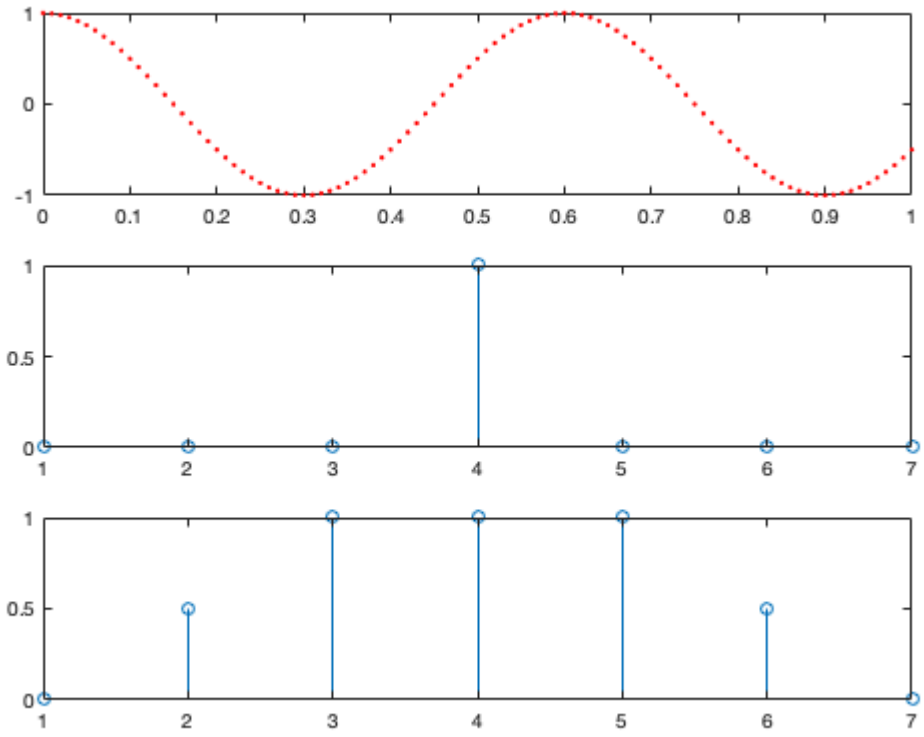
Convoluzione

Convoluzione 1-D

Definiamo due segnali monodimensionali *x* ed *h* da convolvere.

```
t=0:0.01:1;                             % tempo
omega=(2*pi)/0.6;                       % pulsazione
x=cos(omega*t);                         % cosinusoide

hImpulse=[0 0 0 1 0 0 0];               % impulso
hRect=[0 0.5 1 1 1 0.5 0];              % rect

figure;
subplot(3,1,1)
plot(t,x,'r.')
subplot(3,1,2)
stem(hImpulse)
subplot(3,1,3)
stem(hRect)
```



Il calcolo della convoluzione nel discreto è dato semplicemente dalla formula:

