Uncle Steve's

Guide To
Digital Signal Processing

With essential help from Ali Shoeb and Dan Lovell, former 6.115 Students!

Digital Signal Processing

- A satellite has just captured the following image of the moon and is ready to send it to an earth station.


Original Image

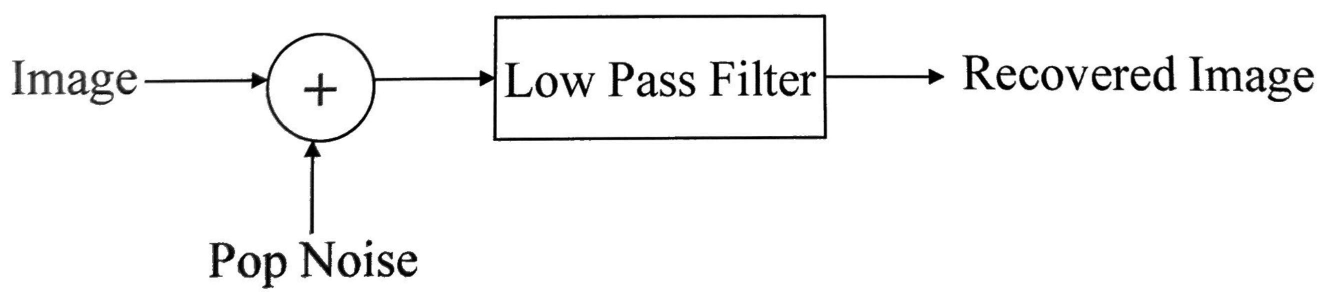
Digital Signal Processing

- Pop-Noise corrupts the the image of the moon as it is transmitted to earth.

Received Image

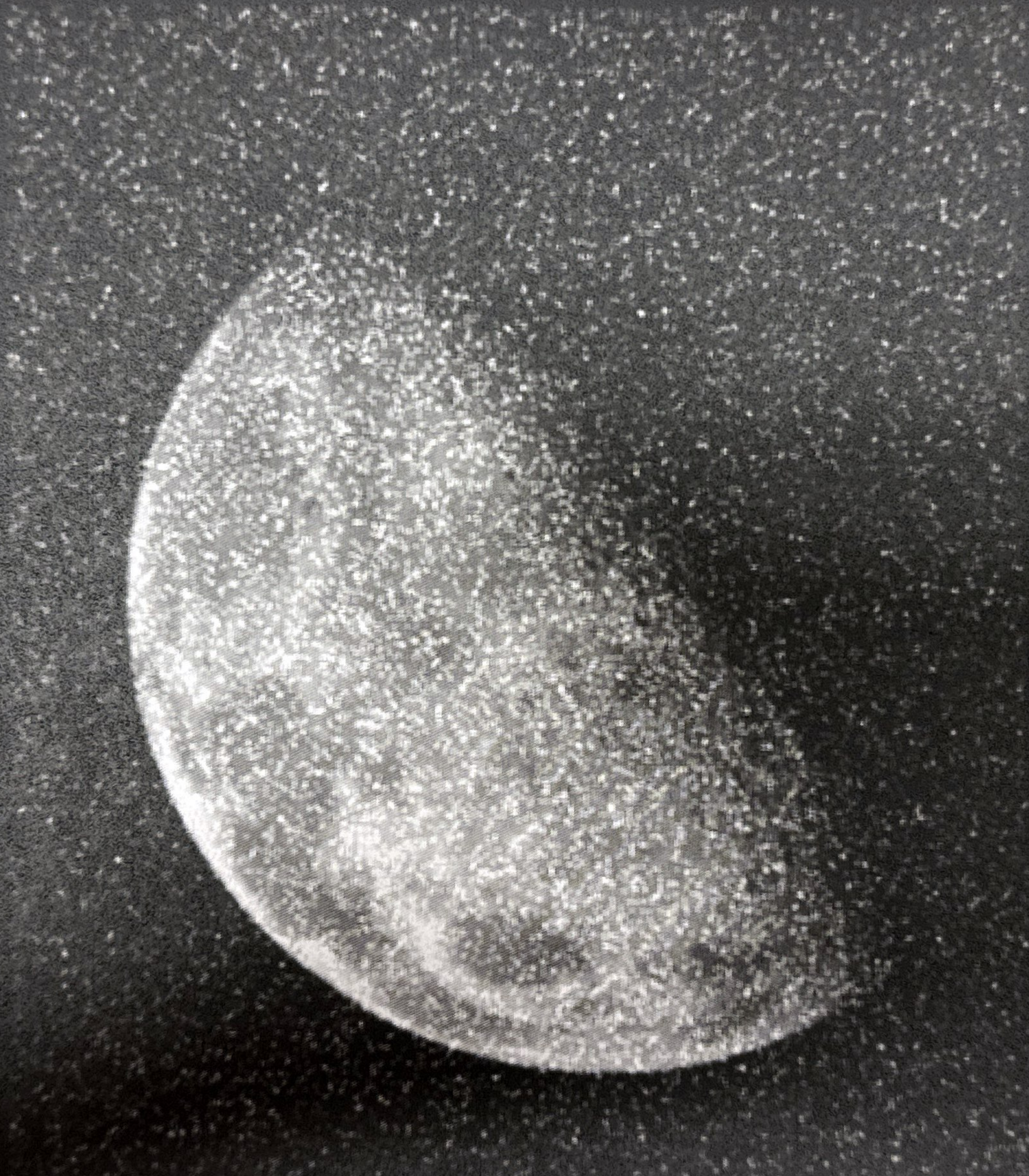
Digital Signal Processing

- High frequencies in images are fast transitions in pixel intensities: black → white or white → black in gray scale images.
- Pop-Noise introduces high frequency artifacts into an image.
- We can reduce high frequency by using a low pass filter.



Received Image

Low Pass Filter



Digital Signal Processing

- The low pass filter blurred the image of the moon and did not remove all of the Pop-Noise.

- Pop Noise corrupts a pixel in a neighborhood of pixels by changing its intensity to an extreme (black or white). This makes it easy to identify a pixel corrupted by Pop Noise from among a group of pixels.

- A Median Filter assigns each pixel in an image the median intensity of pixels in its neighborhood. The size of a pixel's neighborhood is specified by the user.

8-Pixel neighborhood surrounding pixel corrupted by Pop Noise.

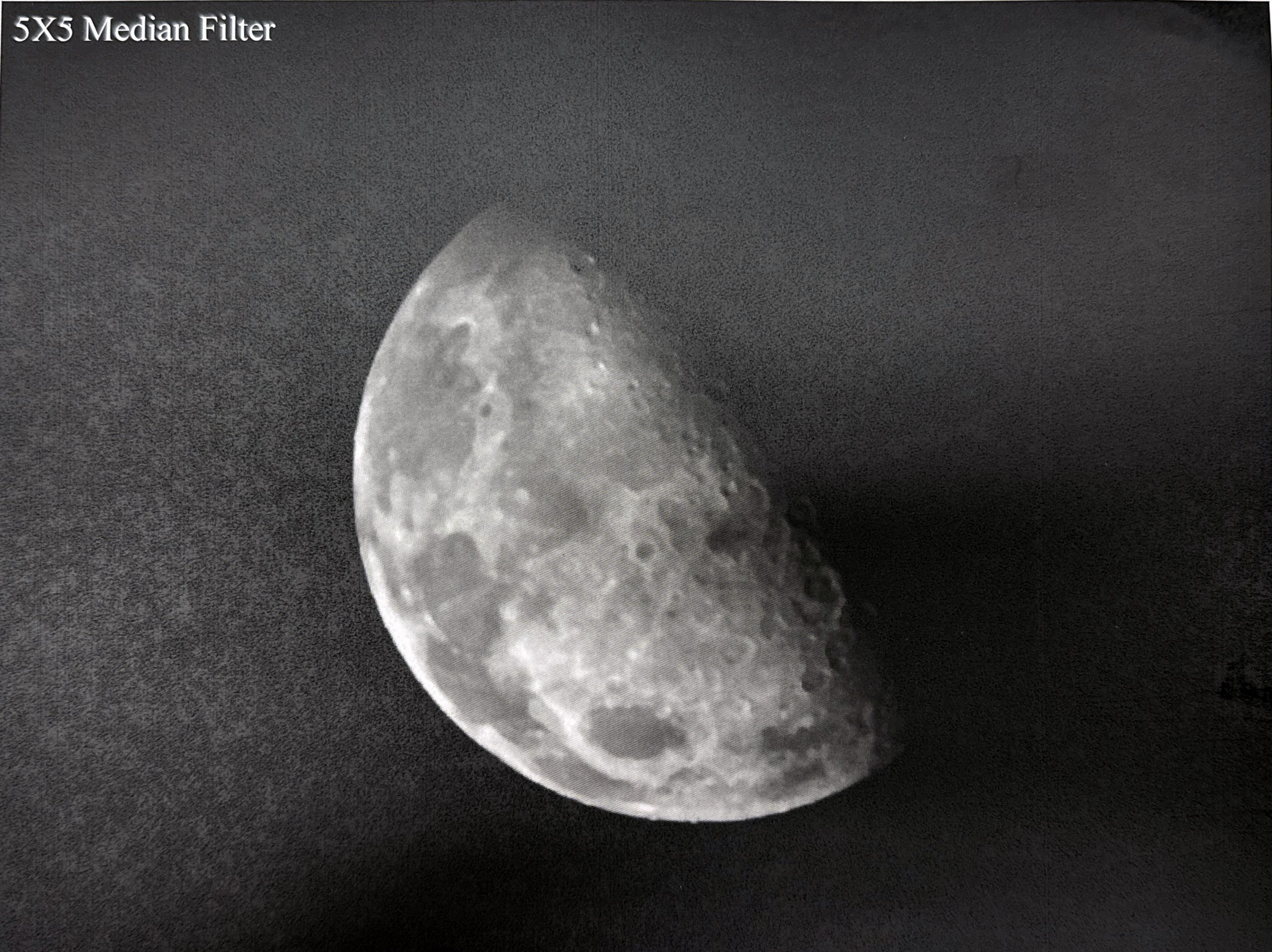
0	0	0
0	1	0
0	0	0

Pop Noise changed this value from a 0 → 1

Median Filter will assign this pixel a value of 0 since the median pixel intensity in neighborhood is zero.

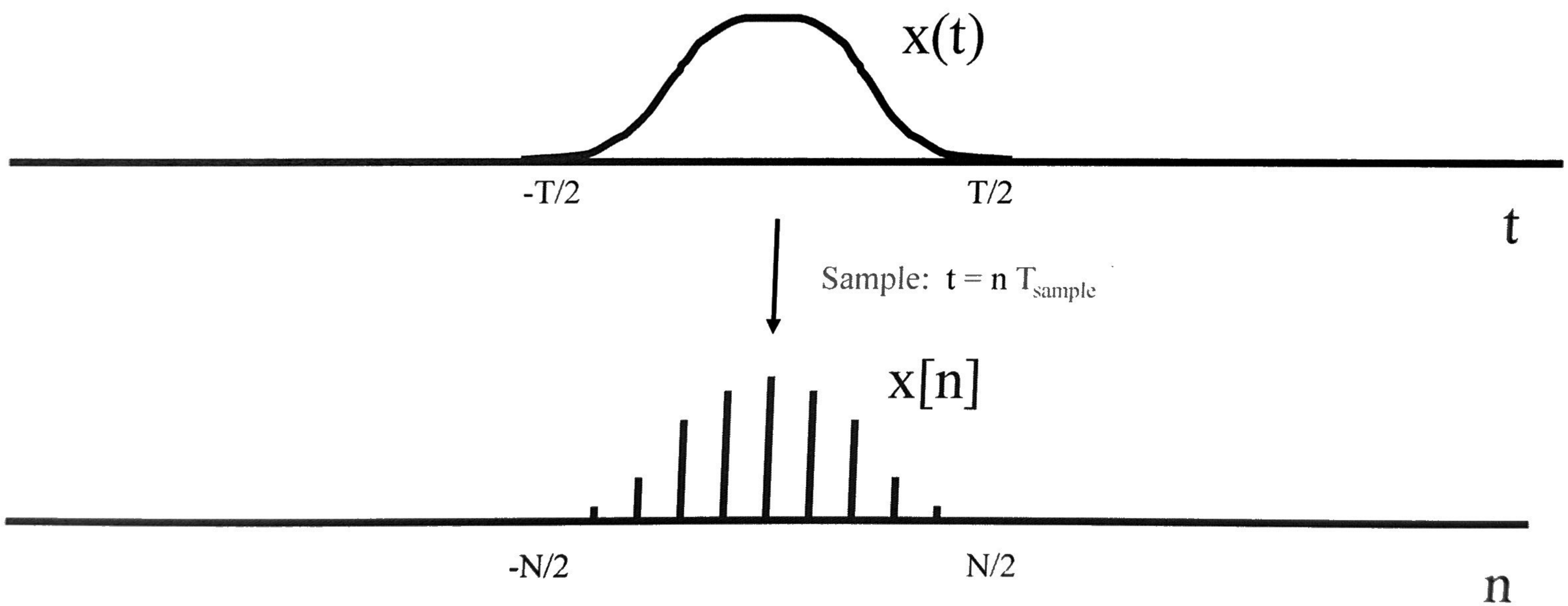
Received Image

2X2 Median Filter


5X5 Median Filter

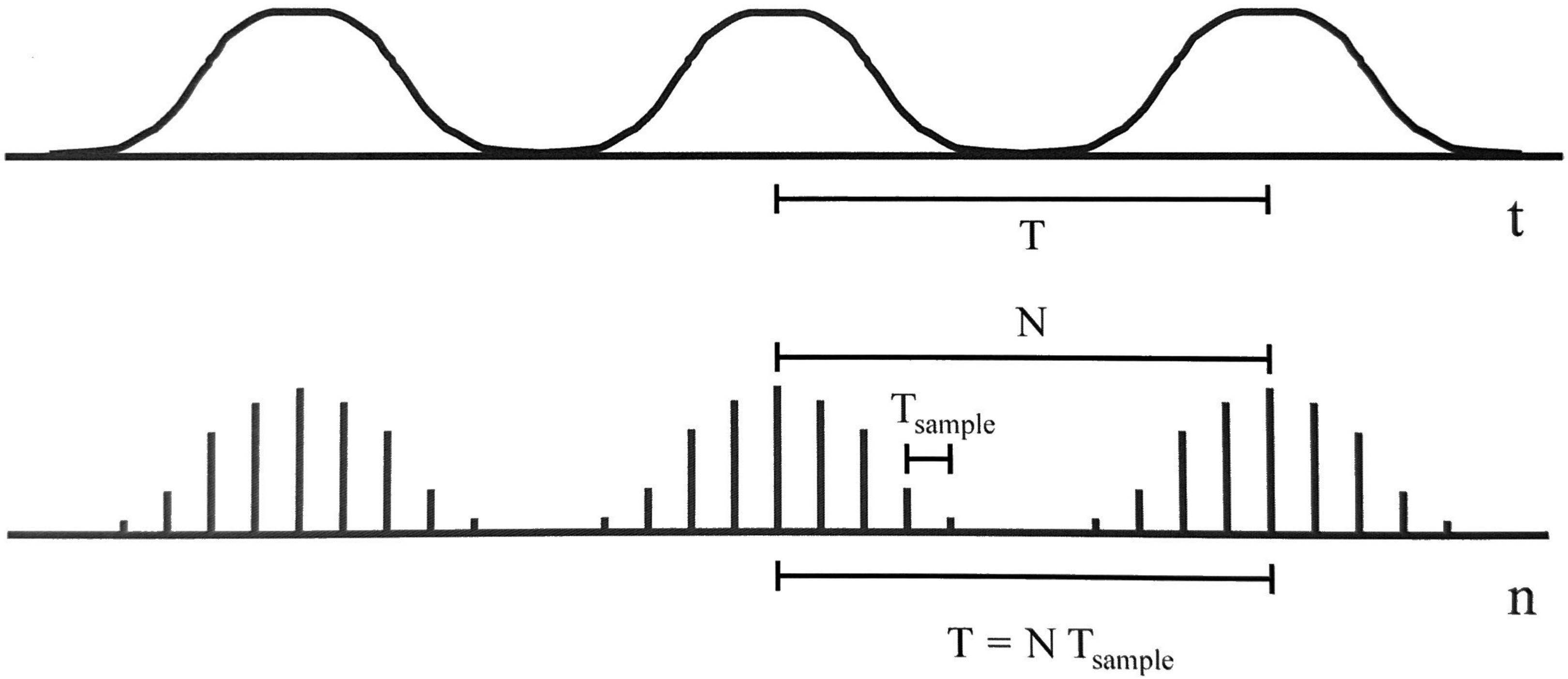
DT Signals are Samples of CT Signals

- We know how to construct a finite length discrete signal x[n] by sampling a continuous signal x(t) at the Nyquist Rate.



