

---

## Esercitazione 6: Filtro RC numerico

---

Scrivere un programma C di nome `esercizio6.c` che, a partire da un segnale composto dalla somma di due sinusoidi, ne esegua il filtraggio emulando un filtro RC. Successivamente graficare il segnale in ingresso e in uscita al filtro mediante il software MATLAB.

Ricordarsi di includere `math.h`...

### Generazione del segnale sinusoidale più disturbo

Generare due sinusoidi di ampiezza  $A$ , frequenze  $f_0$  e  $10 * f_0$  e durata  $D$ .

Le variabili  $A$  [V],  $f_0$  [Hz] e  $D$  [intero] sono fornite mediante standard input.

Sia  $D$  il numero di periodi (intero) che si vuole generare (per esempio  $D=2$ ). Il passo di campionamento  $dt$  deve essere pari a  $T/20$  con  $T=1/(10*f_0)$  il periodo della sinusoide a frequenza più alta (**riprendere l'esercitazione 4 o 5 con minime modifiche**).

In particolare, il segnale (utile più disturbo) sarà generato nel seguente modo:

$$x[i] = A * \cos(2 * \pi * f_0 * t[i]) + A * \cos(2 * \pi * 10 * f_0 * t[i])$$

### Filtro RC numerico

Si richieda da standard input la frequenza di taglio del filtro in Hz.

Il filtro RC opportunamente discretizzato assume la forma ricorsiva:

$$y[i] = \frac{1}{K+1}x[i] + \frac{K}{K+1}y[i-1]$$

dove  $y[i]$  rappresenta l'uscita filtrata mentre  $K = \tau/dt$ , dove  $\tau$  è la costante di tempo del filtro.

A tal fine, si scriva la funzione:

```
double* filtrc(double* x, double dt, double tau, int N)
```

che prende in ingresso un puntatore a `double`, corrispondente all'array  $x$ , il passo  $dt$ , la costante di tempo del filtro  $\tau$  e un intero  $N$  che corrisponde al numero di campioni relativi al segnale in uscita. La funzione deve restituire in uscita un puntatore a `double` (il vettore  $y$ ). Generare il vettore  $y$ , mediante allocazione dinamica della memoria (`double *`).

## Grafico del segnale e dell'uscita filtrata

Per poter graficare il segnale e l'uscita filtrata sarà necessario aprire in scrittura (utilizzando la funzione fopen) il file `filtratoRC.m` e scrivere al suo interno le seguenti informazioni:

```
t = [0.0000, 0.0500, 0.1000 ..., 2.9500 ];  
x = [1.0000, 0.9511, 0.8090, ..., 0.9511 ];  
t2 = [0.0000, 0.0500, 0.1000, ..., 2.9000 ];  
y = [0.04878, 0.09278, 0.12770 ..., -0.04878 ];  
  
plot(t, x);  
  
hold on, plot(t2, y);
```

Una volta creato il file, aprirlo con MATLAB e cliccare su Editor->Run per visualizzare i grafici.

Incollare il grafico qui sotto.

Questo file convertito in PDF deve essere caricato insieme al codice sviluppato e al file `filtratoRC.m`.

[www.fixia.it](http://www.fixia.it)

## Opzionale

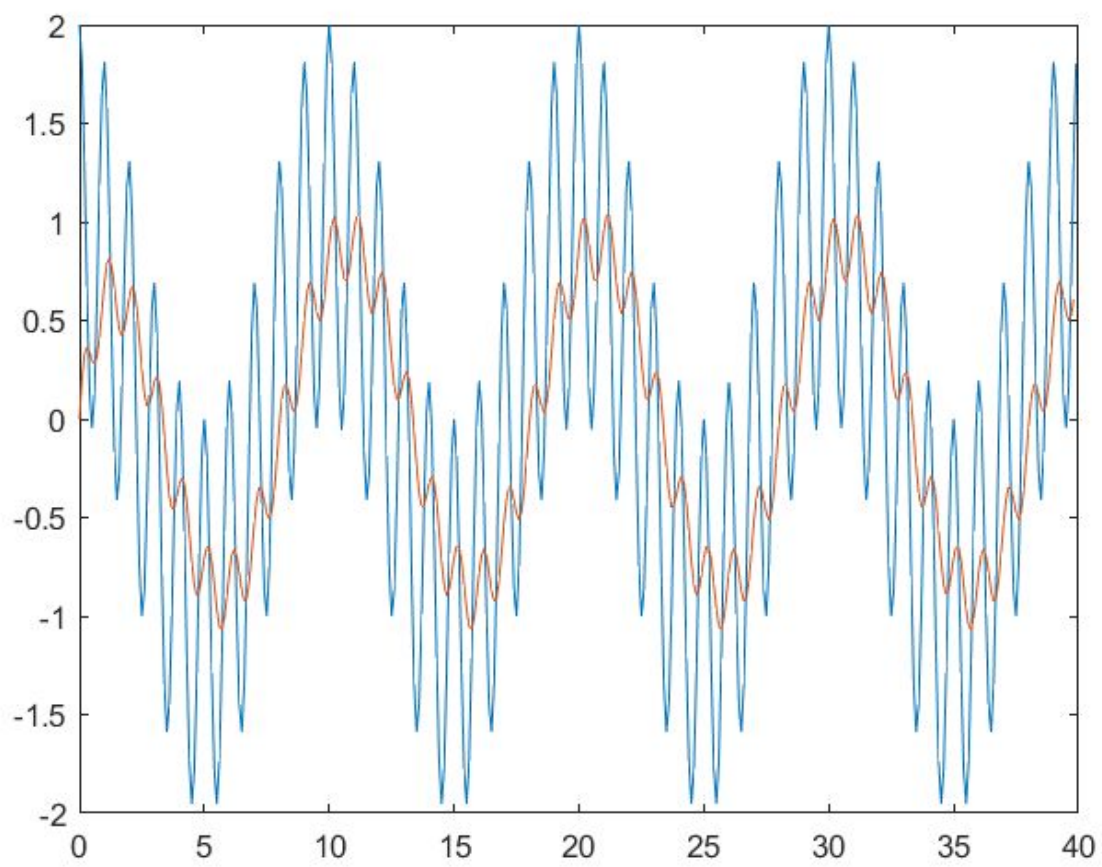
Generare un segnale periodico rettangolare con livelli +1 e 0 fornendo da standard input le seguenti informazioni:

