Esercizio 6 – MatLab – [punti 3]

Si considerino i segnali reali a tempo continuo $x(t)$ e $y(t)$ ad estensione limitata, i cui campioni siano rappresentati in MatLab dai vettori `x` e `y`, con rispettivi tempi di campionamento `tx` e `ty` e con passo di campionamento comune `T` scelto opportunamente.

Si chiede di ideare un semplice script MatLab per derivare e poi disegnare il segnale convoluzione $z(t) = x * y(t)$.

Soluzione Lo script potrebbe essere

```
tz = tx(1)+ty(1):T:tx(end)+ty(end); % regola di estensione della conv.
z = T*conv(x,y); % operazione di convoluzione

plot(tz,z); % plot della convoluzione
```

Esercizio 6 – [punti 3]

Si consideri un segnale a tempo continuo $x(t)$ **reale** e **causale** e sia $X(f)$ la sua trasformata di Fourier; si assuma che il vettore MatLab `X`, di lunghezza `N` (con `N` un numero pari), contenga i campioni di $X(f)$ in corrispondenza delle frequenze `f=F*(-N/2:N/2-1)` in cui il passo di campionamento `F` sia dato.

Si chiede di ideare un semplice script MatLab che calcoli numericamente il segnale $x(t)$ e i tempi associati, quindi ne dia una rappresentazione grafica.

Soluzione Lo script potrebbe essere

```
T = 1/(N*F); % passo di campionamento nel tempo
x = ifft(fftshift(X))/T; % antitrasformata di Fourier
t = (0:N-1)*T; % tempi associati ai campioni del segnale

plot(t,real(x)); % plot del segnale
```

Esercizio 6 – [punti 3]

Si consideri un segnale a tempo continuo $x(t)$ **reale** e **causale** e sia $X(j\omega)$ la sua trasformata di Fourier; si assuma che il vettore MatLab `X`, di lunghezza `N` (con `N` un numero pari), contenga i campioni di $X(j\omega)$ in corrispondenza delle pulsazioni ω=ω_0`*(-N/2:N/2-1)` in cui ω_0 sia dato e pertanto sia noto il passo di campionamento nel tempo `T = 2*pi/(N*`ω_0`)`.

Si chiede di ideare un semplice script MatLab che calcoli numericamente il segnale $x(t)$ e i tempi associati, quindi ne dia una rappresentazione grafica.

Soluzione Lo script potrebbe essere

```
x = ifft(fftshift(X))/T; % antitrasformata di Fourier
t = (0:N-1)*T; % tempi associati ai campioni del segnale

plot(t,real(x)); % plot del segnale
```

Esercizio 6 – MatLab – [punti 3]

Si consideri un segnale reale a tempo continuo $x(t)$ ad estensione limitata, i cui campioni, presi con passo di campionamento `T`, siano rappresentati in MatLab dal vettore `x`.

Si chiede di ideare un semplice script MatLab che derivi numericamente la trasformata di Fourier $X(f)$ e le frequenze associate, quindi ne dia una rappresentazione grafica.

Soluzione Lo script potrebbe essere

```
Nx = length(x); % numero di campioni del segnale
fx = (0:Nx-1)/(Nx*T); % campioni nel dominio della frequenza
X = T*fft(x); % trasformata di Fourier

semilogy(fx,abs(X)); % plot della trasformata di Fourier
```

Esercizio 6 – [punti 3]

Si consideri un segnale reale a tempo continuo $x(t)$ ad estensione limitata, i cui campioni, collezionati con passo di campionamento `T`, siano contenuti nel vettore `x`.

Si chiede di ideare un semplice script MatLab che derivi numericamente la trasformata di Fourier $X(j\omega)$ e le pulsazioni associate, quindi ne dia una rappresentazione grafica.

Soluzione Lo script potrebbe essere

```
Nx = length(x); % numero di campioni del segnale
omx = (0:Nx-1)*2*pi/(Nx*T); % campioni nel dominio della frequenza
X = T*fft(x); % trasformata di Fourier
semilogy(omx,abs(X)); % plot della trasformata di Fourier
```

Esercizio 6 – [punti 3]

Si desidera filtrare numericamente un segnale reale a tempo continuo $x(t)$, ad estensione limitata con un filtro che voglia mantenere le sole frequenze positive nell'intervallo $[2, 4]$ Hz. Si assuma che i campioni del segnale $x(t)$ siano disponibili nel vettore MatLab `x`, con passo di campionamento `T=0.1` e lunghezza `N=5000`;

Si chiede di ideare un semplice script MatLab che applichi il filtraggio in frequenza e ritorni il segnale filtrato nel tempo.

Soluzione Lo script potrebbe essere

```
X = T*fft(x); % trasformata di Fourier
f = (-N/2:N/2-1)/(N*T); % frequenze
X(f<20|f>40) = 0; % filtraggio
y = 1/T*ifft(X); % antitrasformata di Fourier
```