Periodic DT Signals Are Samples of Periodic CT Signals

- We can replicate the discrete signal x[n] and think of it as samples of replicas of the continuous periodic signal x(t).



Representing Continuous Periodic Signals

- Jean Fourier showed that any continuous periodic signal can be represented as infinite sum of harmonically related sinusoids. This representation tells about the frequency content of the signal.

$$x(t) = a_0 + \sum_{k=1}^{\infty} \underbrace{\{\, B_k \cos(k\Omega_0 t) + C_k \sin(k\Omega_0 t) \,\}}_{\text{Weighted Sinusoids}} \qquad -\infty < t < \infty$$

$$\Omega_0 = 2\pi / T$$

Periodic Square Wave

Let us see if Fourier is right!

- Suppose x(t) is a square wave with a period T and Amplitude 1.

- Fourier claims

$$x(t) = (4/\pi) \sum_{n=1}^{\infty} \frac{\sin((2n-1)\Omega_0 t)}{(2n-1)}$$

$$\Omega_0 = 2\pi/T$$

- $x(t) = (4/\pi)\,[\,\sin(\Omega_0 t) + 1/3 \sin(3\,\Omega_0 t) + 1/5 \sin(5\,\Omega_0 t) \ldots]$

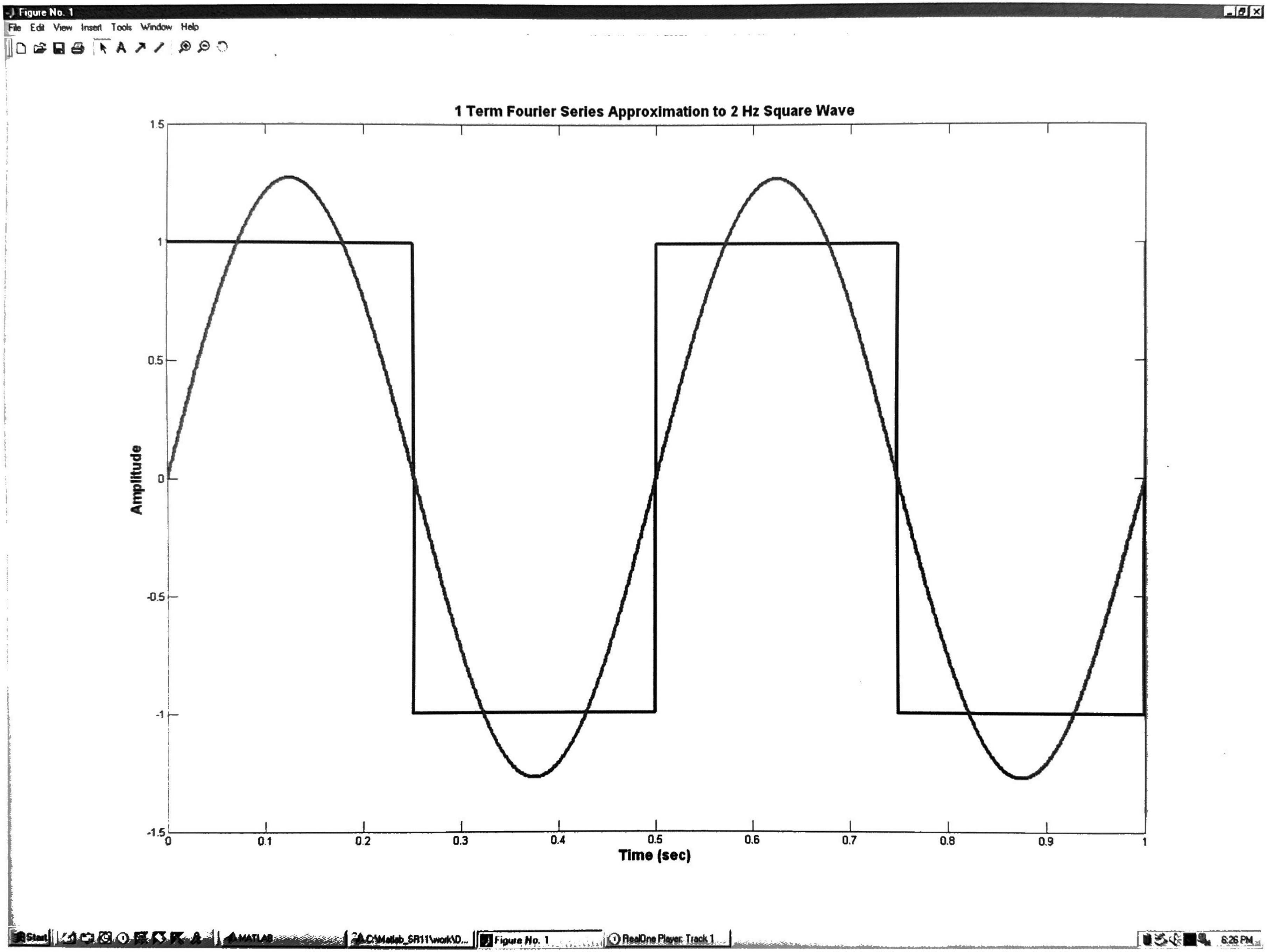


Figure No. 1
File Edit View Insert Tools Window Help
1 Term Fourier Series Approximation to 2 Hz Square Wave
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Start
MATLAB
Figure No. 1
RealOne Player: Track 1
6:26 PM

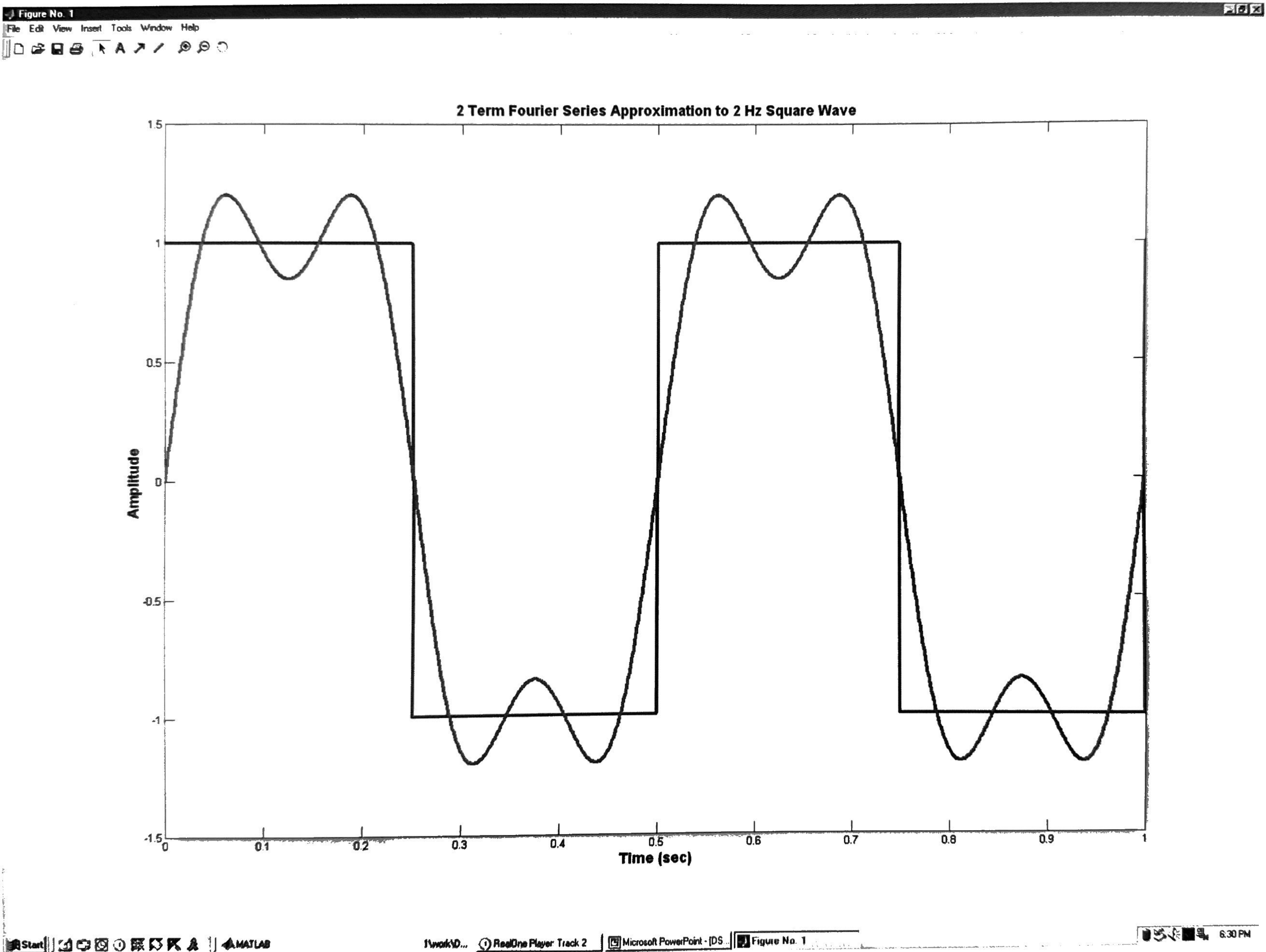
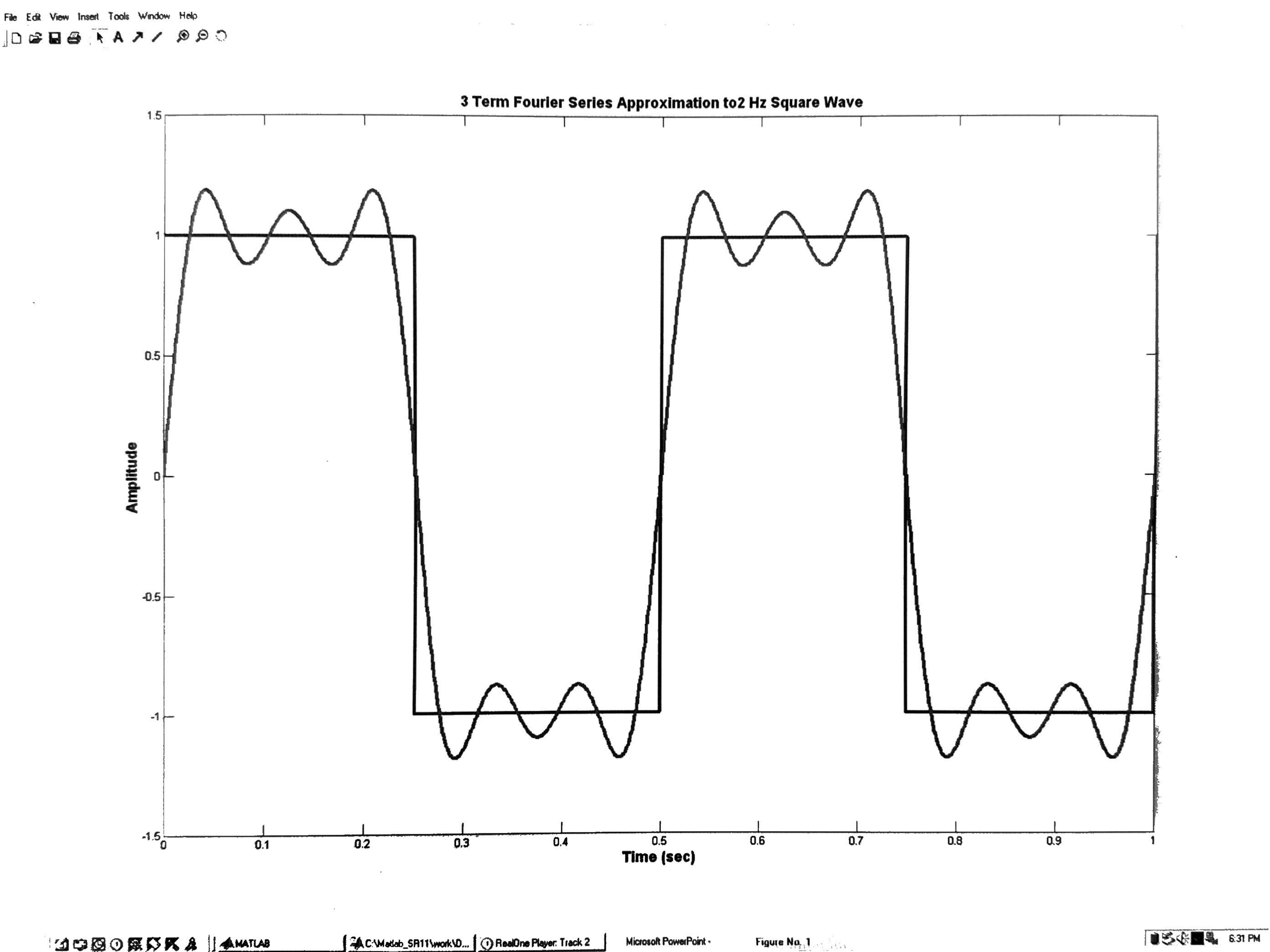
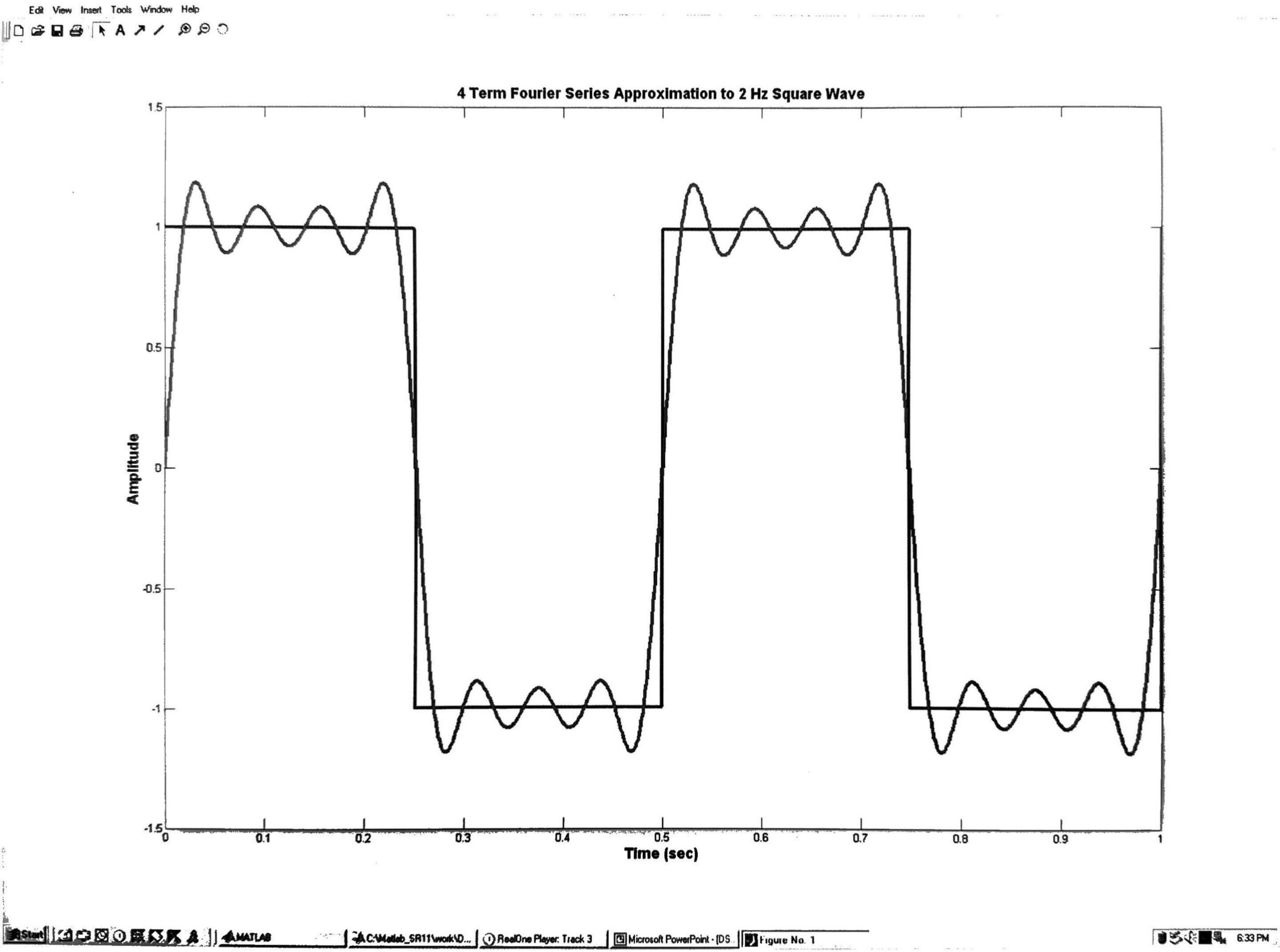
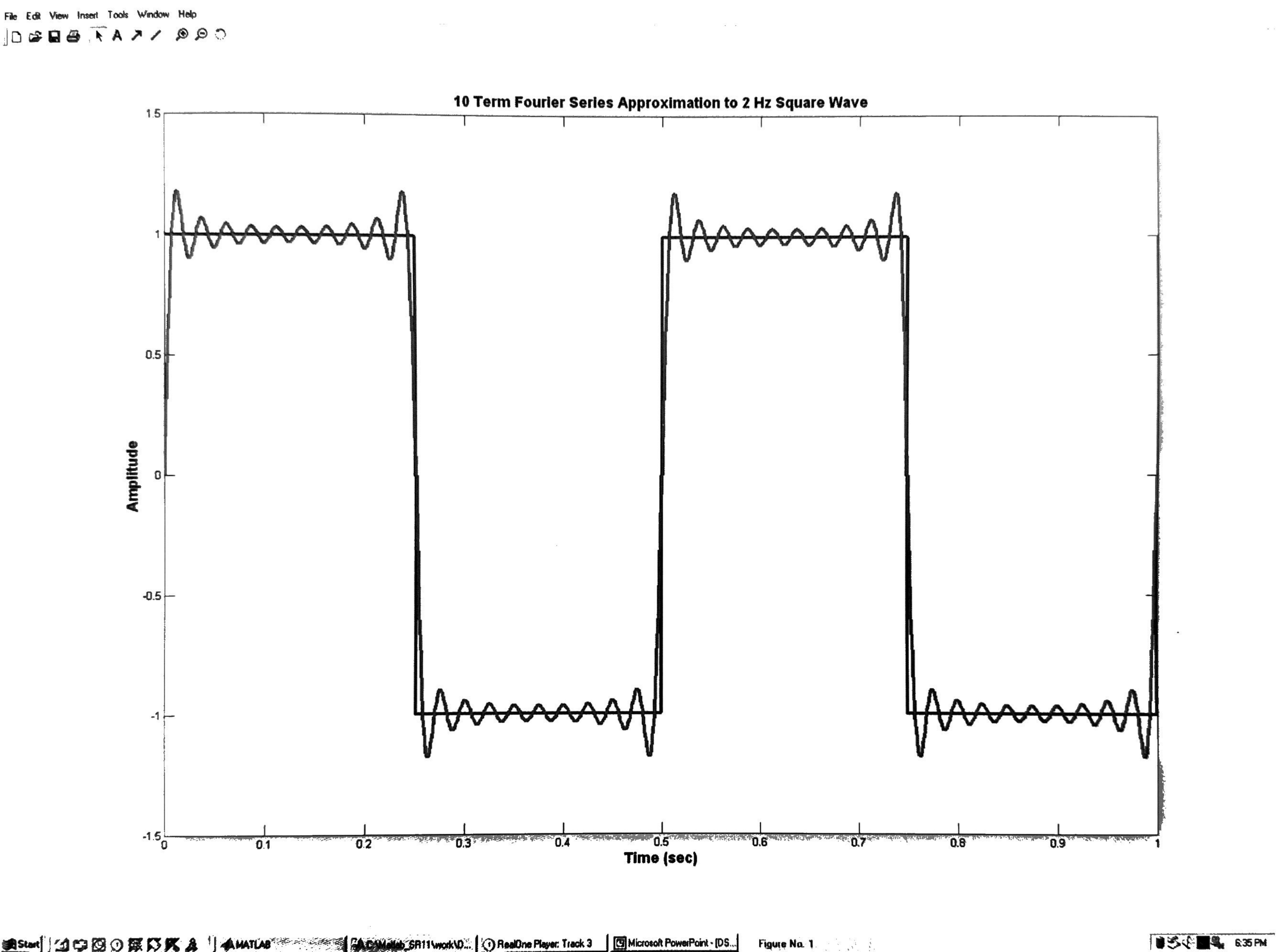