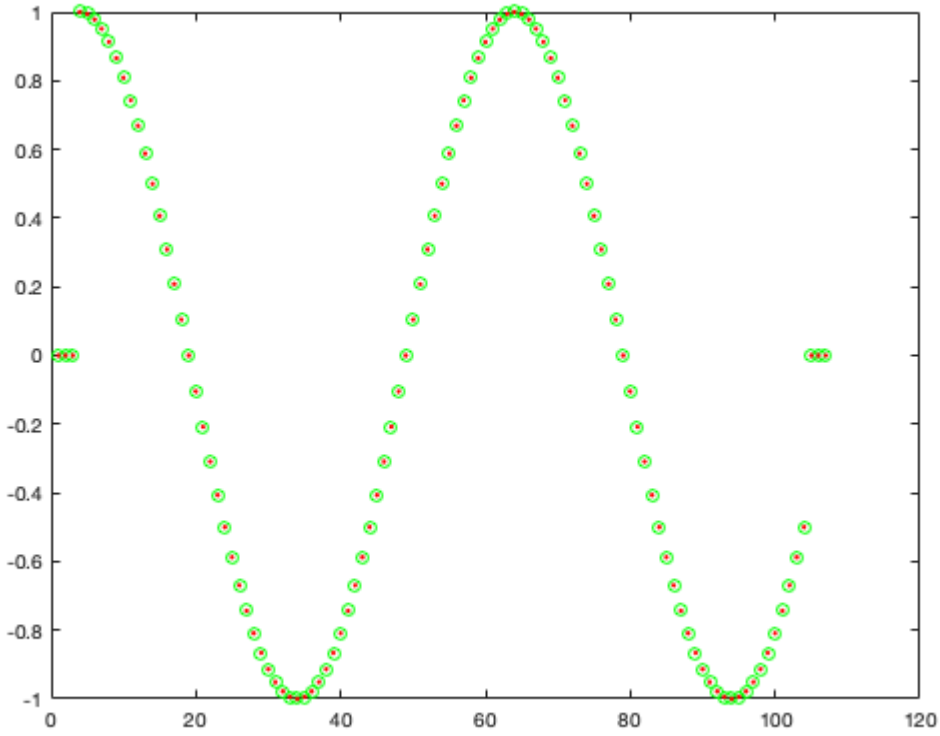
$$w(k) = \sum_{j}^{L} u(j)v(k-j+1)$$

Calcoliamo la convoluzione di *x* con *hImpulse*, implementando manualmente l'operazione attraverso la sua definizione:

```
% User defined convolution

N=length(x)+2*(length(hImpulse)-1);
yImpulse=zeros(1,N);                    % preallocazione output

padSize=(length(hImpulse)-1);           % padding
pad=zeros(1,padSize);
xPad=[pad x pad];

for i=1:N-padSize
yImpulse(i)=sum(xPad(i:i+padSize).*hImpulse);
end

yImpulse(end-padSize+1:end)=[];         % dimensione risposta n+m-1
```

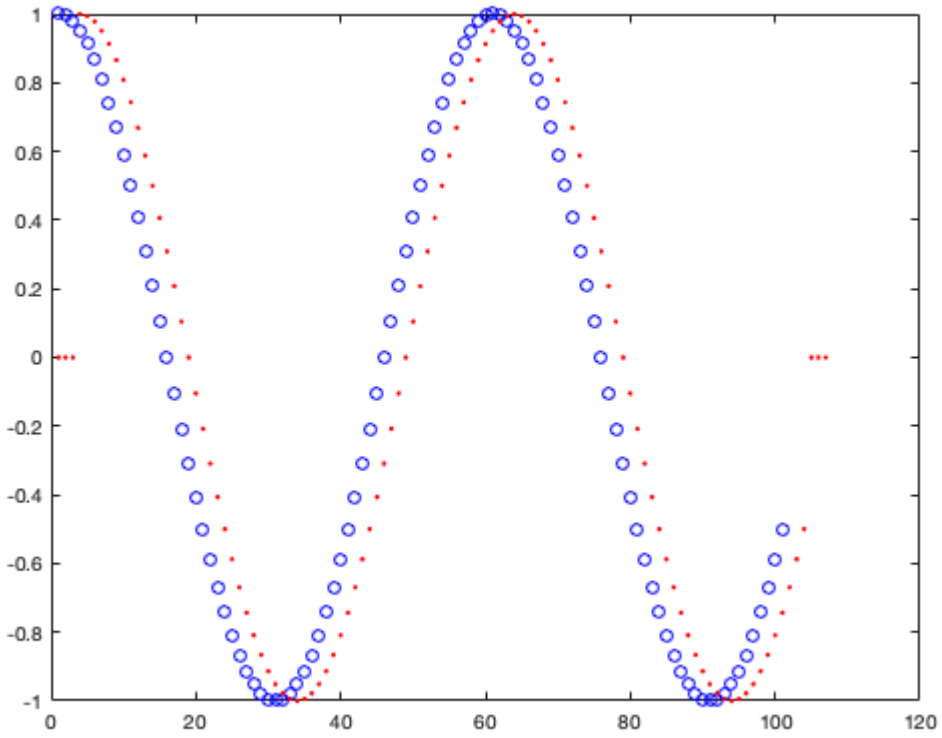
Confrontiamo il risultato della nostra convoluzione con quello della versione built-in MATLAB:

```
yImpulseMat=conv(x,hImpulse,'full');

figure
plot(yImpulse,'og'); hold on; plot(yImpulseMat,'.r');
```