- Duty cycle (dc)
- Periodo (T)
- Durata del segnale come numero intero di periodi (D)
- Numero di campioni per periodo (N)

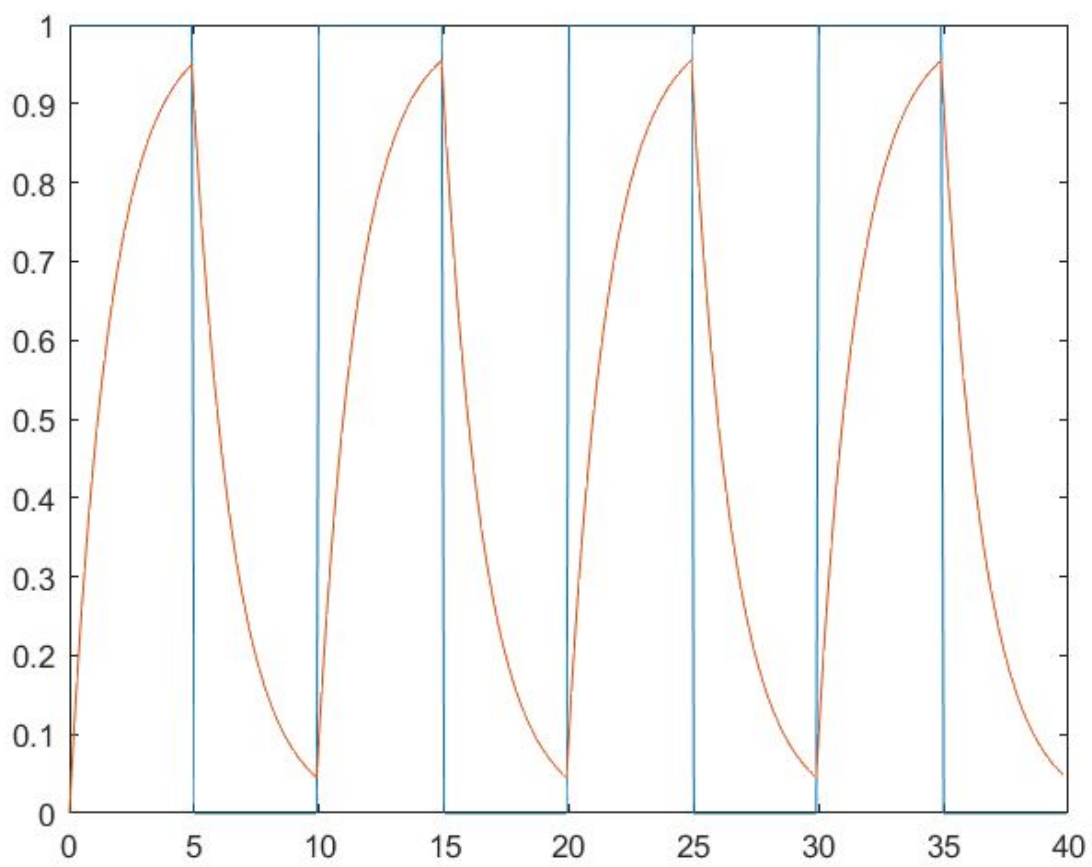
Passare il segnale così generato attraverso il filtro RC.

Graficare nuovamente l'uscita utilizzando Matlab e osservare cosa succede al segnale in uscita dall'integratore al variare del duty cycle (esempio  $dc=0.1$ ,  $dc=0.5$  e  $dc=0.9$ ).

Incollare il grafico ottenuto per  $dc=0.5$  qui sotto.



Segnale a 100Hz(100Hz + 1000Hz) filtrato su  $F_t = 200\text{Hz}$



Segnale quadrato  $f=100\text{Hz}$  e filtrato su  $F_t=100\text{Hz}$