Figure No. 1
File Edit View Insert Tools Window Help
2 Term Fourier Series Approximation to 2 Hz Square Wave
1.5
1
0.5
0
-0.5
-1
-1.5
Amplitude
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (sec)
Start
MATLAB
RealOne Player: Track 2
Microsoft PowerPoint - [DS...
Figure No. 1
6:30 PM


File Edit View Insert Tools Window Help
3 Term Fourier Series Approximation to2 Hz Square Wave
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
MATLAB
C:\Matlab_SR11\work\D...
RealOne Player: Track 2
Microsoft PowerPoint -
Figure No. 1
6:31 PM



Edit View Insert Tools Window Help
4 Term Fourier Series Approximation to 2 Hz Square Wave
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Start
MATLAB
C:\Matlab_SR11\work\D...
RealOne Player: Track 3
Microsoft PowerPoint - [DS...
Figure No. 1
6:33 PM


File Edit View Insert Tools Window Help
10 Term Fourier Series Approximation to 2 Hz Square Wave
Amplitude
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (sec)
Start
MATLAB
RealOne Player: Track 3
Microsoft PowerPoint - [DS...
Figure No. 1
6:35 PM

Periodic Saw Tooth Wave

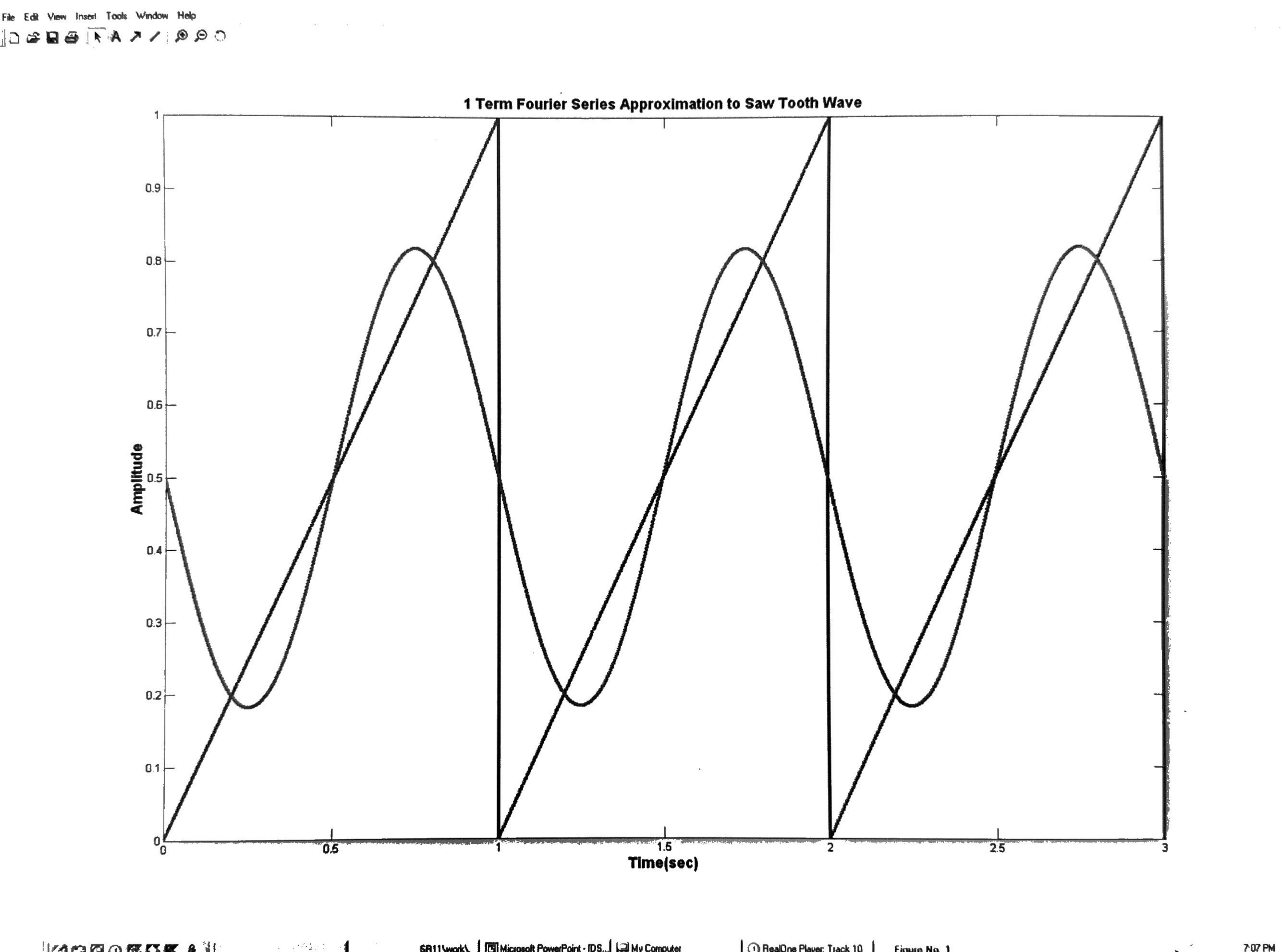
- Suppose x(t) is a saw tooth wave with a period T and Amplitude 1.

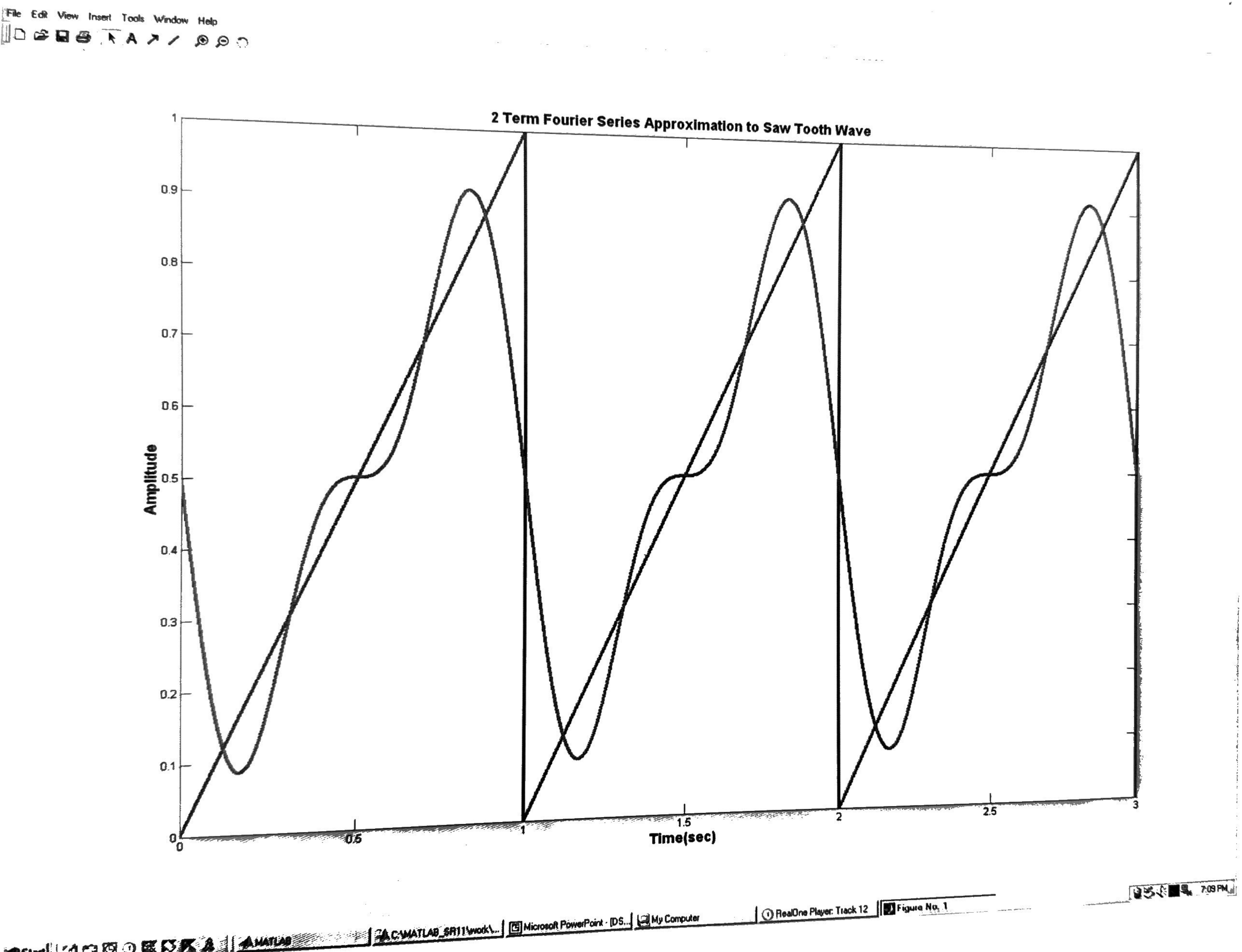
- Fourier claims

$$x(t) = (T/2) - \sum_{n=1}^{\infty} \frac{T}{\pi\, n} \sin(n\omega_0 t)$$

$$\Omega_0 = 2\pi/T$$

- $x(t) = (1/2) - [\, (T/\pi) \sin(\omega_0 t) + (T/2\pi) \sin(2\omega_0 t) \ldots\ldots]$


File Edit View Insert Tools Window Help
1 Term Fourier Series Approximation to Saw Tooth Wave
Amplitude
Time(sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.5
1
1.5
2
2.5
3
6R11\work\
Microsoft PowerPoint - [DS...
My Computer
RealOne Player: Track 10
Figure No. 1
7:07 PM


File Edit View Insert Tools Window Help
2 Term Fourier Series Approximation to Saw Tooth Wave
Amplitude
Time(sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1.5
2
2.5
3
Start
MATLAB
C:\MATLAB_SR11\work\...
Microsoft PowerPoint - [DS...
My Computer
RealOne Player: Track 12
Figure No. 1
7:09 PM

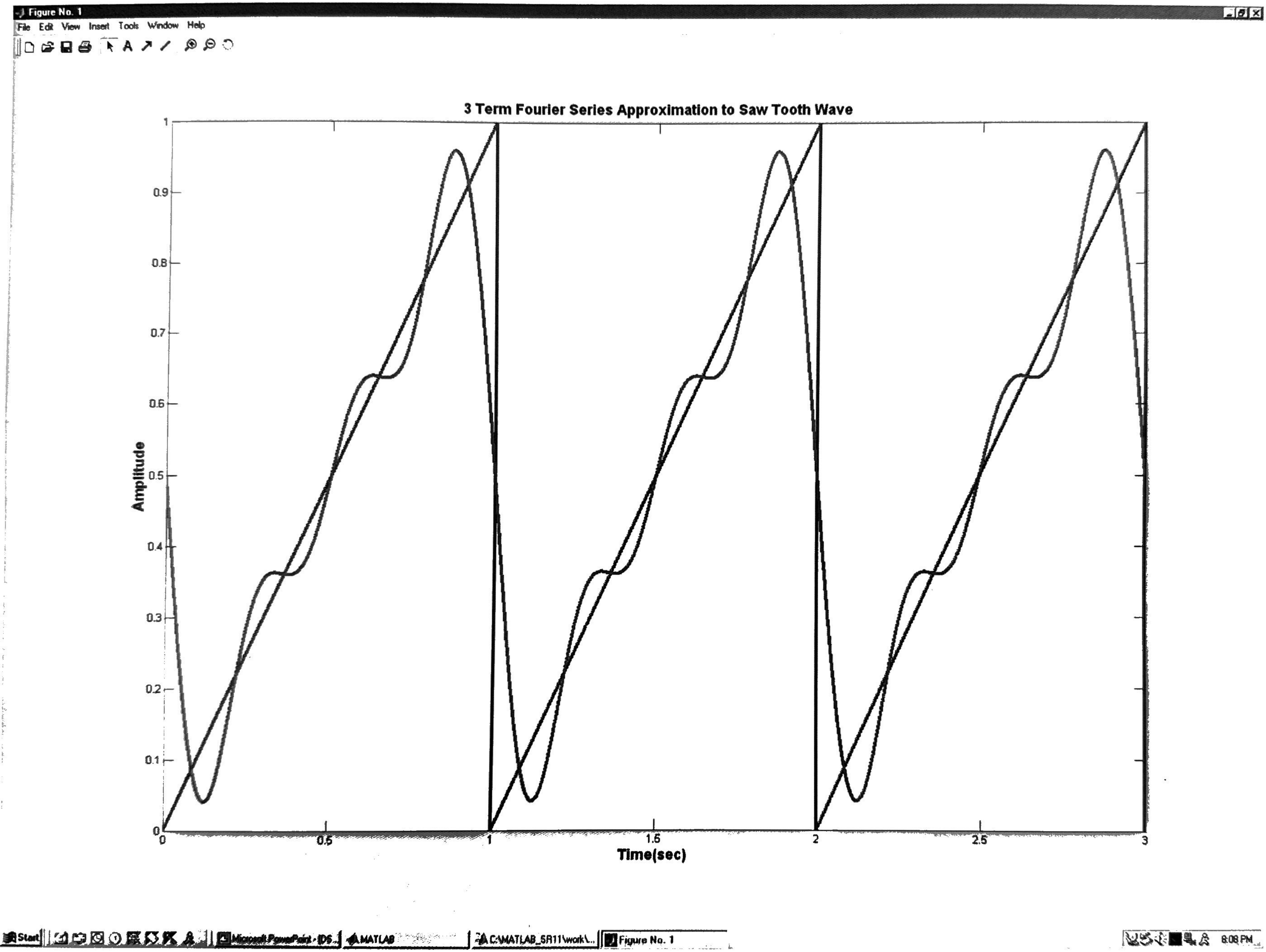


Figure No. 1
File Edit View Insert Tools Window Help
3 Term Fourier Series Approximation to Saw Tooth Wave
Amplitude
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.5
1
1.5
2
2.5
3
Time(sec)
Start
MATLAB
C:\MATLAB_SR11\work\...
Figure No. 1
8:08 PM