Confrontiamo il risultato della convoluzione di *x* con *hImpulse*:

```
figure
plot(x,'ob'); hold on; plot(yImpulseMat,'.r')
```

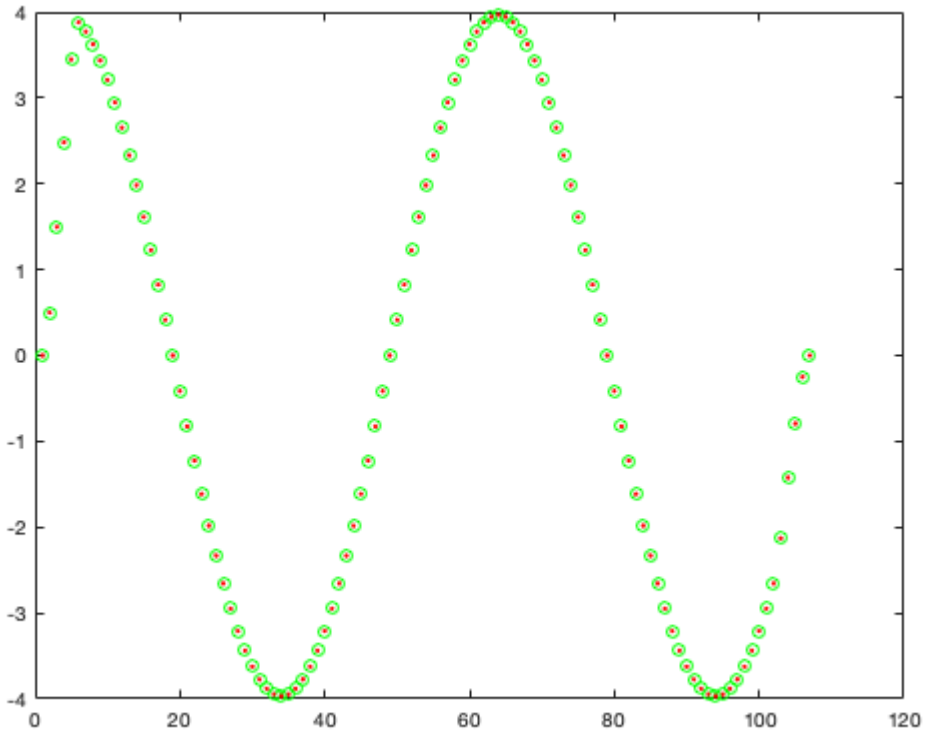


Ripetiamo quanto sopra per hRect:

```
% User defined convolution

N=length(x)+2*(length(hRect)-1);
yRect=zeros(1,N);                          % preallocazione output

padSize=(length(hRect)-1);                 % padding
pad=zeros(1,padSize);
xPad=[pad x pad];

for i=1:N-padSize
yRect(i)=sum(xPad(i:i+padSize).*hRect);
end

yRect(end-padSize+1:end)=[];               % dimensione risposta n+m-1
```

