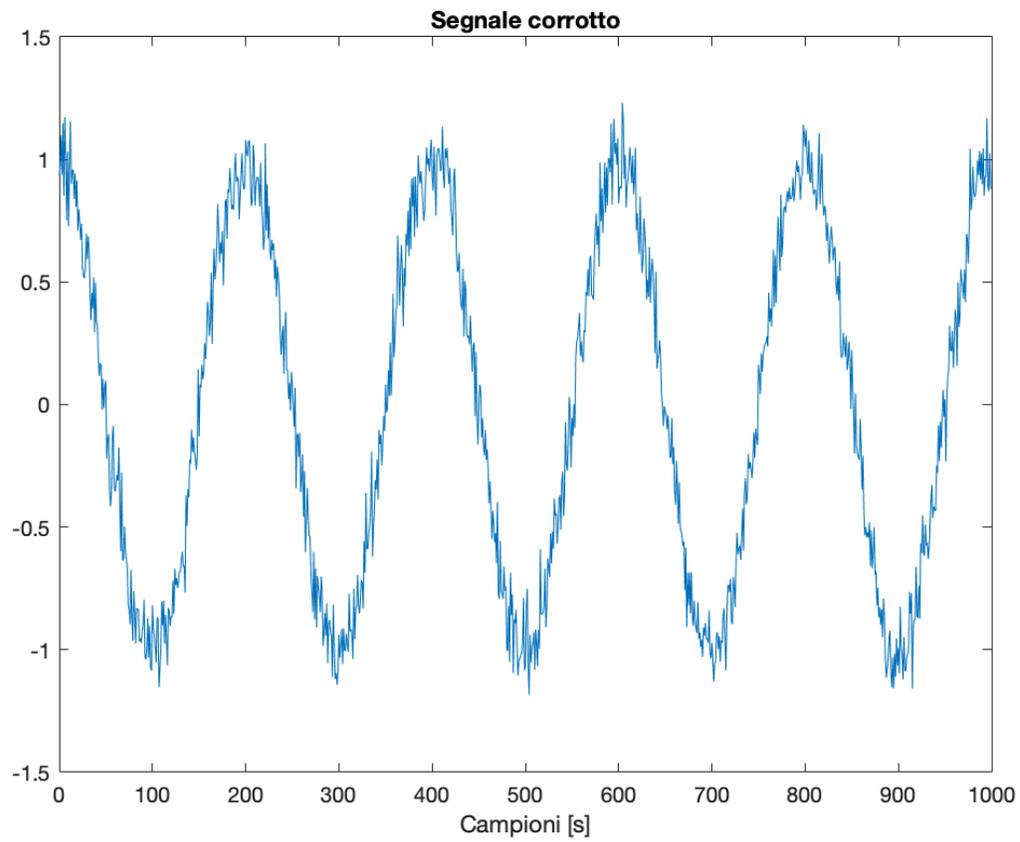
Generazione rumore sintetico

```
e=0.1*randn(1,N);  
figure; histogram(e);  
title('Distribuzione rumore sintetico')
```



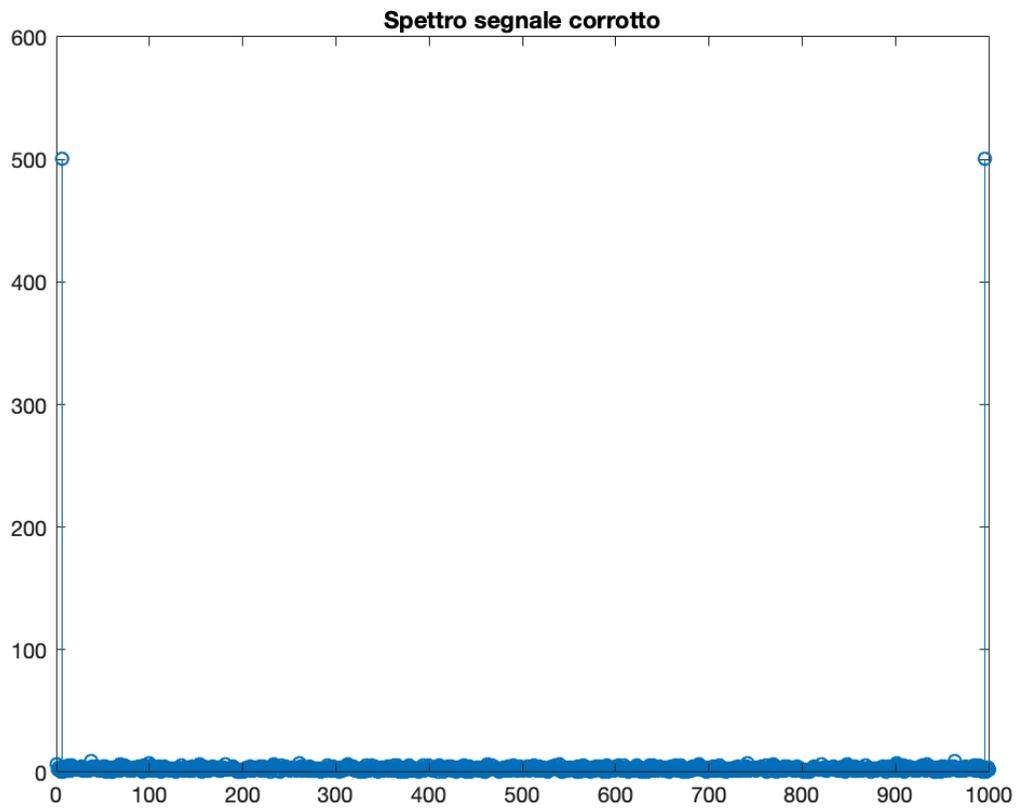
Generazione segnale corrotto (modello additivo)

```
y_=y+e;  
figure; plot(n,y_);  
xlabel('Campioni [s]')  
title('Segnale corrotto')
```



