

## CONVOLUZIONE

```
function X=convoluzione(A,B)

la=length(A);
lb=length(B);
X=zeros(1,la+lb-1); %lunghezza del segnale risultante

B_rib=fliplr(B); %ribalta il segnale B

for k=1:(la+lb-1)
    tmp=[zeros(1,k-1),B_rib,zeros(1,la+lb-k-1)];
    X(k)=sum(A.*tmp(lb:lb+la-1));
end
end
```

## INTERVALLO

```
function x=intervallo(X,dx)
%produce un vettore da -X a X con passo dx
%X=valore assoluto che rappresenta gli estremi
%dell'intervallo simmetrico

%controllo input
if (X<=0 || dx<=0)
    x=0;
    printf ('errore vettore intervallo');
    return;
end
%produco vettore
n=ceil(X/dx);
x=(-n:n)';
x=x*dx;
end
```

## DELTA

```
function y=delta(x)

dt=x(2)-x(1);
y=zeros(size(x));
[minimo,indice]=min(abs(x));
y(indice)=1/dt;
end
```

## RECT

```
function y=rect(x)
%produce un segnale rettangolare
%x=intervallo su cui è definita la funzione

y=(abs(x)<.5)+.5*(abs(x)==.0);

end
```

## TRI

```
function y=tri(x)
%produco un segnale triangolare
%x=intervallo in cui è definita la funzione

y=1-abs(x);
y=y.*(abs(x)<1);
end
```

## SIG\_FFT

```
function [X,f]=sig_fft(x,t)
%costanti per i calcoli
N=length(x); %lunghezza di x
dT=t(2)-t(1); % tempo di campionamento
F=1/dT; %frequenza di campionamento
dF=F/N; %distanza tra i campioni in frequenza
M=floor(N/2);

%calcolo DFT
X=fftshift(fft(x))*dT;
%nel calcolo della DFT considerare la moltiplicazione
%per il fattore di scalatura dT

%trovo frequenze: l'intervallo delle frequenze deve avere
%lo stesso numero di elementi di X
f=(-M:M)'*dF; %arrotondo all'ingresso più vicino verso
sinistra
end
```

## TRENO

```
function [x]=treno(t,T)

%definisco costanti
dt=t(2)-t(1);
%controllo se T è coerente con l'intervallo
if (dt>T)
    fprintf('\nerroe! scegliere un T più grande\n');
end

tmax=max(t);
tmin=min(t); %estremi dell'intervallo temporale
x=zeros(size(t)); %inizializzo il vettore finale

k=0;

%condizione da verificare per l'inserimento degli impulsi
%nelle posizioni multiple di T
while k*t<tmax

x=x+delta(t-k*T);
k=k+1;
end

%creo il treno di impulsi nel semi intervallo negativo
k=-1;

while k*T>=tmin
    x=x+delta(t-k*T);
    k=k-1;
end
end
```