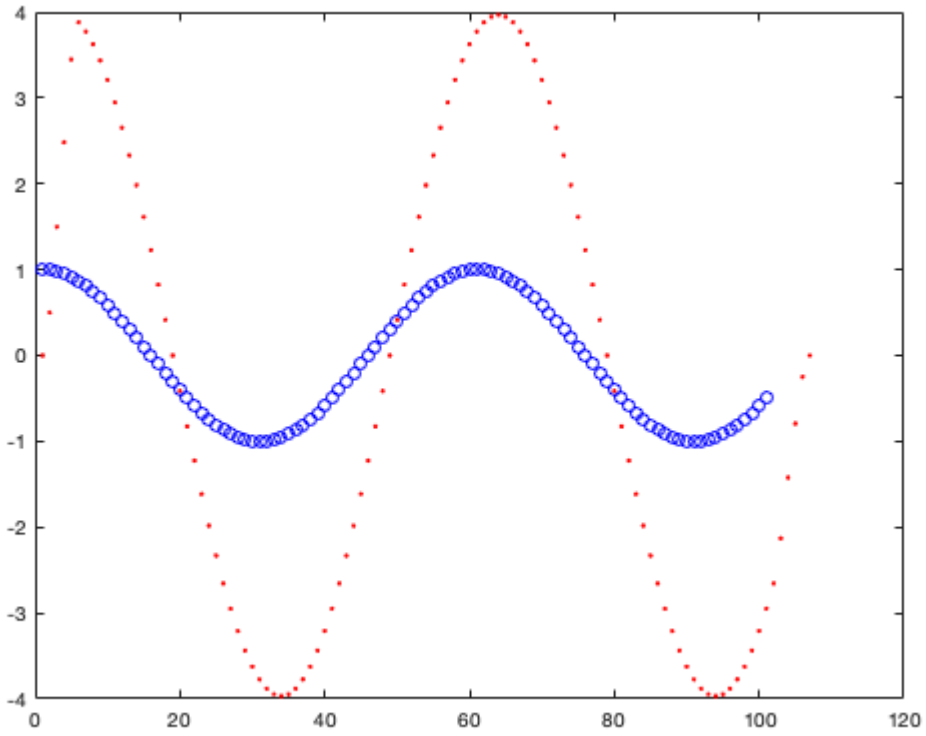
Confrontiamo ancora il risultato della nostra convoluzione con quello della versione built-in MATLAB:

```
yRectMat=conv(x,hRect,'full');

figure
plot(yRect,'og'); hold on; plot(yRectMat,'.r');
```



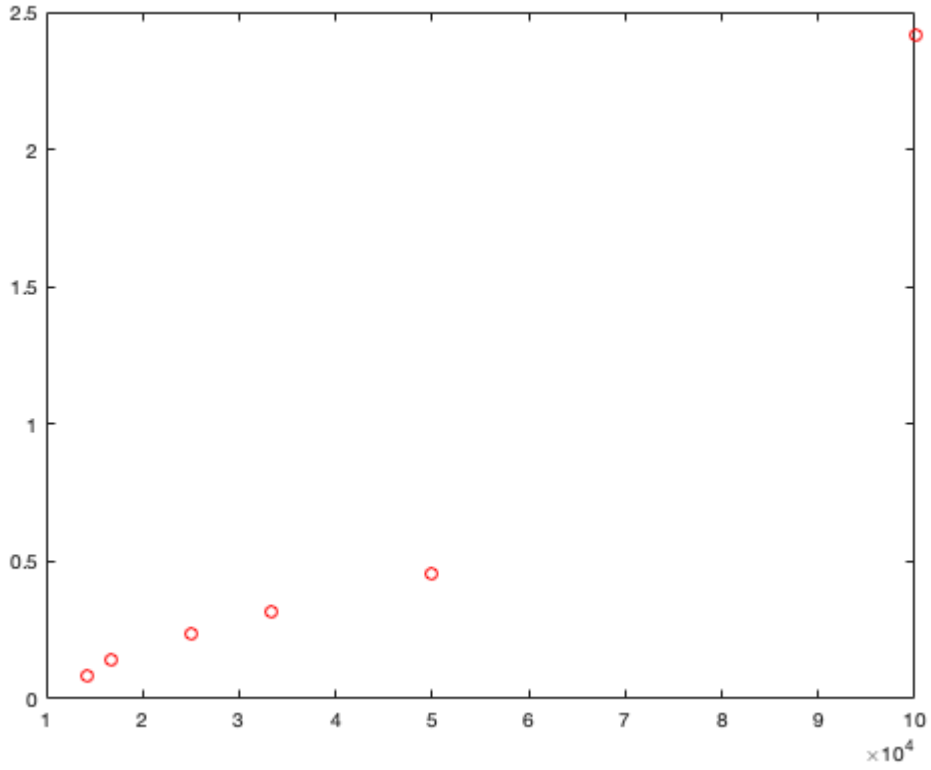
Confrontiamo il risultato della convoluzione di *x* con *hRect*:

```
figure;
plot(x,'ob'); hold on; plot(yRectMat,'.r')
```