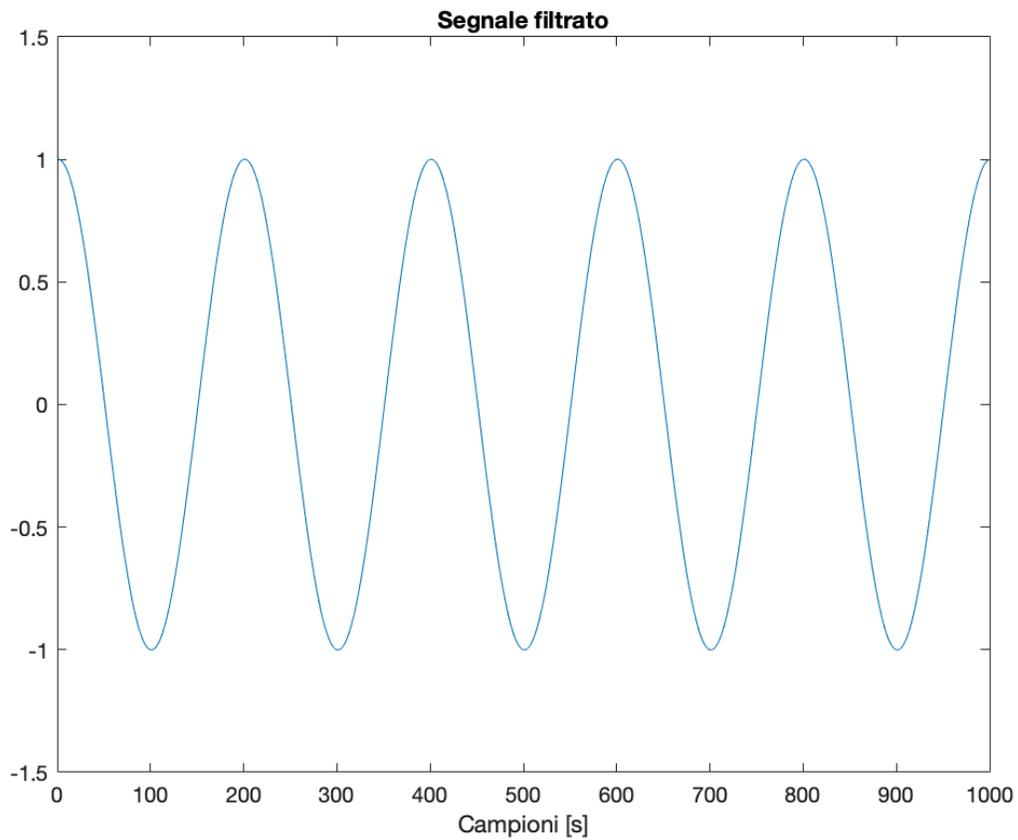
Spettro segnale corrotto

```
Hy_=fft(y_);  
figure; stem(abs(Hy_))  
title('Spettro segnale corrotto')
```



Filtraggio

```
Hf=Hy_.*(abs(Hy)>100);  
yf=ifft(Hf);  
  
figure; plot(yf)  
xlabel('Campioni [s]')  
title('Segnale filtrato')
```



