# Lab 3
# Pre-processing of ECG Signals: Notes

Meng-Lin Li (李夢麟), Ph. D.
mlli@ee.nthu.edu.tw
Department of Electrical Engineering
National Tsing Hua University

# 1. Order Selection of a FIR filter and Usage of fir1()

## fir1   FIR filter design using the window method.

**B = fir1(N,Wn) designs an N'th order lowpass FIR digital filter and returns the filter coefficients in length N+1 vector B.**

**(impulse response)**

**The cut-off frequency Wn must be between 0 < Wn < 1.0, with 1.0 corresponding to half the sample rate.**

**B = fir1(N,Wn,'high') designs an N'th order highpass filter.**
**You can also use B = fir1(N,Wn,'low') to design a lowpass filter.**

**Check frequency response via MATLAB freqz()**
**MATLAB filter()?**

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$$

$$= y_{ss}[n] + y_{tr}[n]$$

$$= y_{zi}[n] + y_{zs}[n]$$

$$\boxed{\begin{cases} y_{ss}[n]: \text{steady state response} & y_{ss}[n] = \lim_{n\to\infty} y[n] \\ y_{tr}[n]: \text{transient response} & y_{tr}[n] = y[n] - y_{ss}[n] \end{cases}}$$

$y_{zi}[n]$ : zero input response : the response due to initial conditions alone w/ the input set to 0. (i.e., $x[n]=0$)

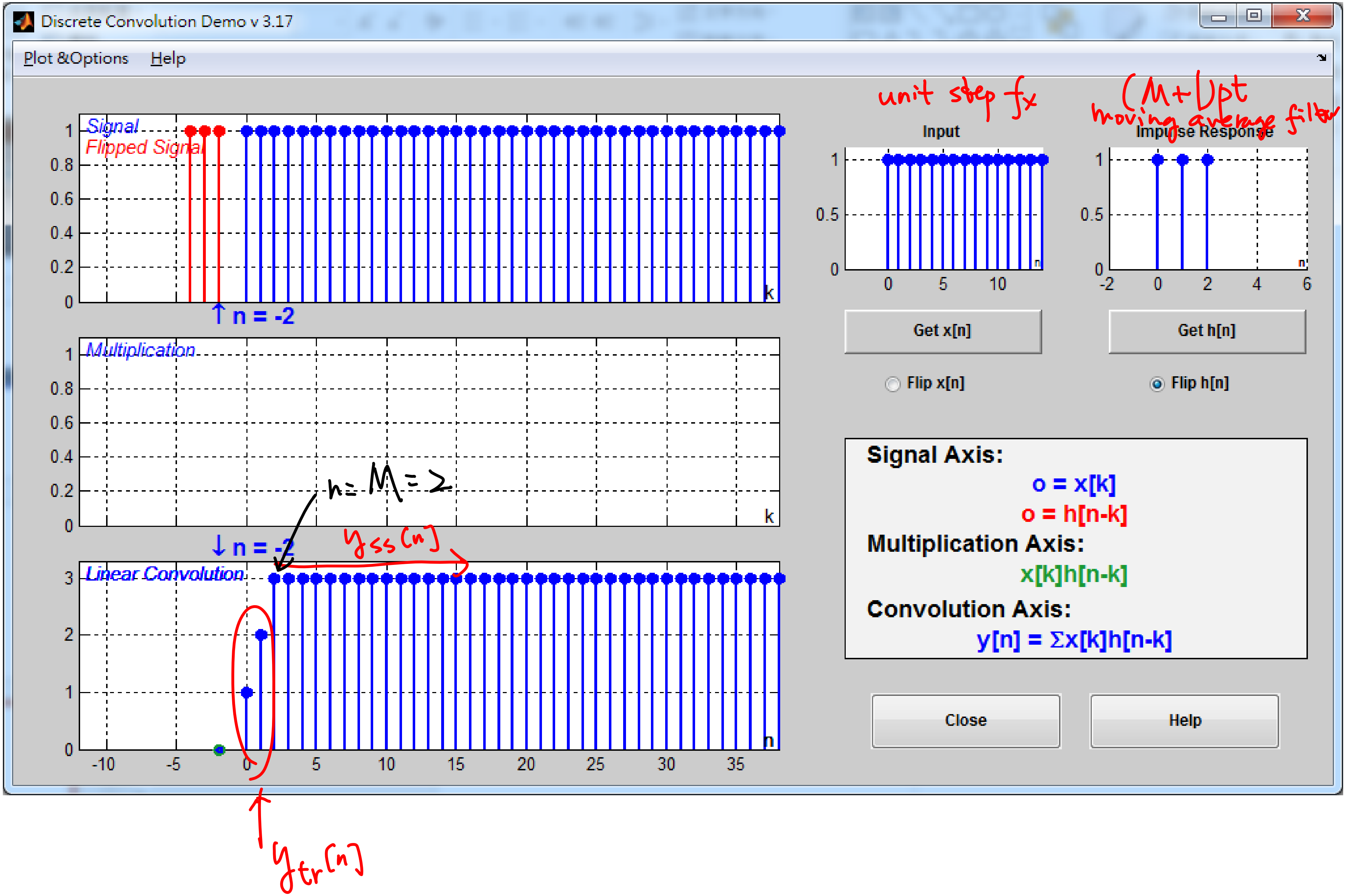
$y_{zs}[n]$ : zero state response : the response due to the system input w/ initial conditions set to 0