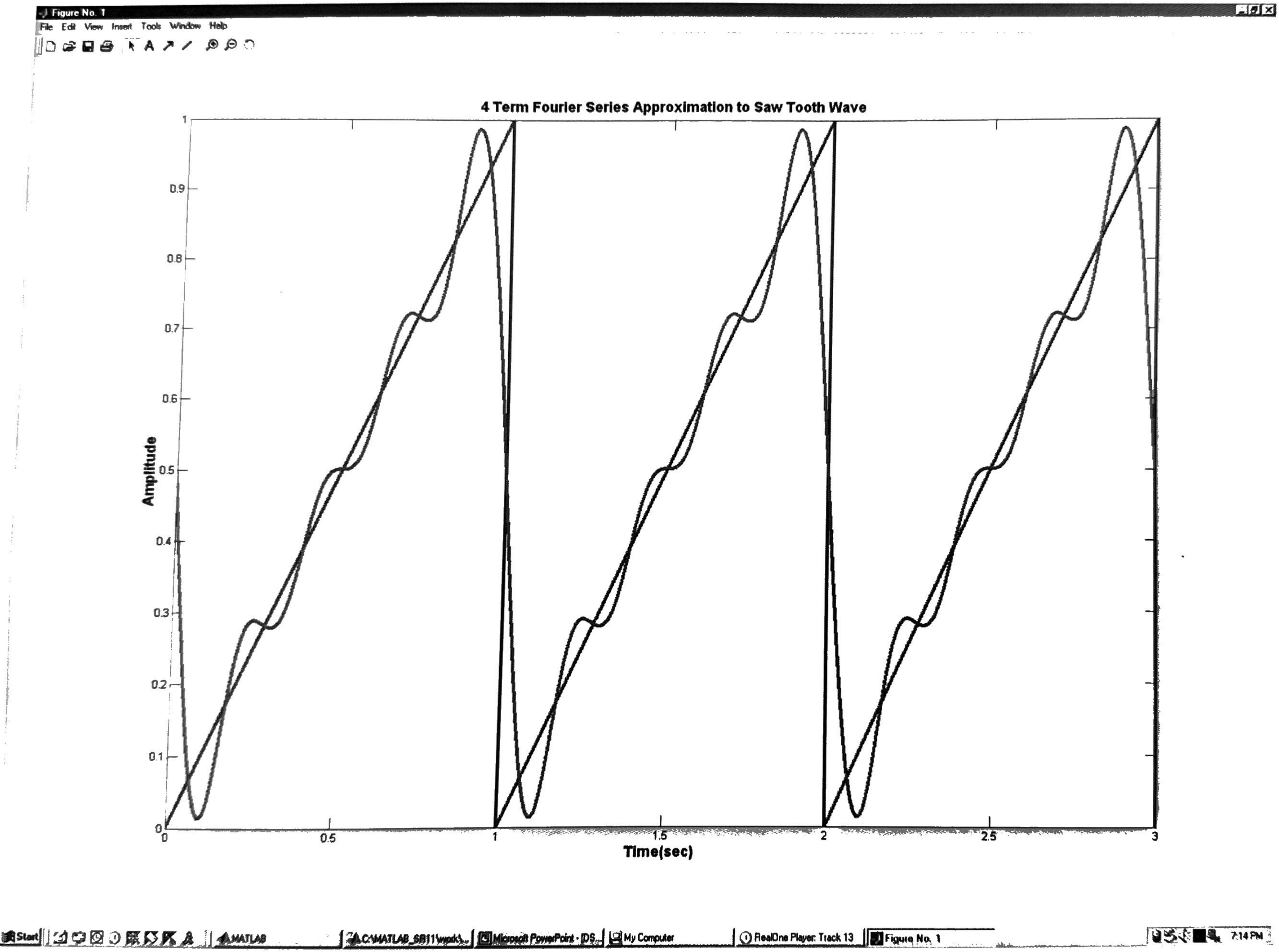


Figure No. 1
File Edit View Insert Tools Window Help
4 Term Fourier Series Approximation to Saw Tooth Wave
Amplitude
Time(sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.5
1
1.5
2
2.5
3
Start
MATLAB
My Computer
RealOne Player: Track 13
Figure No. 1
7:14 PM

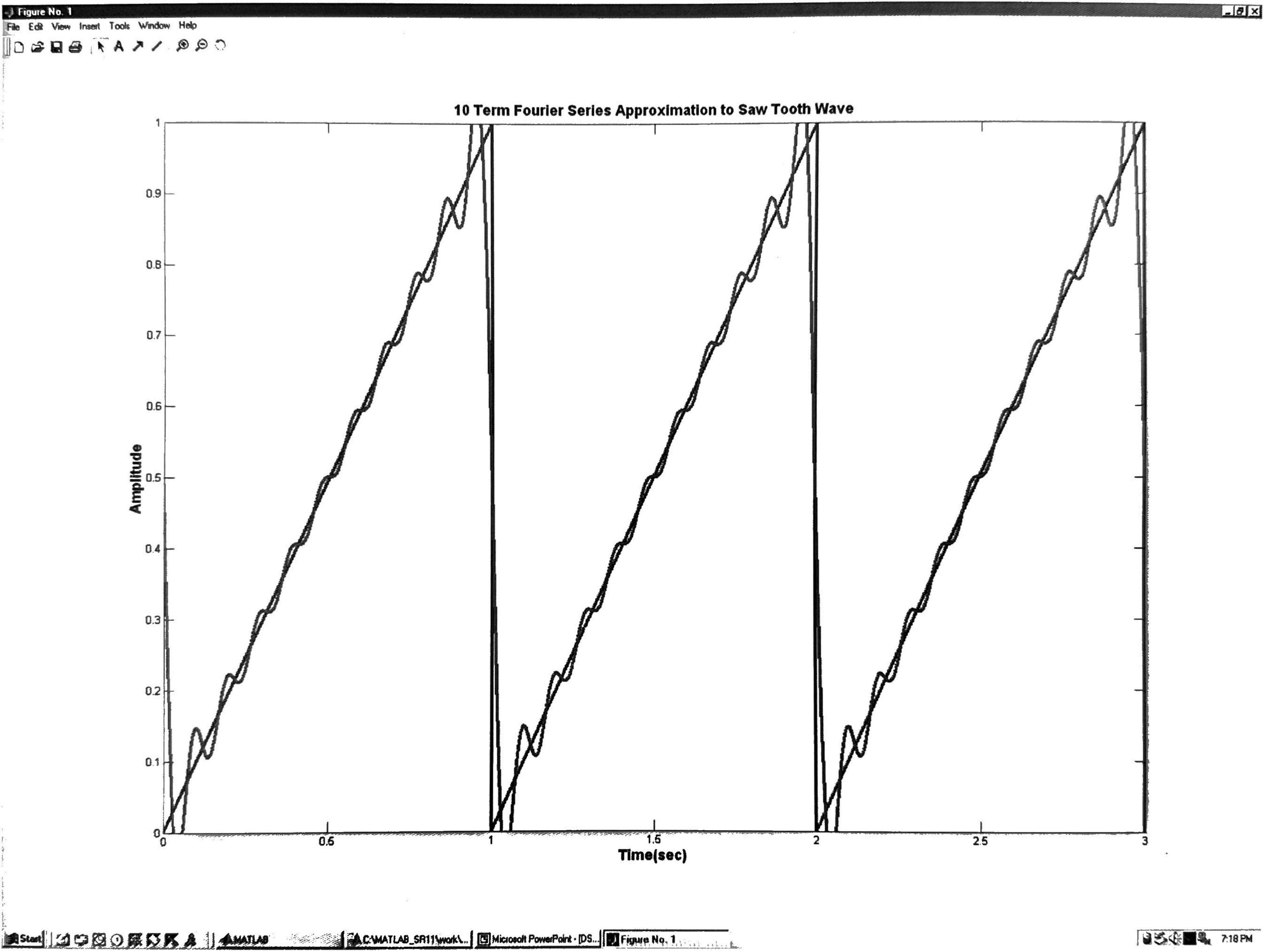


Figure No. 1
File Edit View Insert Tools Window Help
10 Term Fourier Series Approximation to Saw Tooth Wave
Amplitude
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time(sec)
0
0.6
1
1.5
2
2.5
3
Start
MATLAB
C:\MATLAB_SR11\work\...
Microsoft PowerPoint - [DS...
Figure No. 1
7:18 PM

Periodic Triangle Wave

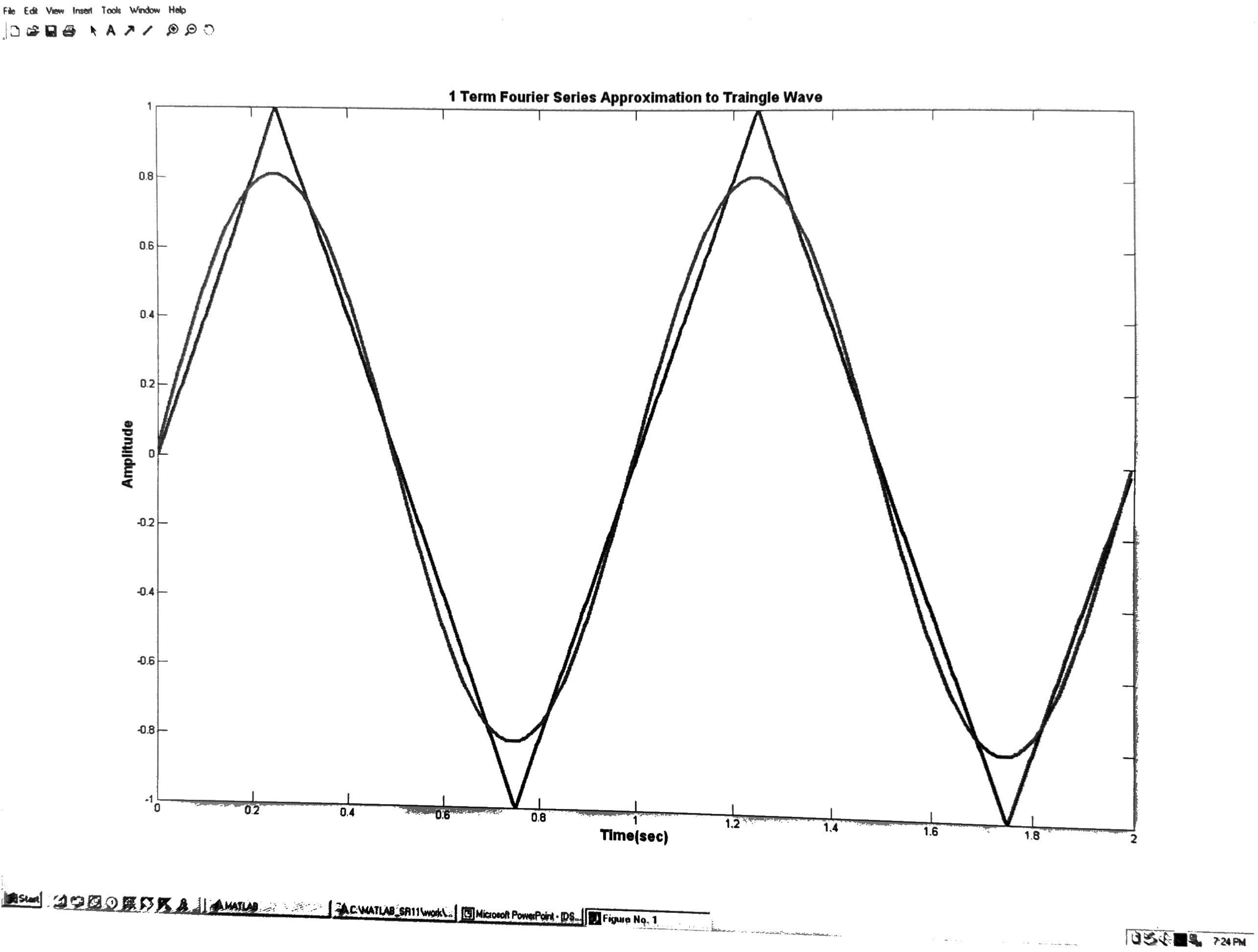
- Suppose x(t) is a triangle wave with a period T and Amplitude 1.

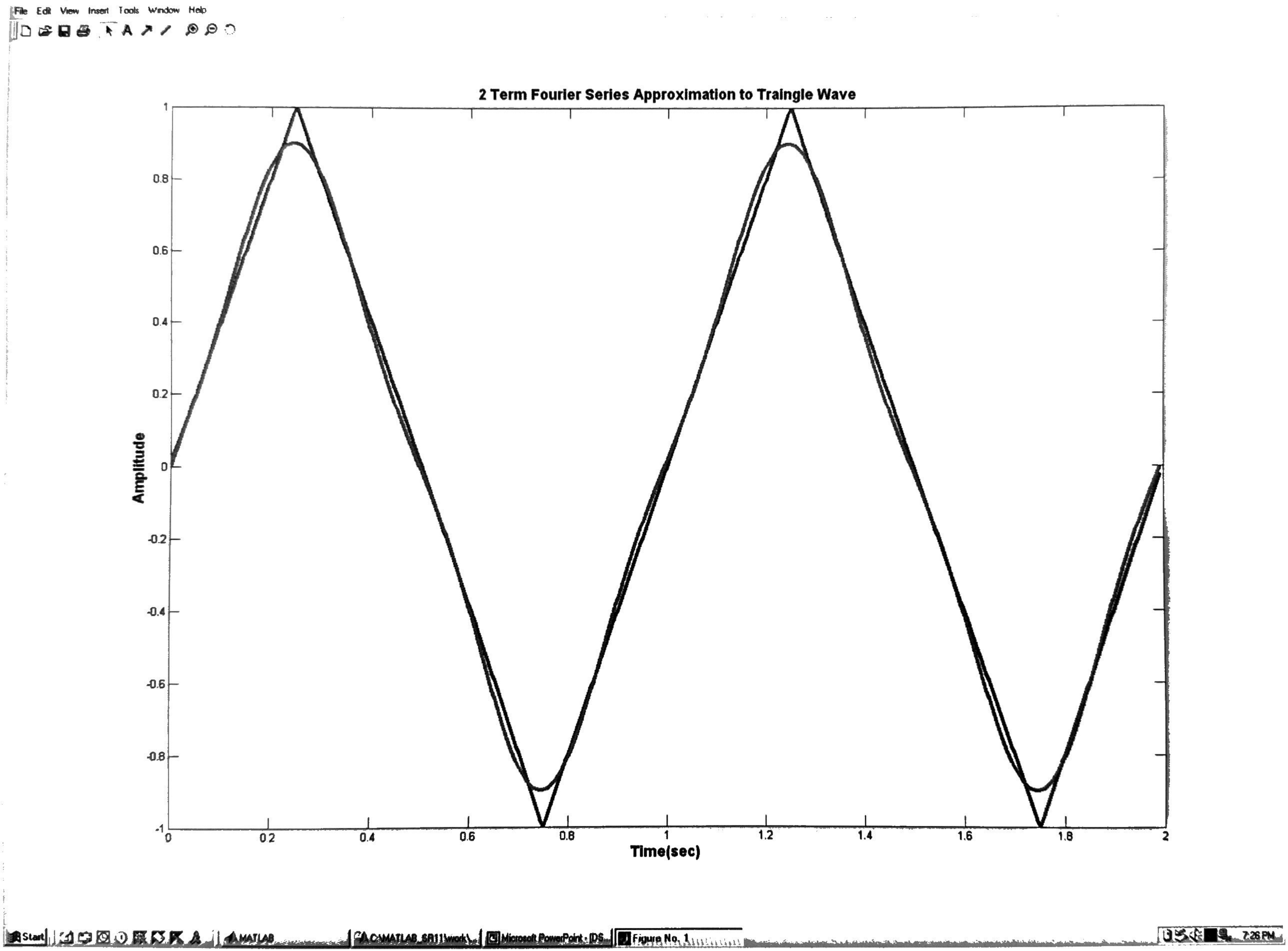
- Fourier claims

$$x(t) = (8/\pi^2) \sum_{n = 1,3,5,7\ldots} \frac{(-1)^{(n-1)/2}}{n^2} \sin(n\pi t/T)$$

$$\Omega_0 = 2\pi/T$$

- $x(t) = (8/\pi^2)\,[\sin(\pi t/T) - (1/9)\sin(3\pi t/T) \ldots\ldots]$


File Edit View Insert Tools Window Help
1 Term Fourier Series Approximation to Traingle Wave
Amplitude
Time(sec)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Start
MATLAB
C:\MATLAB_SR11\work\...
Microsoft PowerPoint - [DS...
Figure No. 1
7:24 PM