Zero-padding interpolation

Generazione segnale

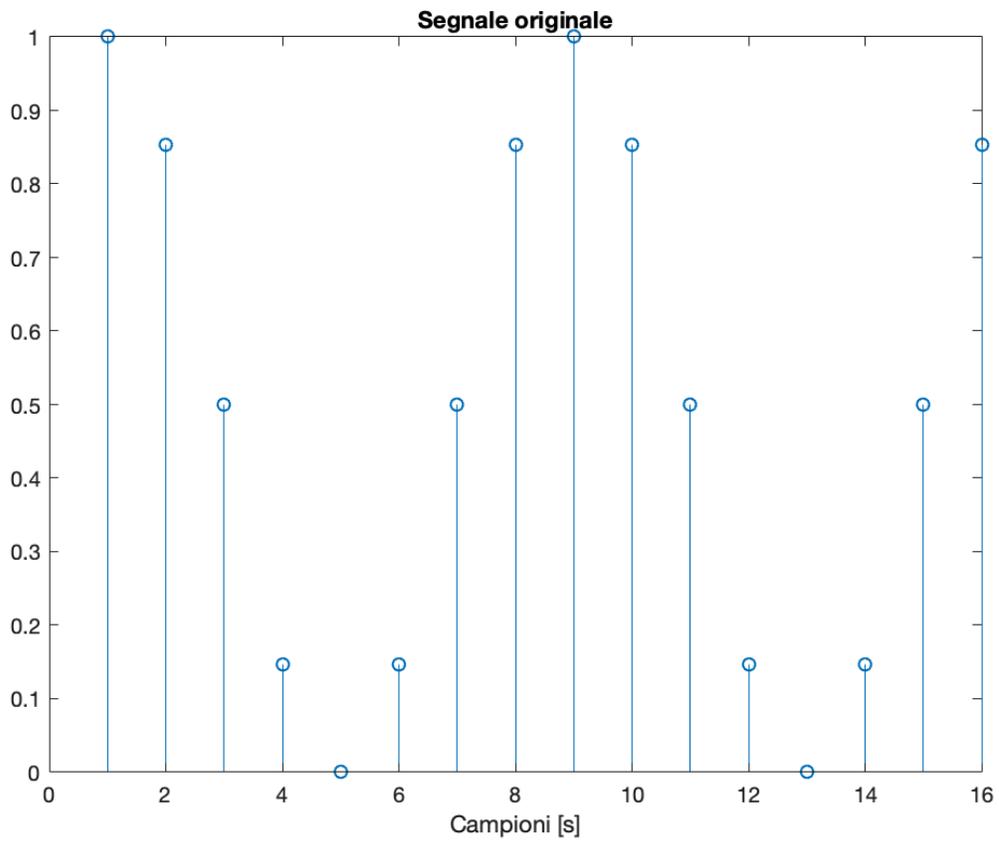
```

N=16;           %numero campioni
n=0:N-1;       %tempo [s]
O=10;          %fattore di sovracampionamento

x=cos(((2*pi)/N)*n).^2;

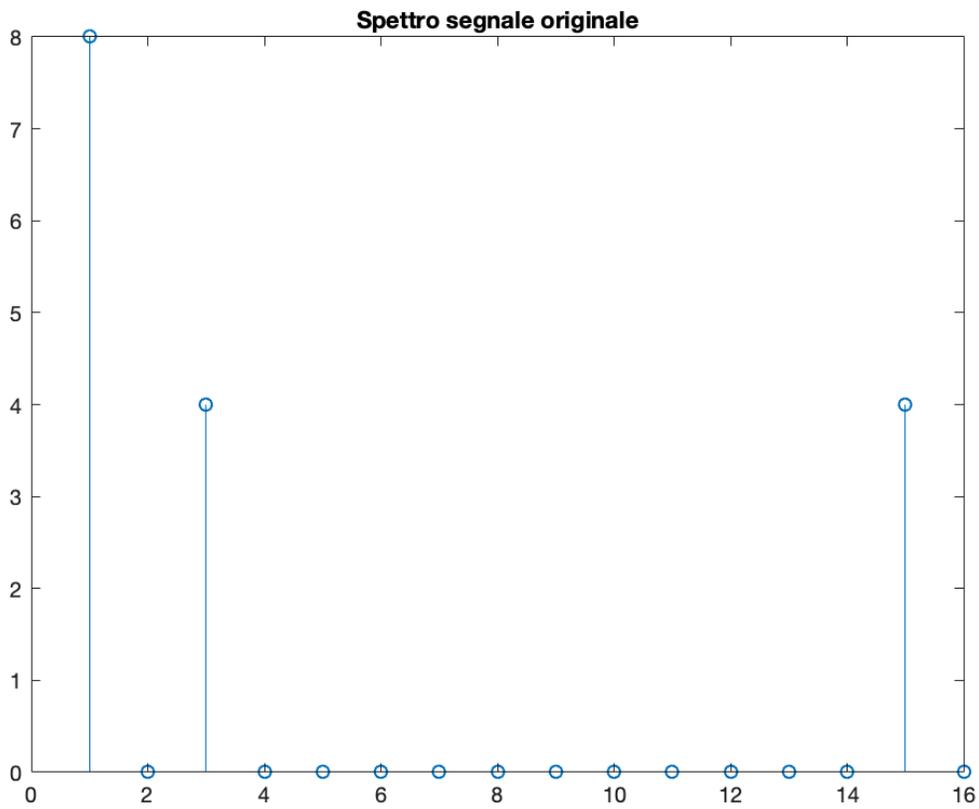
figure; stem(x)
xlabel('Campioni [s]')
title('Segnale originale')