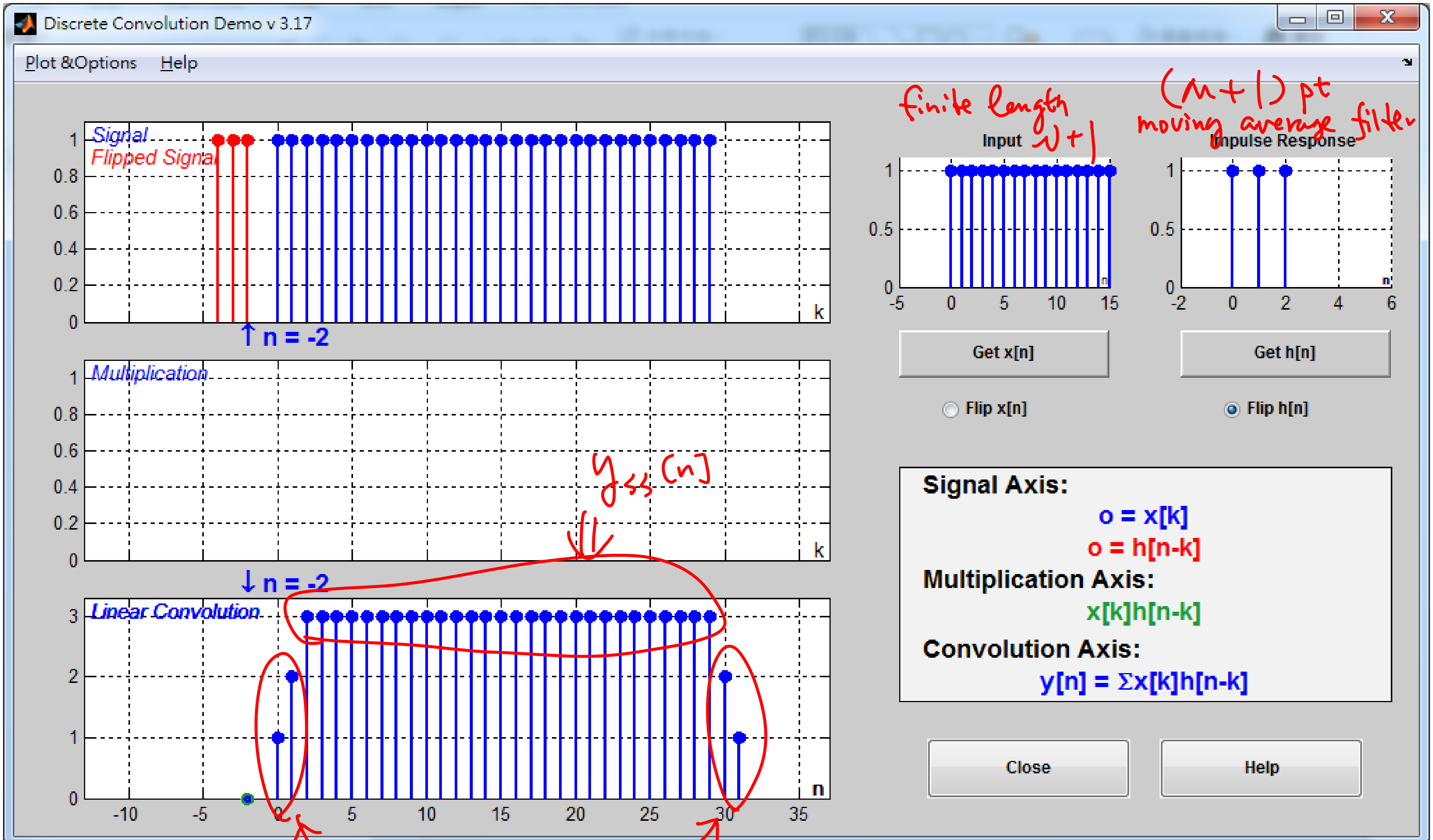
i.e., initially at rest, $y[n]=0$, $n<0$