[Esperimento] Stima costo computazionale della convoluzione in funzione del numero di campioni:

```
delta=[1e-5 2e-5 3e-5 4e-5 6e-5 7e-5];
numberOfElements=1./delta;

dt=zeros(size(delta));

for i=1:size(delta,2)
    t=0:delta(i):1;
    x=cos(omega*t);
    y=sin(omega*t);
    tic
    yRect=conv(x,x);
    dt(i)=toc;
end

figure; plot(numberOfElements,dt,'or')
```

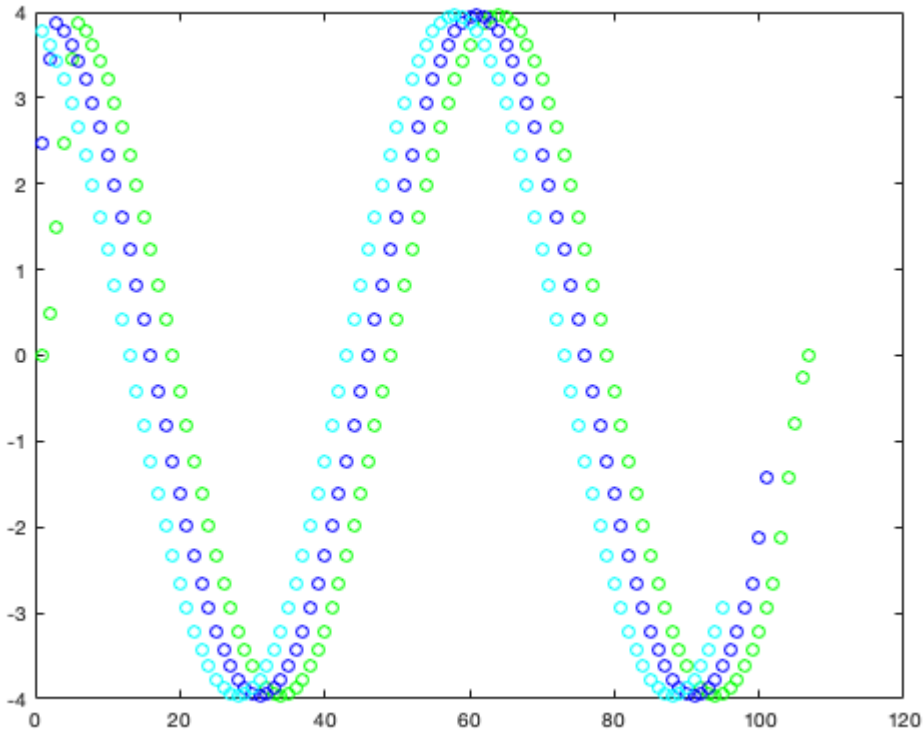


Padding

La funzione built in di MATLAB offre tre possibilità diverse di padding: *full, valid e same:*

- *full* restituisce l'intero segnale in uscita (implementazione di default)
- *same* restituisce solo i valori in uscita relativi ai campioni del primo segnale in ingresso (no padding segnale in uscita)
- *valid* restituisce i soli campioni per cui la convoluzione è stata calcolata interamente sui campioni del segnale in ingresso (no padding segnale in ingresso)

```
t=0:0.01:1;                             % tempo
omega=(2*pi)/0.6;                       % pulsazione
x=cos(omega*t);

yRectMatFull=conv(x,hRect);
yRectMatSame=conv(x,hRect,'same');
yRectMatValid=conv(x,hRect,'valid');

figure
plot(yRectMat,'og'); hold on; plot(yRectMatSame,'ob'); hold on; plot(yRectMatValid,'oc
```


4
3
2
1
0
-1
-2
-3
-4
0
20
40
60
80
100
120