

Corso di Elaborazione numerica dei segnali
Anno accademico 2002/2003.
Prof. Pasquale Daponte.

Esercitazione n°1
Generazione di segnali.

1) Visualizzare l'andamento di una funzione.

In MATLAB una funzione deve essere costruita per punti. E' necessario quindi, prima, definire il vettore dei punti su cui la funzione deve essere calcolata:

```
t=linspace(0,0.01,500);
```

Poi si costruisce il vettore dei valori della funzione:

```
x=3*sin(2*pi*1000*t);
```

Il vettore x contiene 500 campioni di una sinusoide di ampiezza 3 e frequenza 1 kHz. Per visualizzare il grafico della funzione si può usare il comando plot:

```
plot(t,x);
```

Il grafico della funzione è mostrato in figura 1 :

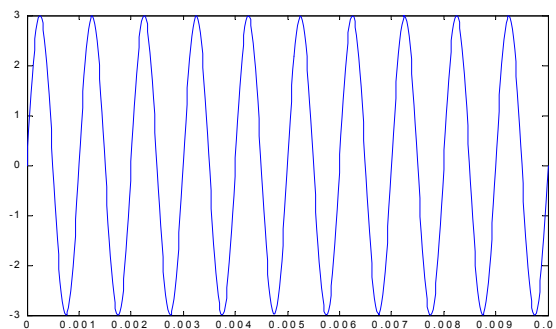


Figura 1 - Grafico della funzione seno.

2) Generare un'onda quadra.

In MATLAB è possibile scrivere procedure contenenti un insieme di istruzioni (script files). Tali procedure verranno eseguite richiamando il file corrispondente dalla command window.

Gli script files operano direttamente sulle variabili attualmente in memoria e non consentono l'uso di variabili locali. In pratica è come scrivere un insieme di istruzioni MATLAB e poi farle eseguire in blocco.

Il seguente script genera e visualizza un'onda quadra di ampiezza unitaria e periodo 100 campioni.

```
t=linspace(0,0.01,500); % vettore dei tempi
A=1; % valore del primo campione
count=0;
for i=1:500,
    if count<50 % Per 50 campioni si mantiene
        count=count+1; % costante il campione corrente
    else
        count=0; % Ogni 50 campioni si inverte
        A=-A; % l'ampiezza
    end
    x(i)=A;
end
plot(t,x); % visualizza l'onda generata
axis([0 0.01 -1.2 1.2]);
```

In figura 2 è mostrato il risultato dello script.

Con il comando axis è possibile modificare la visualizzazione del grafico.

La sintassi del comando è la seguente:

```
axis([xmin xmax ymin ymax])
```

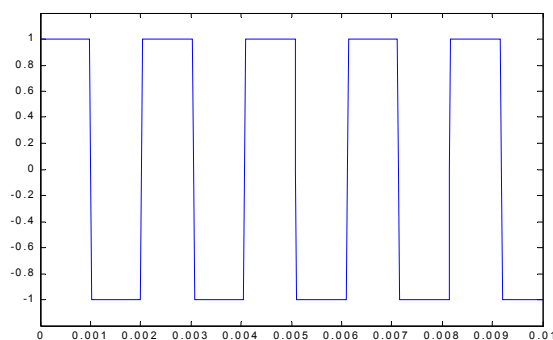


Figura 2 - Generazione di un onda quadra.



Verifica – Provate a generare e visualizzare, allo stesso modo dell'onda quadra, un'onda triangolare ed una a dente di sega.

Lo stesso risultato può essere ottenuto scrivendo una `function` invece che uno script. Una function si crea in MATLAB in maniera analoga ad uno script, tranne che per l'intestazione che deve essere:

```
function [variabili di uscita]=nomeFunzione(variabili di ingresso)
```

Le variabili interne sono tutte locali e quindi dopo l'esecuzione della function non restano nello spazio di lavoro.