(see textbook p62~p64)

# More about LTI FIR Systems, Convolution Sum, Transient Response and Steady State Response

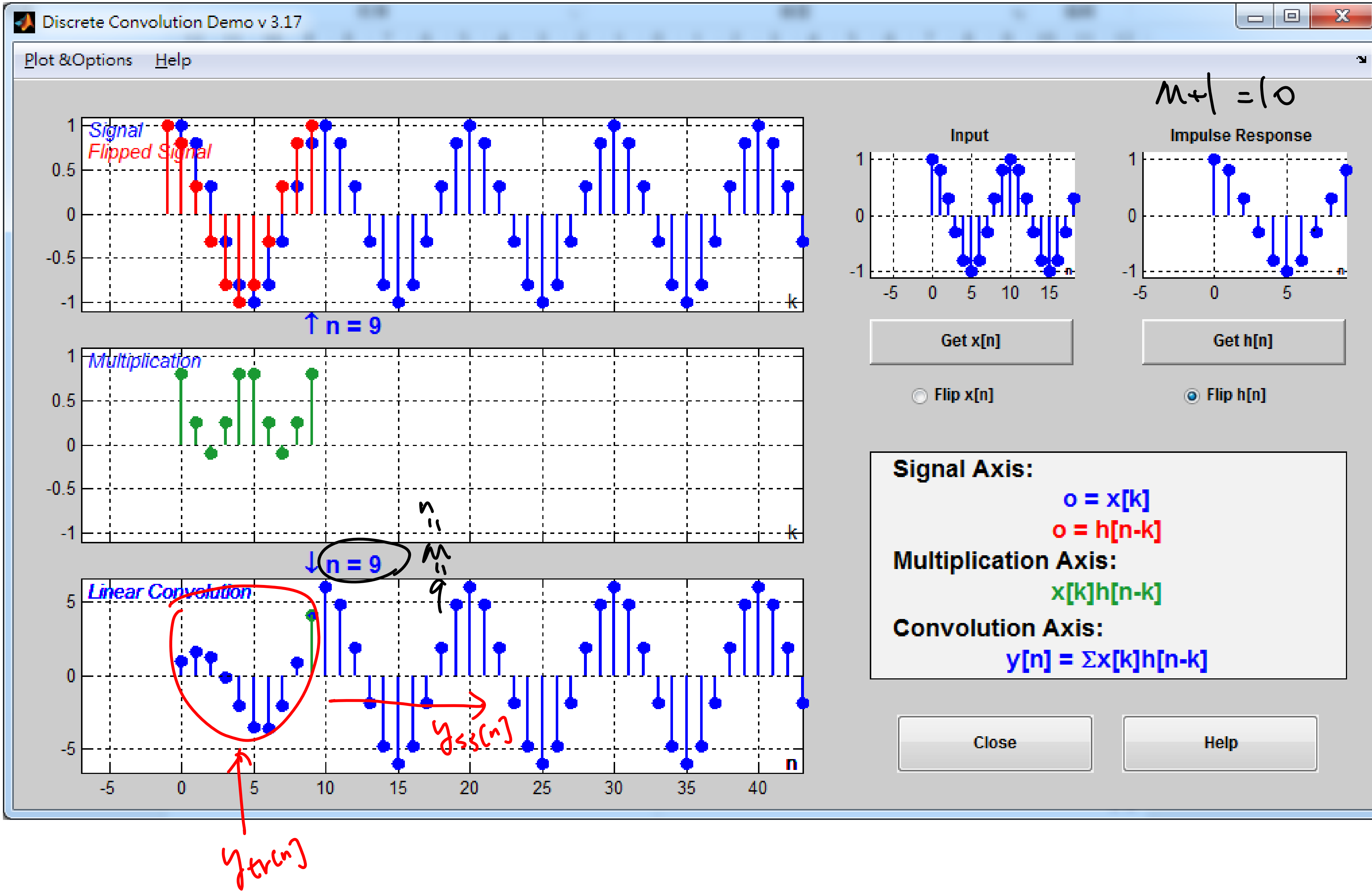
$$y_{ss}[n] = \lim_{n \to \infty} y[n]$$