File Edit View Insert Tools Window Help
2 Term Fourier Series Approximation to Traingle Wave
Amplitude
Time(sec)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Start
MATLAB
Figure No. 1
7:26 PM

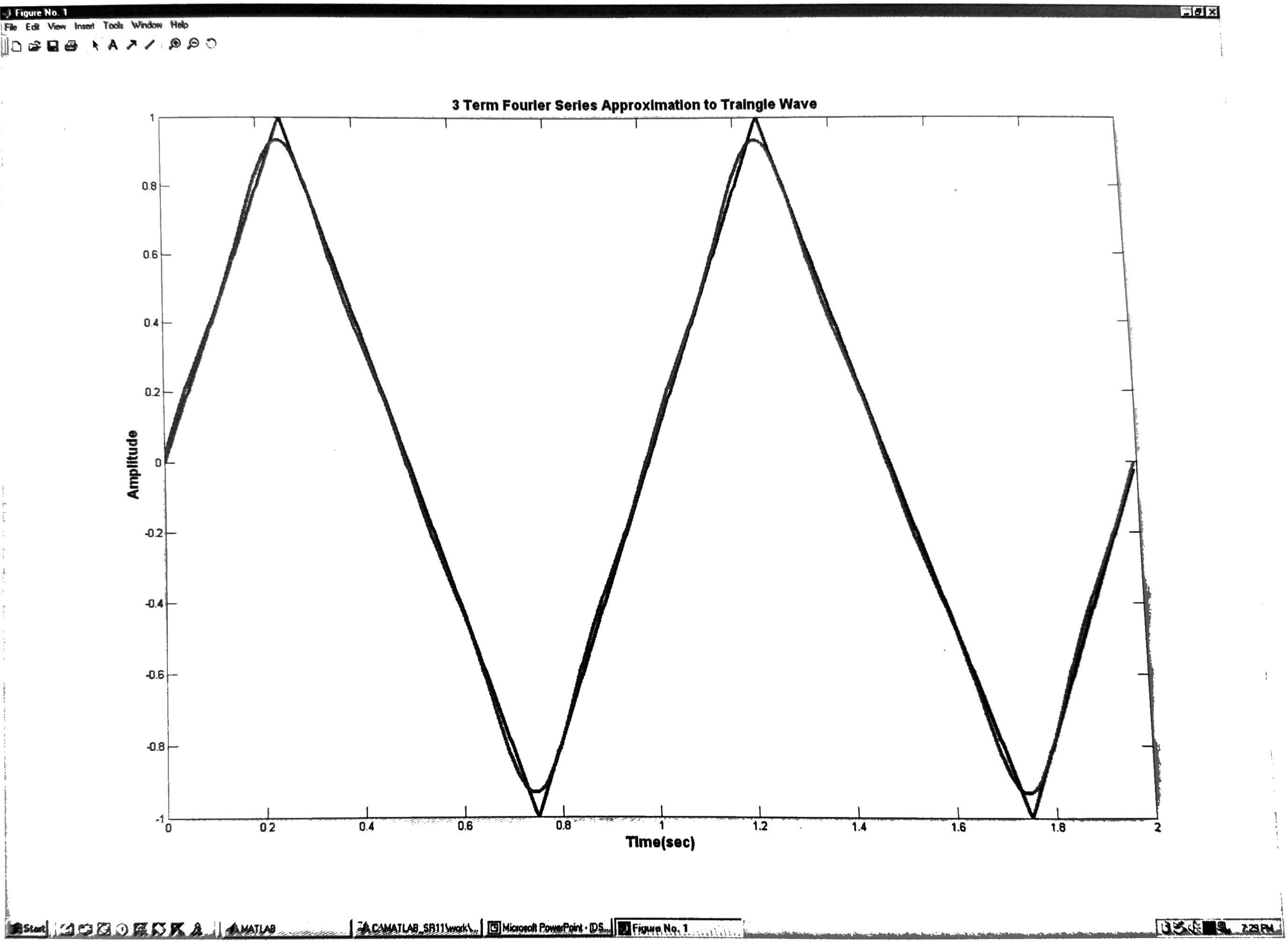


Figure No. 1
File Edit View Insert Tools Window Help
3 Term Fourier Series Approximation to Traingle Wave
Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time(sec)
Start
MATLAB
Microsoft PowerPoint - [DS...
Figure No. 1
7:29 PM

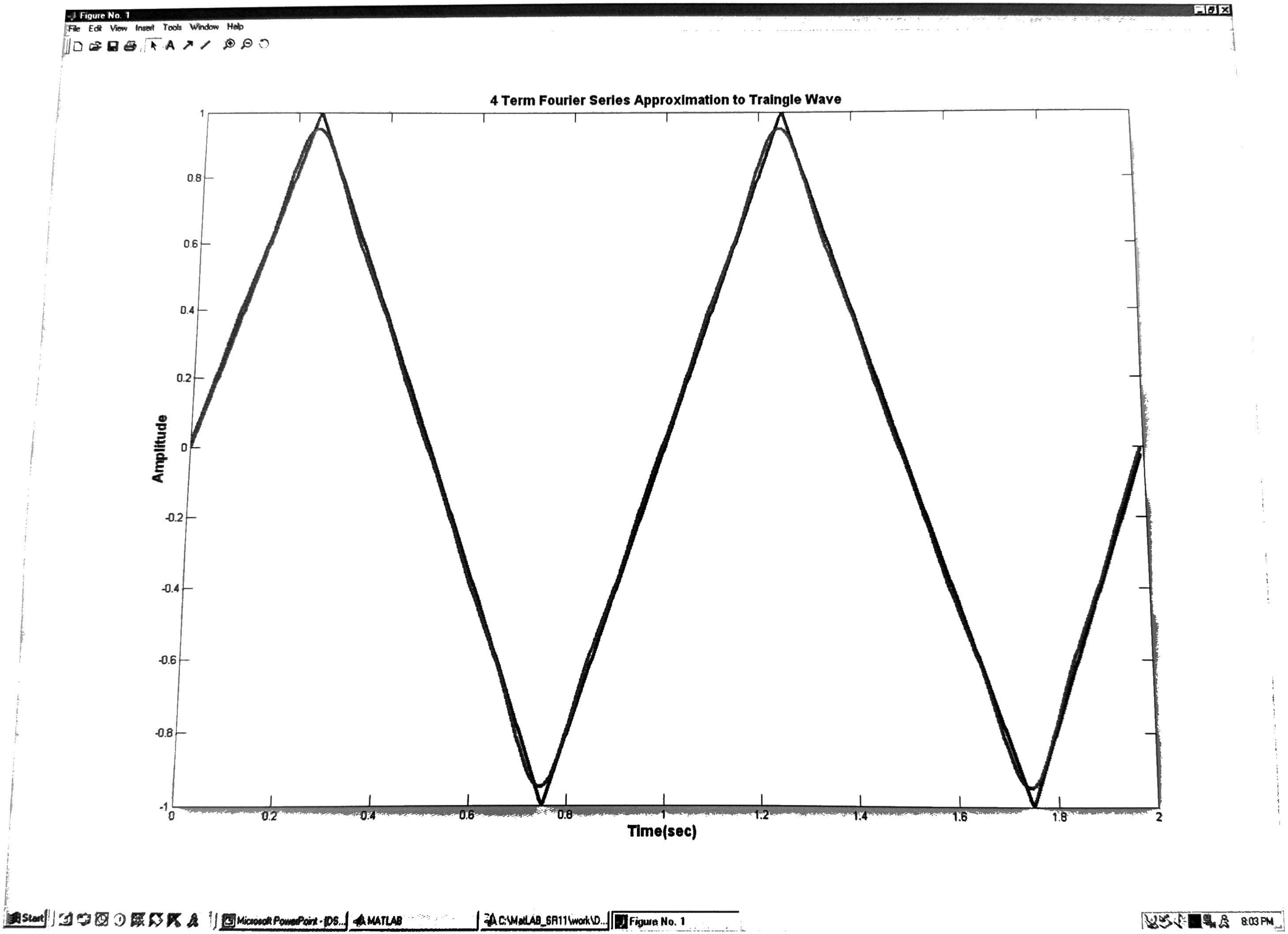


Figure No. 1
File Edit View Insert Tools Window Help
4 Term Fourier Series Approximation to Traingle Wave
Amplitude
Time(sec)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Start
Microsoft PowerPoint - [DS...
MATLAB
C:\MatLAB_6R11\work\D...
Figure No. 1
8:03 PM

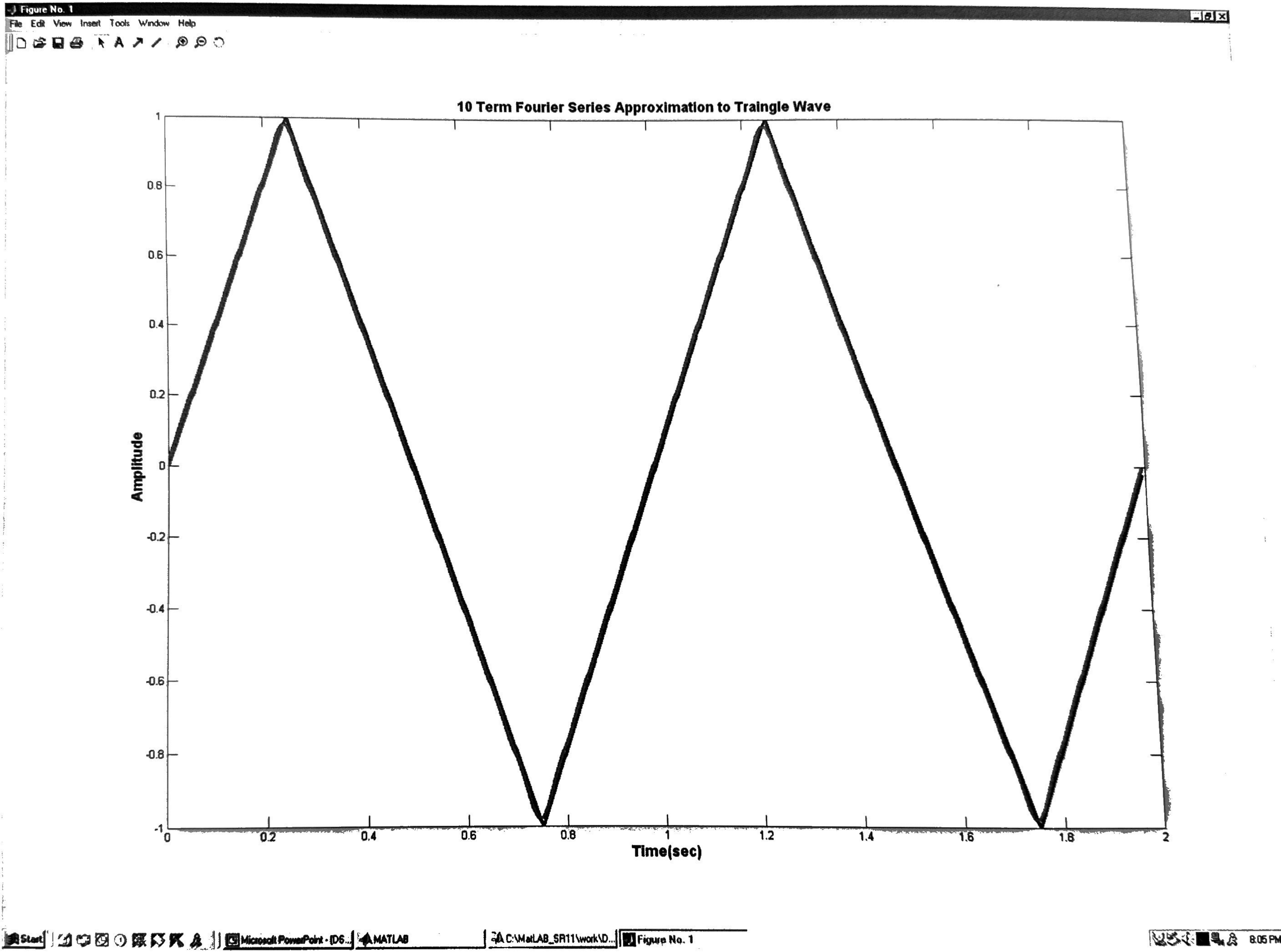


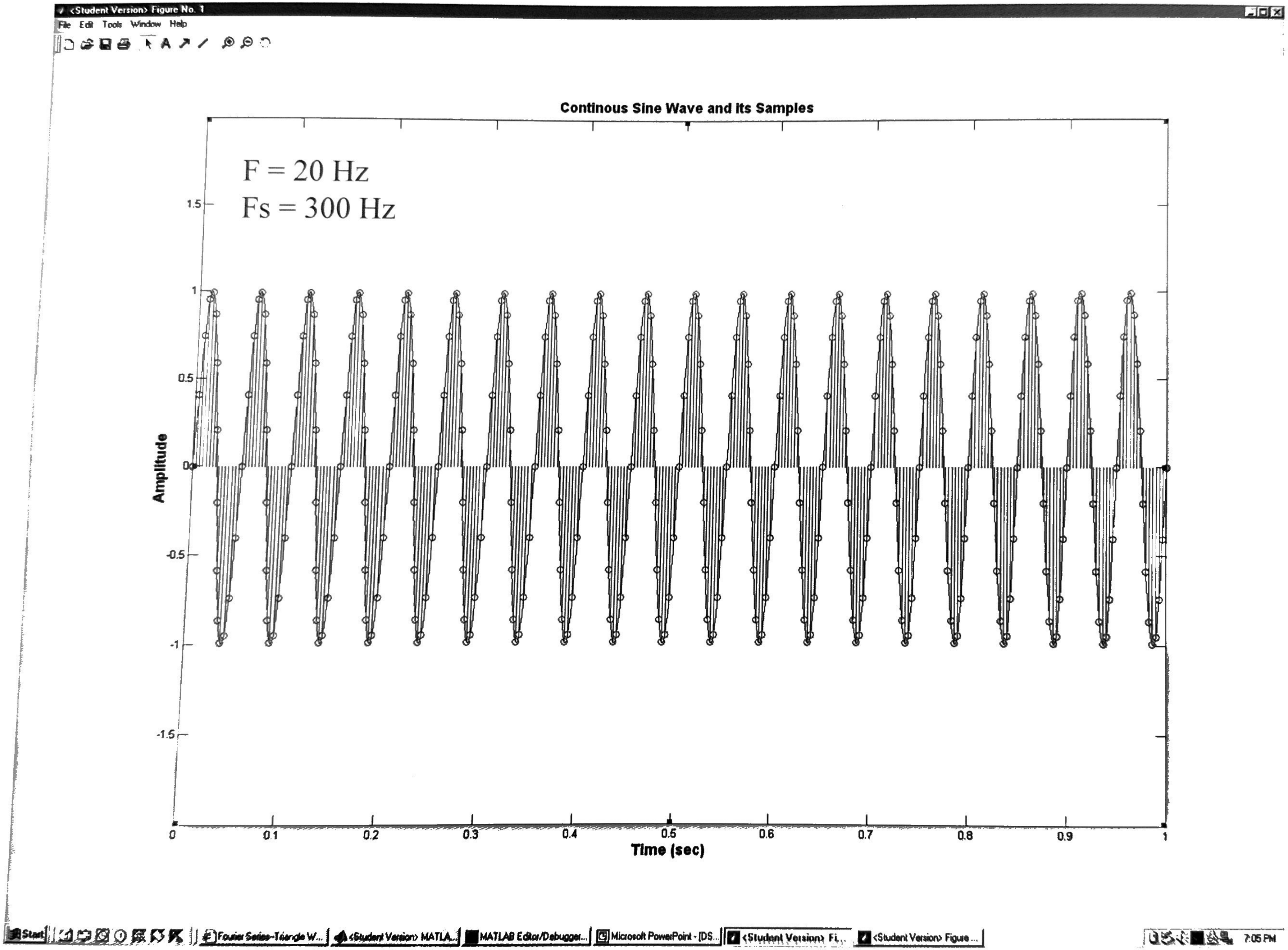
Figure No. 1
File Edit View Insert Tools Window Help
10 Term Fourier Series Approximation to Traingle Wave
Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time(sec)
Start
Microsoft PowerPoint - [D6...
MATLAB
C:\MatLAB_SR11\work\D...
Figure No. 1
8:05 PM