Per generare un'onda quadra si può scrivere la seguente function:

```
function x=ondaQuadra(A,Toss,f,fc)
t=linspace(0,Toss,round(Toss*fc)); % vettore dei tempi
A=1; % valore del primo campione
count=0;
for i=1:length(t),
    if count<round(fc/(2*f)-1) % Per mezzo ciclo si mantiene
        count=count+1; % costante il campione corrente
    else
        count=0; % Ogni mezzo ciclo si inverte
        A=-A; % l'ampiezza
    end
    x(i)=A;
end
plot(t,x); % visualizza l'onda generata
```

Le functions vanno salvate in m-files, files il cui nome è *nomeFunzione* con l'estensione `.m` e possono essere utilizzate come istruzioni dalla command window o all'interno di altre funzioni o script files. La sintassi dell'istruzione corrispondente è quella specificata nel prototipo.

3) Modulazione di ampiezza.

Un segnale modulato in ampiezza si può ottenere mediante la formula:

$$v_{AM}(t) = (V_p + V_m \cos \omega_m t) \cos \omega_p t,$$

dove V_p , ω_p , V_m ed ω_m sono rispettivamente l'ampiezza e la pulsazione della portante e l'ampiezza e la pulsazione del segnale modulante.

Il seguente script genera un segnale modulato in ampiezza in cui la modulante è una senoide.

```
t=linspace(0,0.02,1000); % vettore dei tempi
xp=cos(2*pi*2000*t); % onda portante
xm=cos(2*pi*200*t); % onda modulante
xam=(4+1.5*xm).*xp; % segnale modulato in ampiezza
plot(t,xam);
```

Il risultato dello script è mostrato in figura 3.

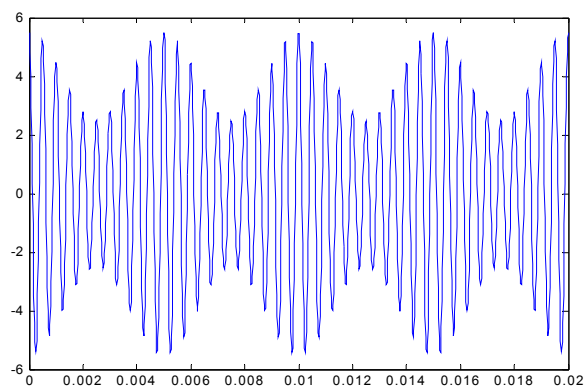


Figura 3 - Segnale modulato in ampiezza.

4) Modulazione di frequenza.

Un segnale modulato in frequenza si può ottenere mediante la formula:

$$v_{FM}(t) = V_p \cos(\omega_p t + m \sin \omega_m t),$$

Il seguente script genera un segnale modulato in frequenza in cui la modulante è una senoide.

```
t=linspace(0,0.02,1000); % vettore dei tempi
xm=sin(2*pi*200*t); % onda modulante
xfm=4*cos(2*pi*2000*t+0.5*xm); % segnale modulato in frequenza
plot(t,xfm);
```

Il risultato dello script è mostrato in figura 4.

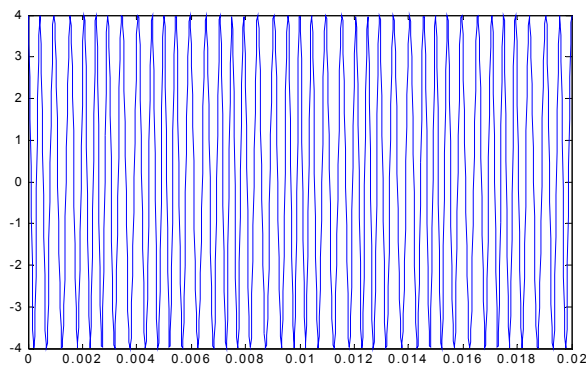


Figura 4 - Segnale modulato in frequenza.

5) Generazione di un segnale aleatorio uniforme.

In MATLAB è possibile generare un vettore di variabili aleatorie uniformi mediante il comando `rand`. La sintassi è la seguente:

```
rand(M,N)
```

Questa funzione genera una matrice $M \times N$ con elementi casuali, scelti da una distribuzione uniforme sull'intervallo $[0,1]$.