Discrete Convolution Demo v 3.17
Plot &Options
Help
Signal
Flipped Signal
n = -2
Multiplication
Linear Convolution
unit step fx
(M+1)pt
moving average filter
Input
Impulse Response
Get x[n]
Get h[n]
Flip x[n]
Flip h[n]
Signal Axis:
o = x[k]
o = h[n-k]
Multiplication Axis:
x[k]h[n-k]
Convolution Axis:
y[n] = Σx[k]h[n-k]
Close
Help
n = M = 2
y_ss(n)
y_tr(n)



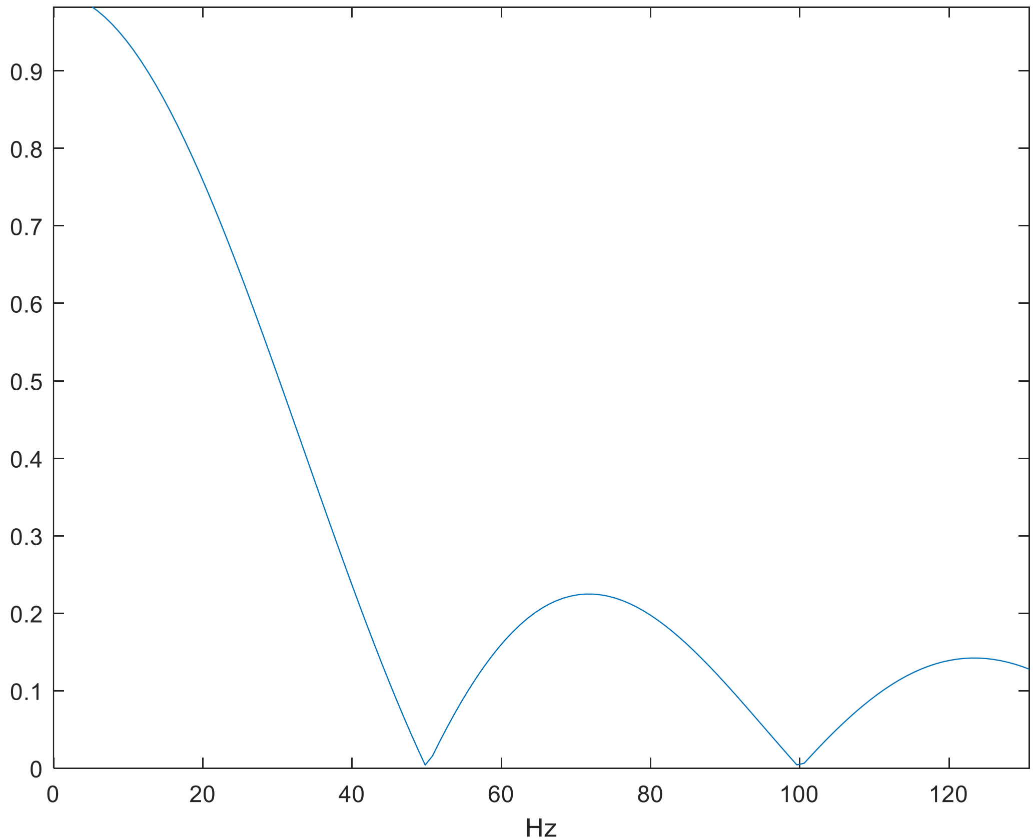

$y_{tr}[n]$

what if $(N+1) < (M+1)$?