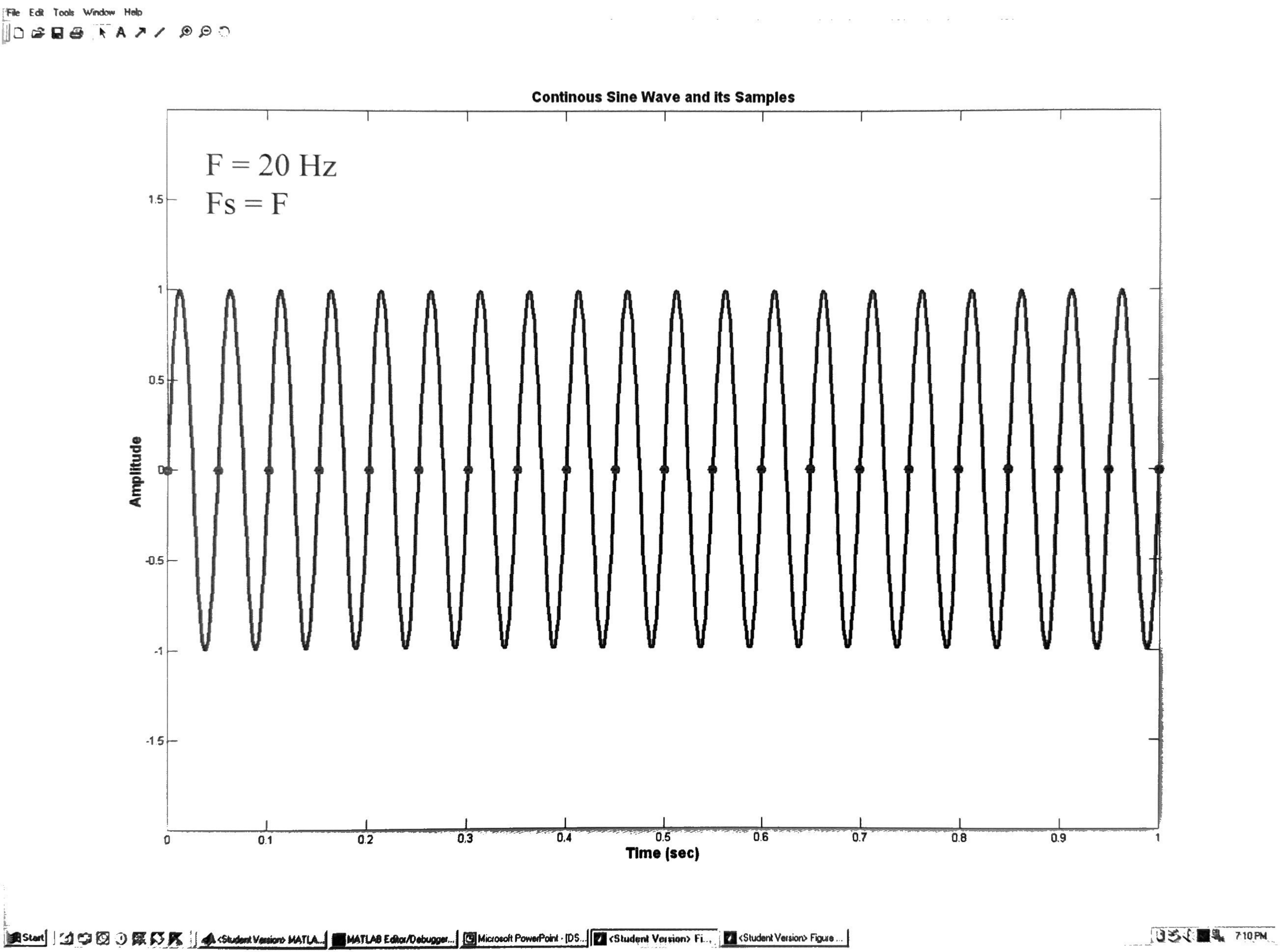
Sampling and Aliasing

- Sampling is the process by which a continuous-time (CT) signal is converted to a discrete-time (DT) signal.

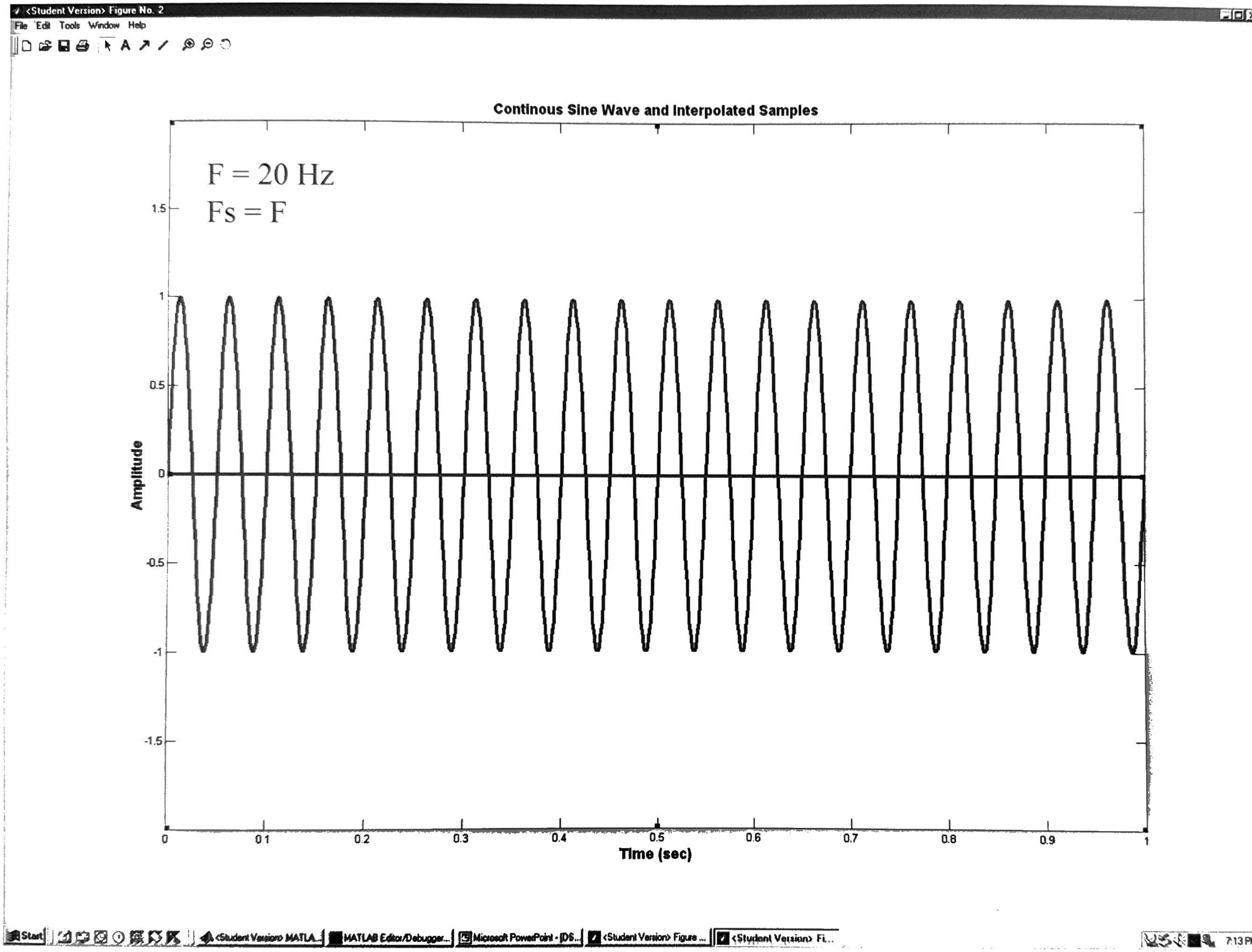
- Nyquist indicates that we must sample a signal at a rate greater than twice its maximum bandwidth.

- Aliasing occurs when signal is sampled at a rate that does not meet the Nyquist criterion.

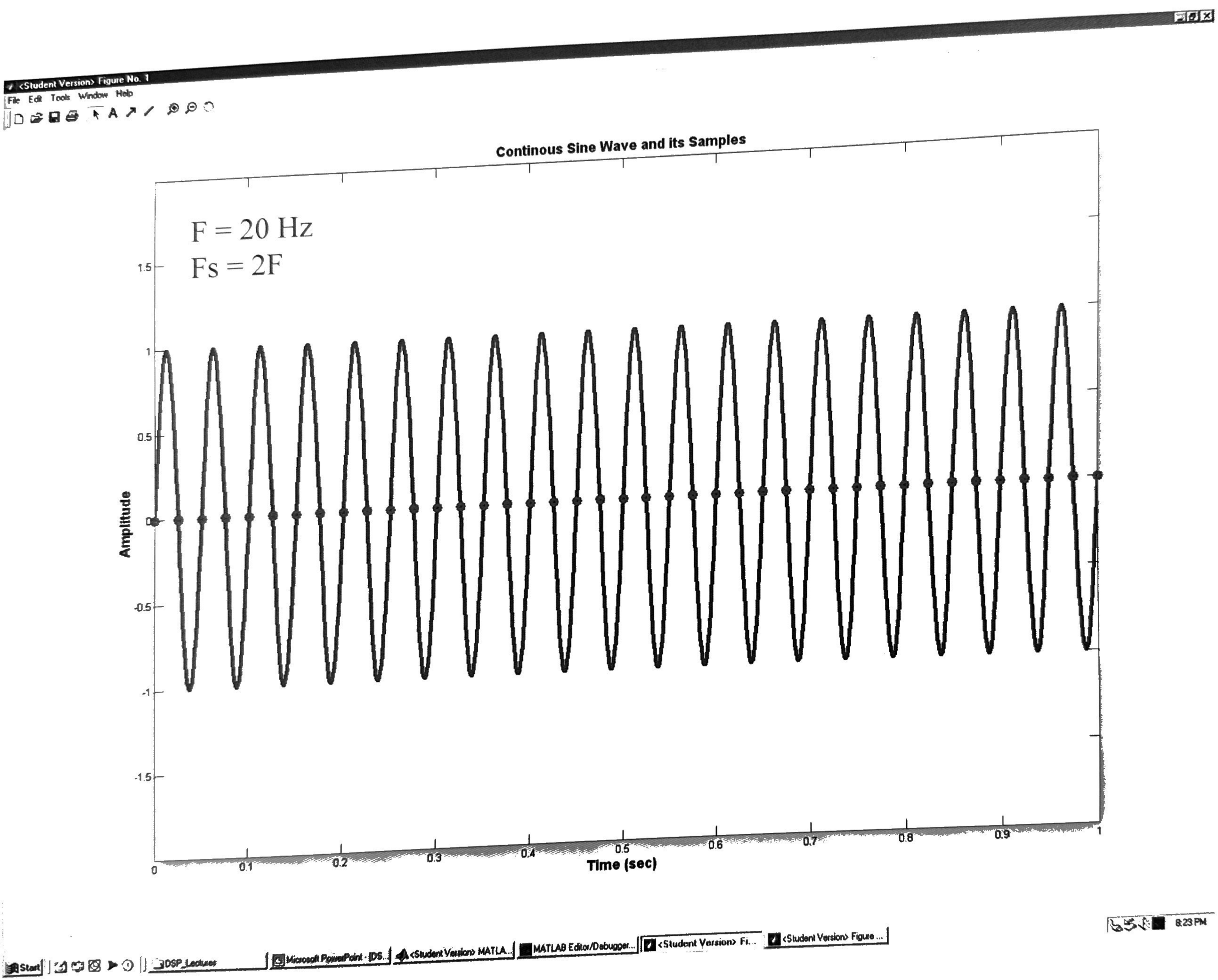

<Student Version> Figure No. 1
File Edit Tools Window Help
Continous Sine Wave and its Samples
F = 20 Hz
Fs = 300 Hz
1.5
1
0.5
0
-0.5
-1
-1.5
Amplitude
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (sec)
Start
Microsoft PowerPoint - [DS...
MATLAB Editor/Debugger...
<Student Version> Figure ...
7:05 PM



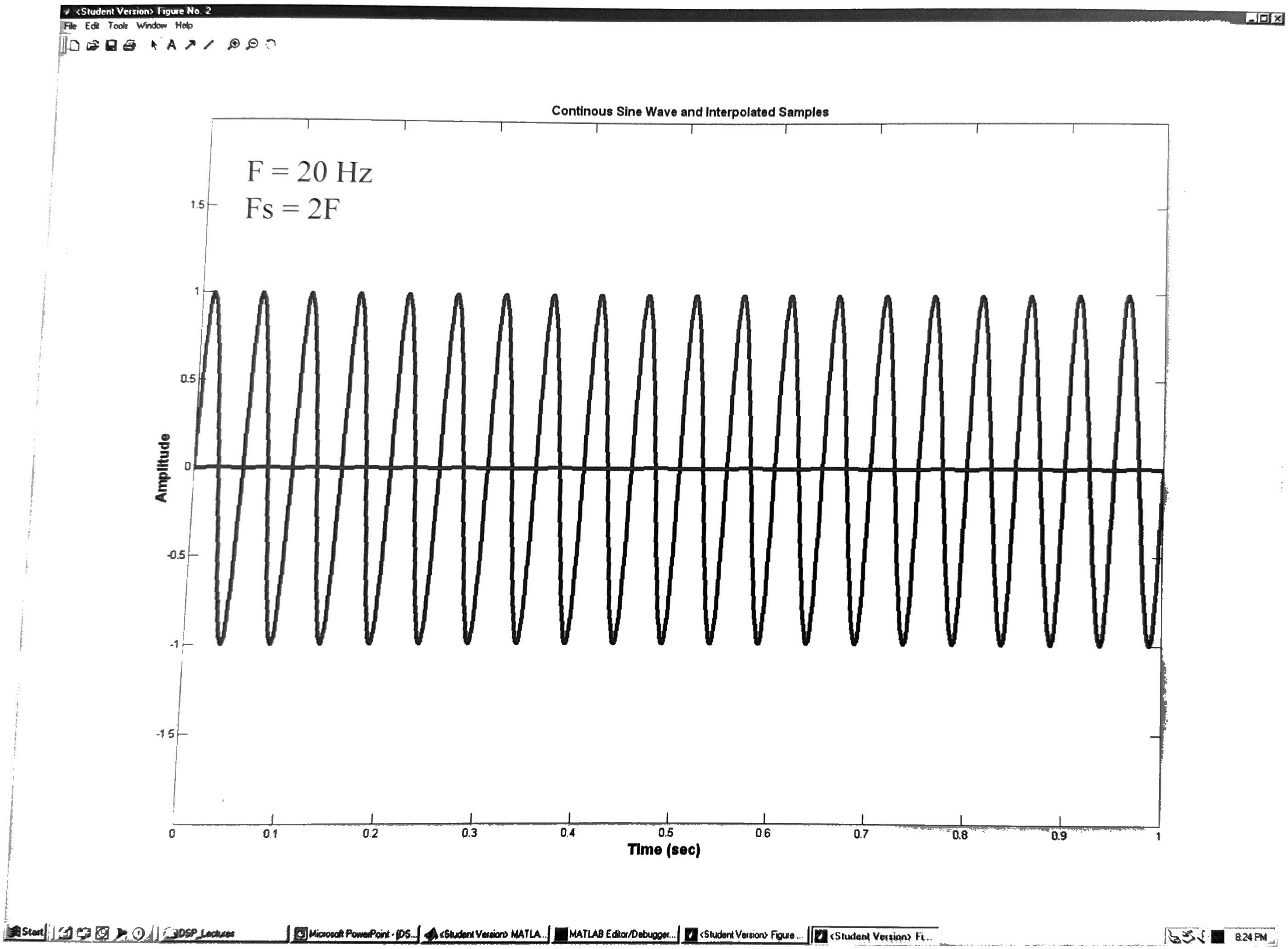
File Edit Tools Window Help
Continous Sine Wave and its Samples
F = 20 Hz
Fs = F
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Start
<Student Version> MATLA...
MATLAB Editor/Debugger...
Microsoft PowerPoint - [DS...
<Student Version> Fi...
<Student Version> Figure ...
7:10 PM