```



Spettro segnale originale

```
X=fft(x);  
figure; stem(abs(X))  
title('Spettro segnale originale')
```



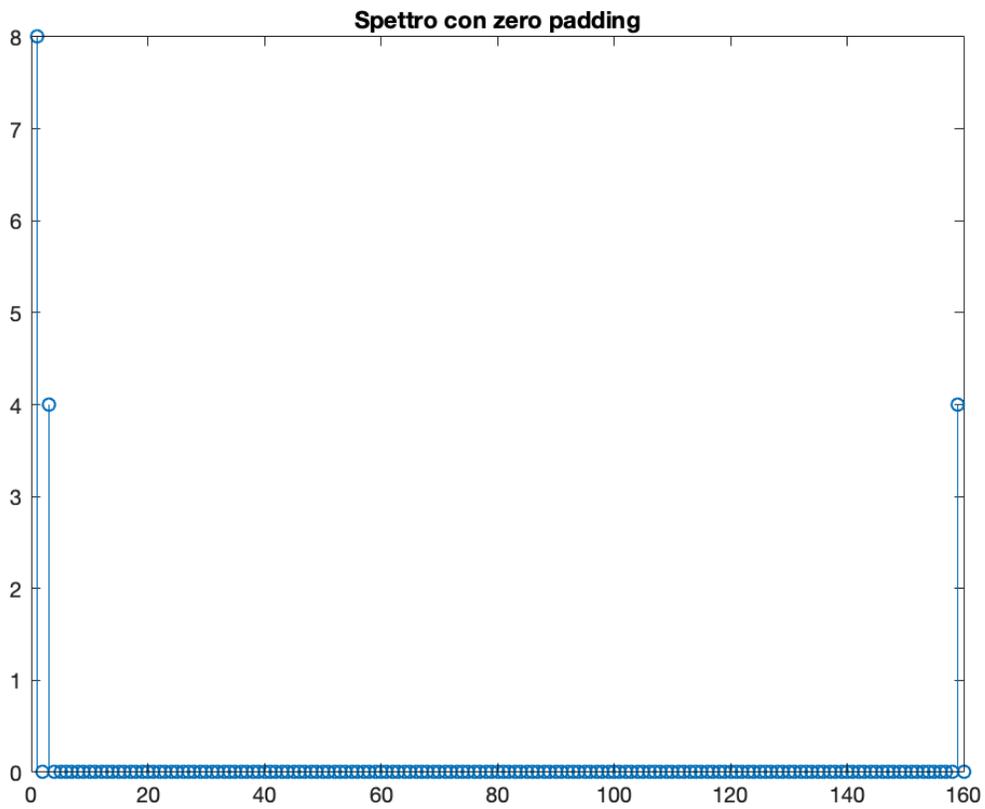
Zero-padding dello spettro

```

XPad=zeros(1,N*O);

XPad(1:N/2)=X(1:N/2);
XPad(N*O:-1:N*O-N/2+1)=X(N:-1:N/2+1);
figure; stem(abs(XPad))
title('Spettro con zero padding')

```



Restituzione segnale interpolato

```
y=ifft(XPad*O);  
  
figure; stem(y,'b'); hold on;  
  
xlabel('Campioni [s]')  
title('Segnale originale (rosso) vs segnale interpolato (blu)')
```