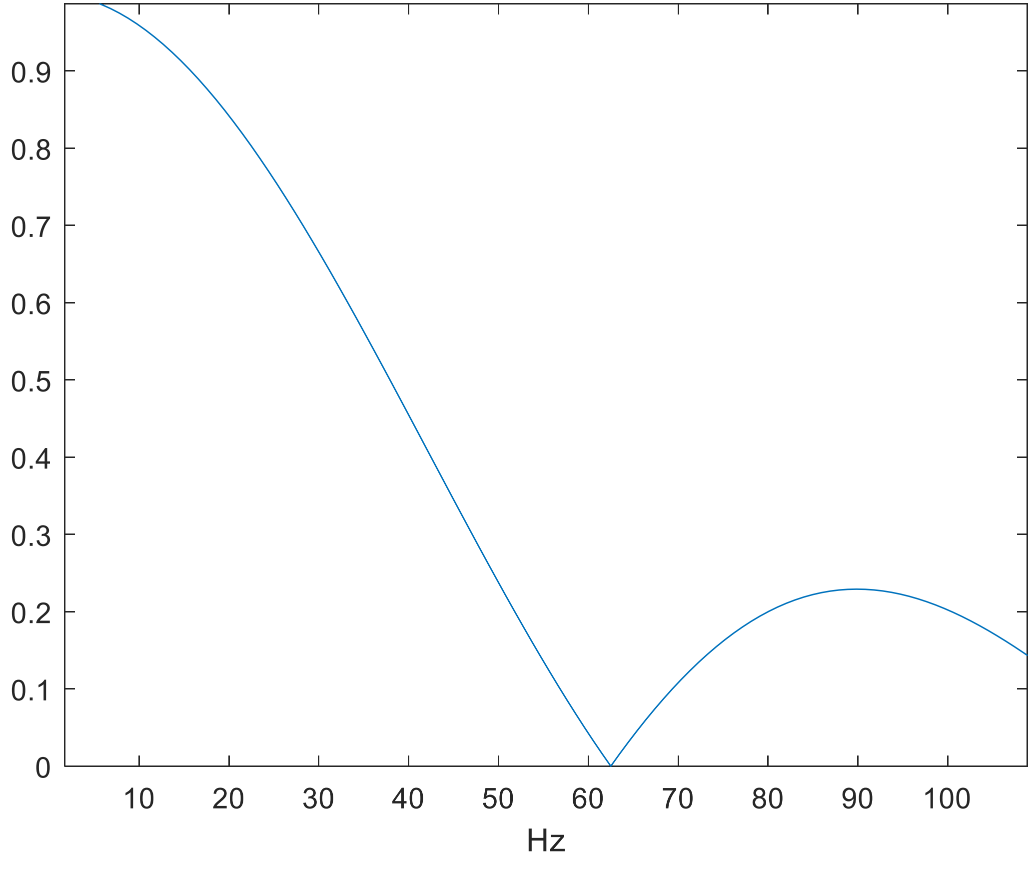

Discrete Convolution Demo v 3.17
Plot &Options
Help
M+1 = 10
Signal
Flipped Signal
↑ n = 9
Multiplication
n = M = 9
↓ n = 9
Linear Convolution
$y_{ss}[n]$
$y_{tr}[n]$
Input
Impulse Response
Get x[n]
Get h[n]
Flip x[n]
Flip h[n]
Signal Axis:
o = x[k]
o = h[n-k]
Multiplication Axis:
x[k]h[n-k]
Convolution Axis:
y[n] = Σx[k]h[n-k]
Close
Help