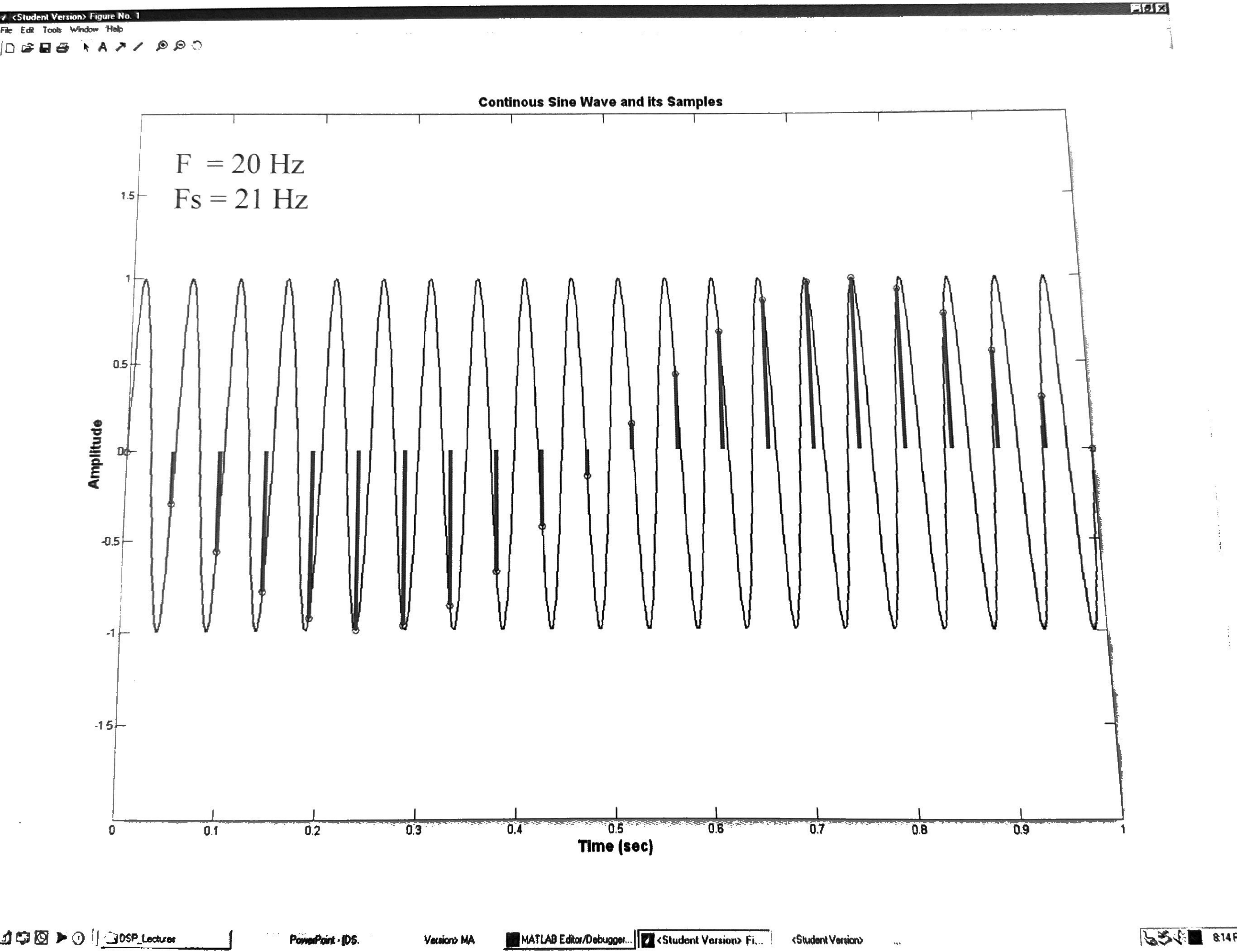
<Student Version> Figure No. 2
File Edit Tools Window Help
Continous Sine Wave and Interpolated Samples
F = 20 Hz
Fs = F
Amplitude
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (sec)
Start
<Student Version> MATLA...
MATLAB Editor/Debugger...
Microsoft PowerPoint - [DS...
<Student Version> Figure ...
<Student Version> Fi...
7:13 PM



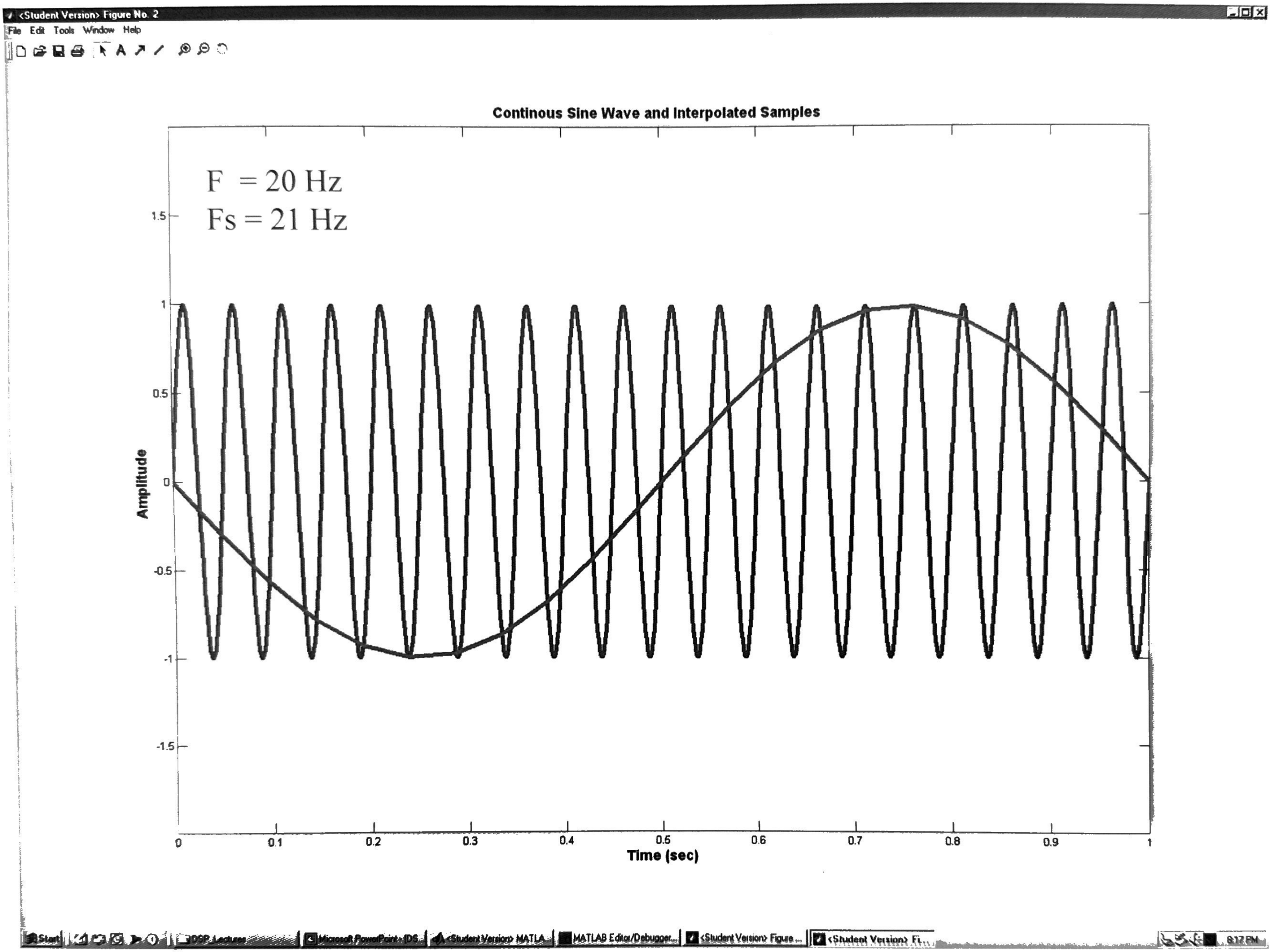
<Student Version> Figure No. 1
File Edit Tools Window Help
Continous Sine Wave and its Samples
F = 20 Hz
Fs = 2F
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Start
DSP_Lectures
Microsoft PowerPoint - [DS...
<Student Version> MATLA...
MATLAB Editor/Debugger...
<Student Version> Fi...
<Student Version> Figure ...
8:23 PM



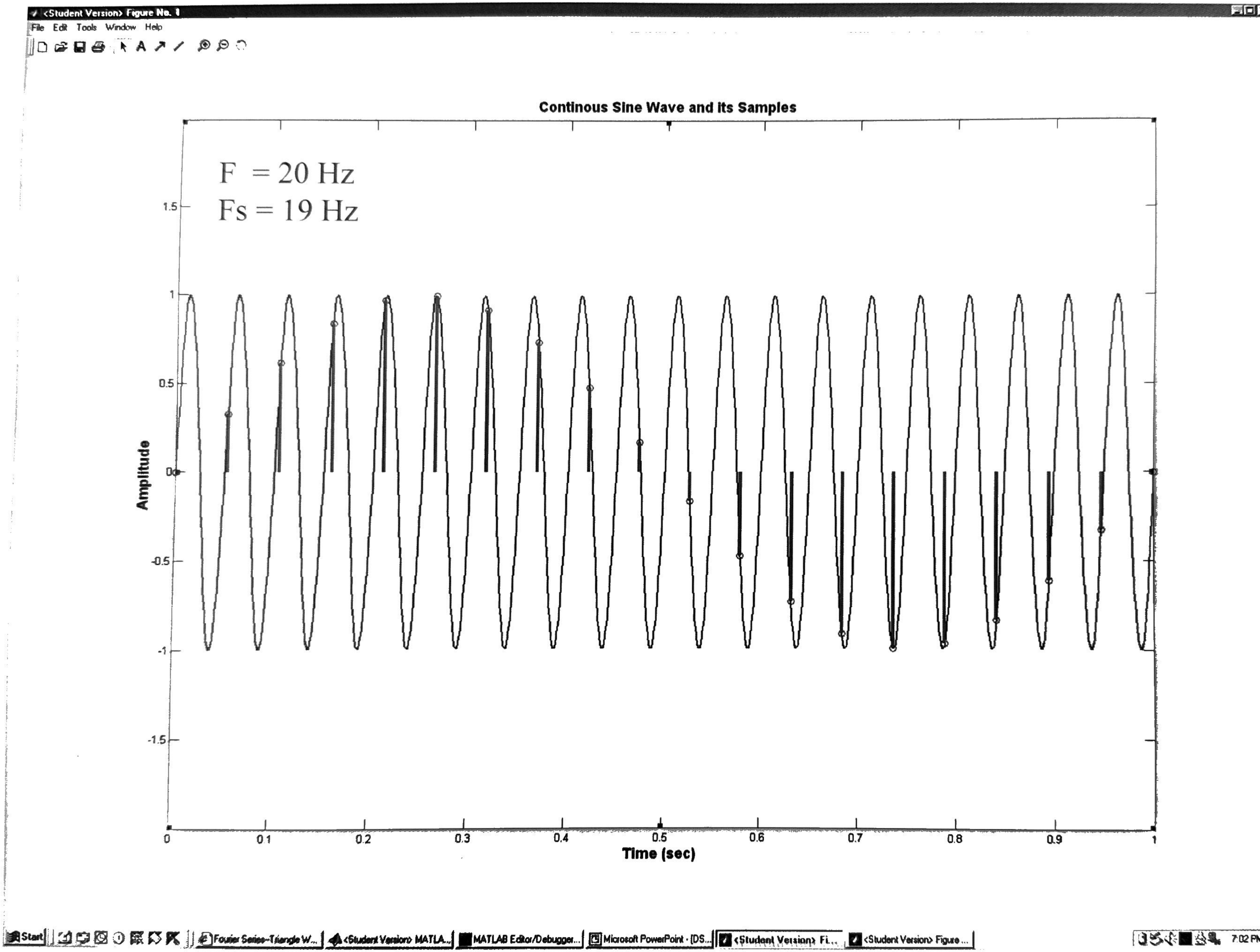
<Student Version> Figure No. 2
File Edit Tools Window Help
Continous Sine Wave and Interpolated Samples
F = 20 Hz
Fs = 2F
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Start
DSP_Lectures
Microsoft PowerPoint - [DS...
<Student Version> MATLA...
MATLAB Editor/Debugger...
<Student Version> Figure ...
<Student Version> Fi...
8:24 PM



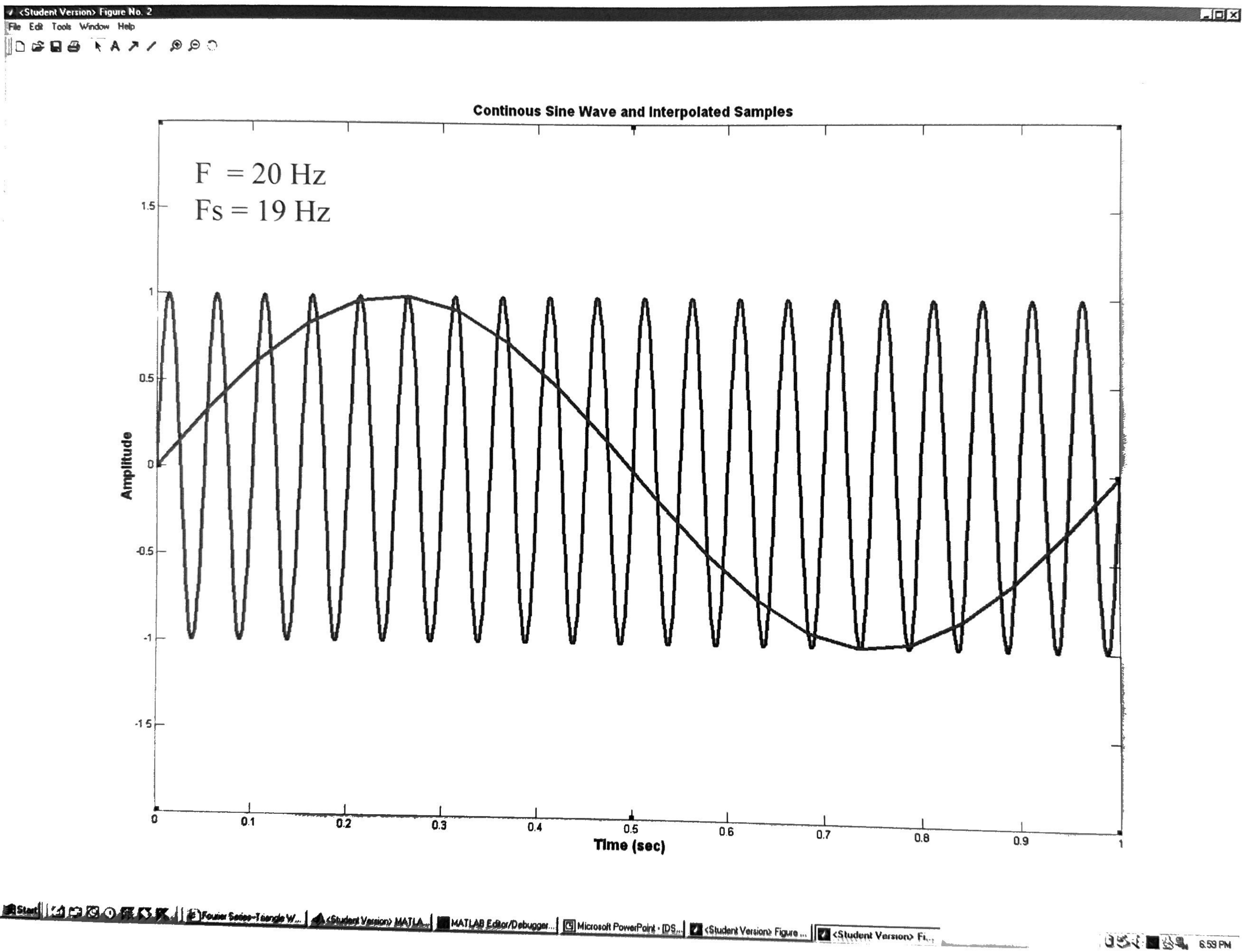
<Student Version> Figure No. 1
File Edit Tools Window Help
Continous Sine Wave and its Samples
F = 20 Hz
Fs = 21 Hz
Amplitude
1.5
1
0.5
0
-0.5
-1
-1.5
Time (sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
DSP_Lectures
MATLAB Editor/Debugger...
<Student Version> Fi...
<Student Version>