I seguenti comandi MATLAB consentono di generare 1000 campioni di un segnale aleatorio gaussiano e visualizzare l'istogramma delle ampiezze (figura 5).

```
x=rand(1000,1);  
hist(x,20);
```

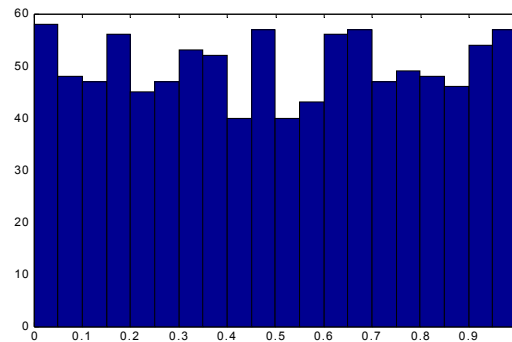


Figura 5 - Iistogramma delle ampiezze di un segnale aleatorio uniforme.

Si può notare come, all'aumentare del numero di campioni, l'istogramma diventa sempre più piatto.

6) Generazione di un segnale aleatorio gaussiano.

Un segnale aleatorio gaussiano si può generare in maniera analoga a quello uniforme, mediante il comando `randn`:

```
randn (M, N)
```

`randn` genera una matrice $M \times N$ di elementi scelti da una distribuzione gaussiana a media nulla e varianza unitaria.

La sequenza di comandi seguente genera un segnale aleatorio gaussiano a media nulla e varianza 4

```
x=2*randn (1000, 1) ;  
hist (x, 20) ;
```

In figura 6 è mostrato l'istogramma delle ampiezze del segnale generato.

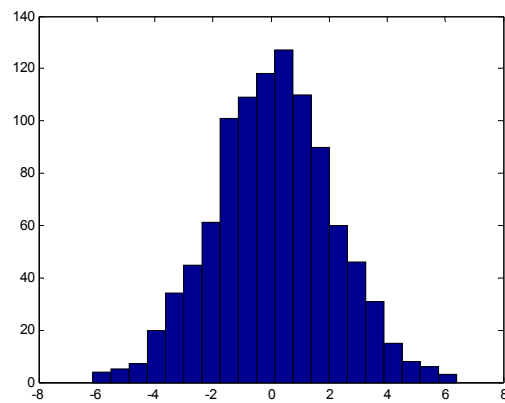


Figura 6 - Iistogramma delle ampiezze di un segnale aleatorio gaussiano.

7) Aggiunta di rumore bianco ad un segnale.

La seguente funzione aggiunge al segnale x rumore bianco gaussiano con il valore di SNR indicato nei parametri.

```
function xn=addNoise(x,snr)
Px=var(x); % Calcola la potenza del segnale
Pn=Px/(10^(snr/10)); % calcola la potenza del rumore
n=sqrt(Pn)*randn(length(x),1); % genera un segnale
% gaussiano
xn=x+n; % somma il rumore al segnale
```

In figura 7 è mostrato un segnale sinusoidale con l'aggiunta di rumore bianco gaussiano con SNR=20 dB.

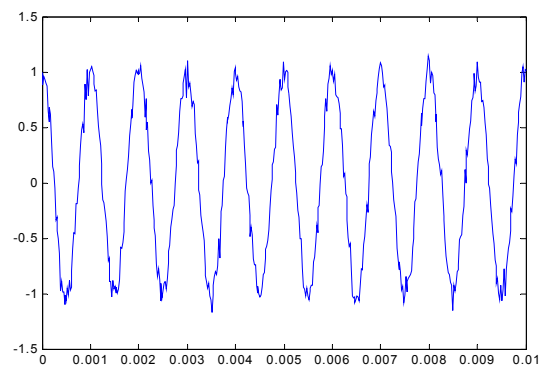


Figura 7 - Rumore sovrapposto ad un segnale sinusoidale.