# 2. Moving Average Filtering and Difference Filtering

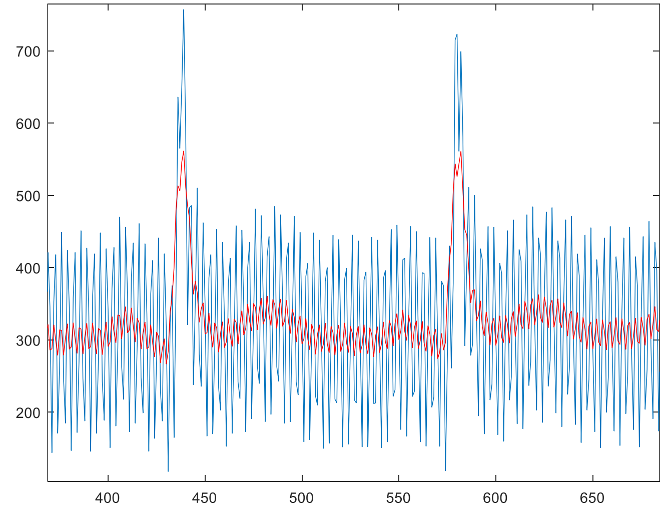
```
>> Fs = 500; % in Hz
>> ma = ones(1,10)/10;
>> figure
>> plot((0:511)*Fs/512, abs(fft(ma,512)));
>> xlabel('Hz')
```



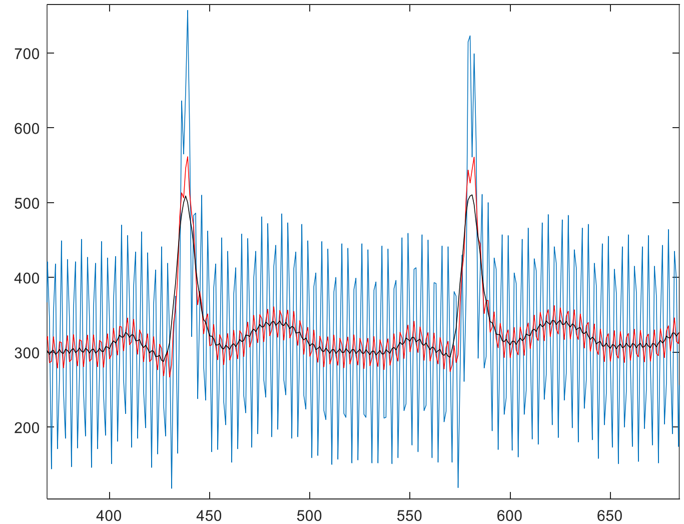
```
>> Fs = 500; % in Hz
>> ma = ones(1,8)/8;
>> figure
>> plot((0:511)*Fs/512, abs(fft(ma,512)));
>> xlabel('Hz')
```



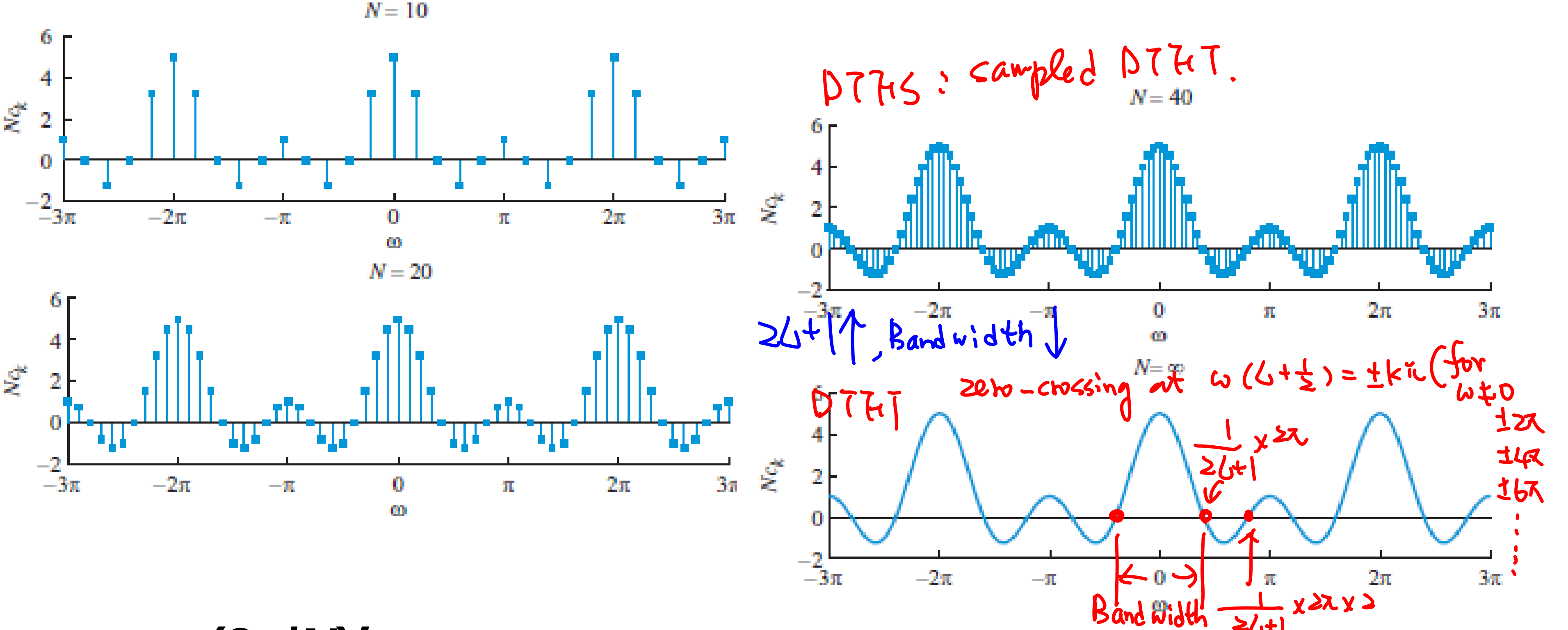
```
ECG_notchfiltered = conv(raw_ECG,ma,'same');
```




```
ECG_notchfiltered = conv(ECG_notchfiltered ,ma,'same'); % casecaded system
```



# Example 4.8




$\omega_k = (2\pi/N)k$

Figure 4.22 How the DTFS converges to the DTFT as the period $N$ of a fixed-width ($2L + 1 = 5$ samples) rectangular pulse tends to infinity.

from(4.19) $\Rightarrow X(e^{j\omega}) = \begin{cases} 2L+1, & \omega = 0, \pm 2\pi, \pm 4\pi \cdots \\ \dfrac{\sin \omega(L+\frac{1}{2})}{\sin \omega/2}, & \text{otherwise} \end{cases}$