<Student Version> Figure No. 2
File Edit Tools Window Help
Continous Sine Wave and Interpolated Samples
F = 20 Hz
Fs = 21 Hz
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9



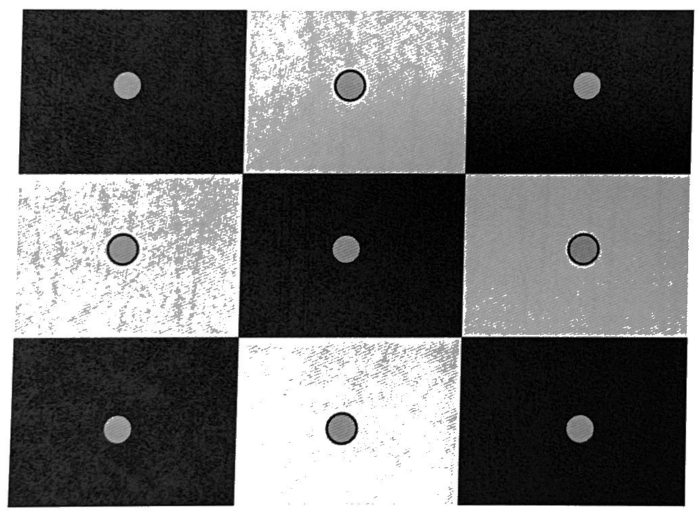
<Student Version> Figure No. 1
File Edit Tools Window Help
Continous Sine Wave and its Samples
F = 20 Hz
Fs = 19 Hz
Amplitude
Time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Start
Microsoft PowerPoint
MATLAB Editor/Debugger
7:02 PM



<Student Version> Figure No. 2
File Edit Tools Window Help
Continous Sine Wave and Interpolated Samples
F = 20 Hz
Fs = 19 Hz
Amplitude
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (sec)
Start
MATLAB Editor/Debugger...
Microsoft PowerPoint - [DS...
<Student Version> Figure ...
6:59 PM

Sampling and Aliasing

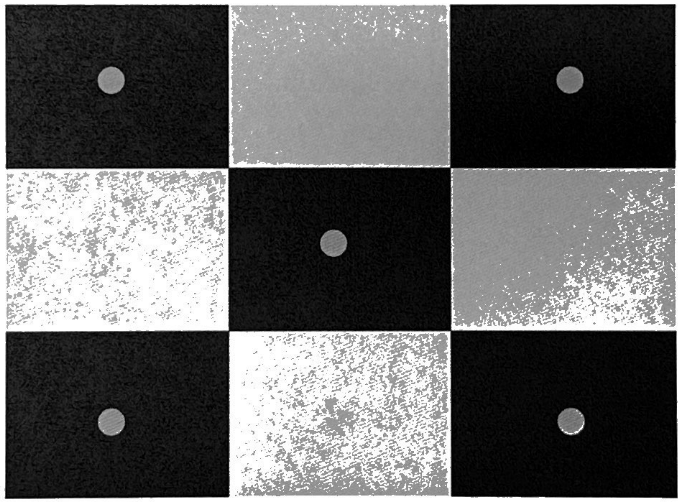
2D Sampling at Nyquist Rate



Samples

Sampling and Aliasing

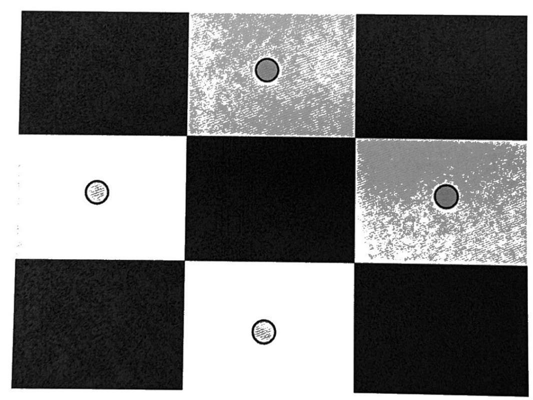
Aliasing in 2D Sampling Below Nyquist Rate



Samples

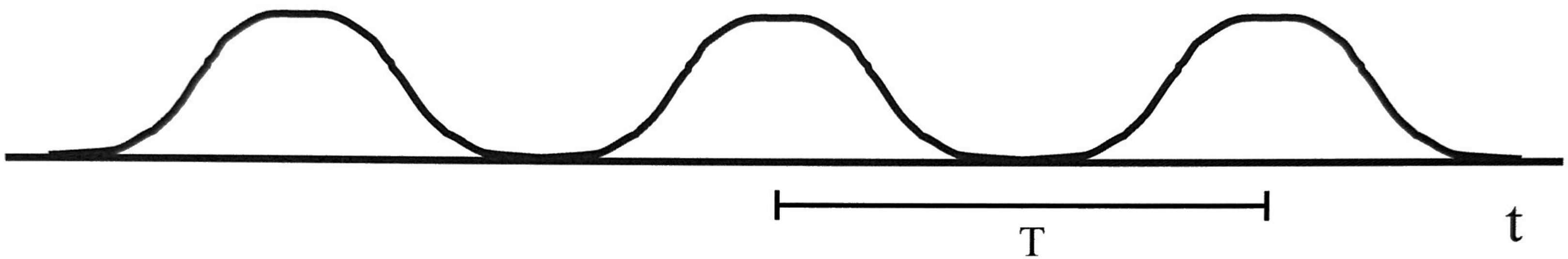
Sampling and Aliasing

Aliasing in 2D Sampling Below Nyquist Rate



- Samples

Periodic CT Signals



$$x(t) = a_0 + \sum_{k=1}^{\infty} \{ B_k \cos(k\Omega_0 t) + C_k \sin(k\Omega_0 t) \} \qquad -\infty < t < \infty$$

$$\Omega_0 = 2\pi / T$$

Representing Finite Length Discrete Signals

- Since a discrete periodic signal x[n] is just samples of a continuous periodic signal x(t), we can figure out its frequency content by sampling the Continuous Fourier Series of the continuous periodic signal at the Nyquist Rate

$$x(t) = a_0 + \sum_{k=1}^{\infty} \{ B_k \cos(k\Omega_0 t) + C_k \sin(k\Omega_0 t) \} \qquad \Omega_0 = 2\pi / T \quad -\infty < t < \infty$$

Sample:

$t = n\, T_{sample}$

$T = N\, T_{sample}$

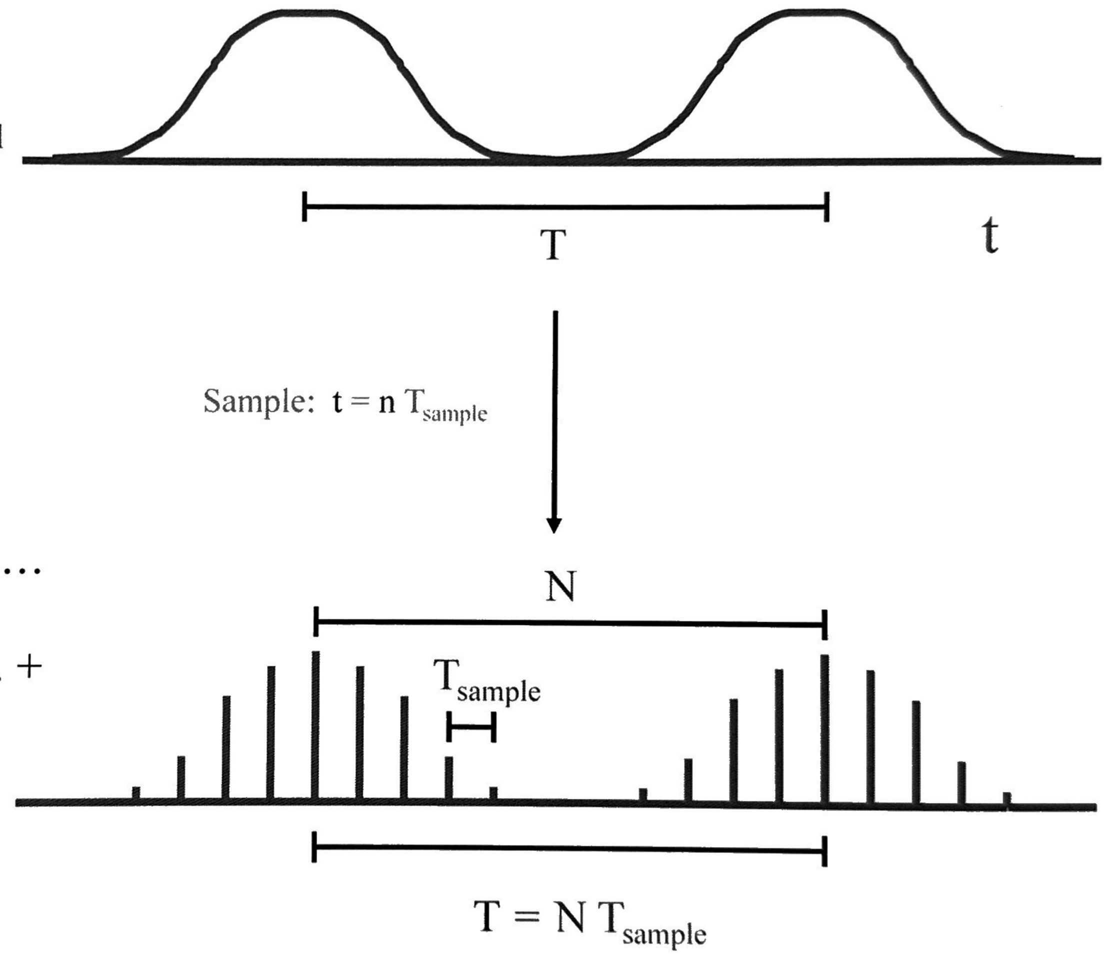
This can't be true because of the Nyquist Sampling Criterion

$$x[n] = a_0 + \sum_{k=1}^{\infty} \{ B_k \cos[k\omega_0 n] + C_k \sin[k\omega_0 n] \} \qquad \omega_0 = 2\pi / N$$

$$-\infty < n < \infty$$

Implications of Nyquist Sampling Criterion

- Nyquist Sampling Criterion requires highest frequency in continuous signal to be $\leq F_{sample}/2$



- Continuous Fourier Series can express continuous signal as

$$a_0 + B_1 \cos \Omega_0 t + B_2 \cos 2\Omega_0 t + \ldots + B_{Nyquist} \cos \Omega_{Nyquist} t + C_1 \sin \Omega_0 t + C_2 \sin 2\Omega_0 t + \ldots + C_{Nyquist} \sin \Omega_{Nyquist} t$$

$$\Omega_{Nyquist} = 2\pi\,(F_{sample}/2)$$

Representing Finite Length Discrete Signals

- The Fourier Series for the replicated discrete signal x[n] is

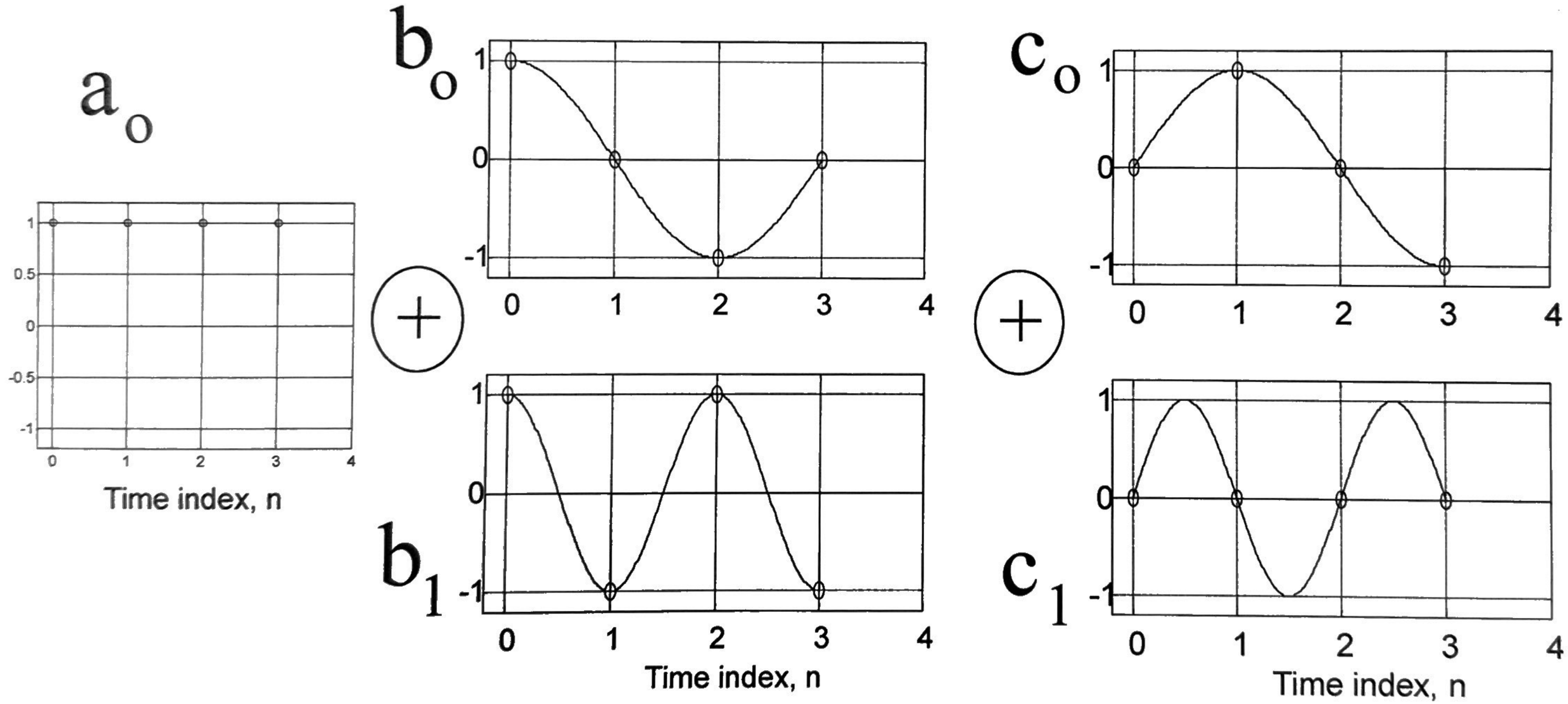
$$x[n] = a_0 + \sum_{k=1}^{N/2} \{ B_k \cos[k\omega_0 n] + C_k \sin[k\omega_0 n] \} \qquad \omega_0 = 2\pi / N \quad -\infty < n < \infty$$

- The Fourier Series for the finite length discrete signal x[n] is

$$x[n] = a_0 + \sum_{k=1}^{N/2} \{ B_k \cos[k\omega_0 n] + C_k \sin[k\omega_0 n] \} \qquad \omega_0 = 2\pi / N \quad -N/2 <= n < N/2$$

What does this all mean?

It means that, for example, any 4 data points [d0 d1 d2 d3] can be represented by the sum of the following DT sequences (CT sines and cosines included as a guide for the eye):



Alternate Ways to Represent Data

So, the information in the four points

$$[d_0 \quad d_1 \quad d_2 \quad d_3]$$

are equally well $\downarrow$ represented by:

$$[a_0 \quad b_0 \quad c_0 \quad b_1 \quad \cancel{c_1}]$$

These "spectral coefficients" are easy to find!

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ b_1 \end{bmatrix}$$

Invert this equation to solve for spectral coefficients.

The Professional Way: FFT

Matlab offers a powerful tool for finding spectral coefficients, the FFT. Suppose that, in Matlab, we define:

$$D = [d_0 \quad d_1 \quad d_2 \quad d_3]$$

Then fft(D) returns the answer:

$$P^* \; [a_0 \quad (b_0 - jc_0)/2 \quad b_1 \quad (b_0 + jc_0)/2]$$

where P is the number of points in D (4 in this case).

More FFT Examples from Matlab

The FFT of a P = 8 point sequence would return:

$P^*\ [a_0 \quad (b_0 - jc_0)/2 \quad (b_1 - jc_1)/2 \quad (b_2 - jc_2)/2 \quad b_3 \quad \ldots$

$(b_2 + jc_2)/2 \quad (b_1 + jc_1)/2 \quad (b_0 + jc_0)/2\ ]$

Some 4 point examples: i = [0 0.25 0.5 0.75]

FFT([1 1 1 1]) returns [4 0 0 0]

FFT(sin(2*pi*i)) returns [0 -2j 0 2j]

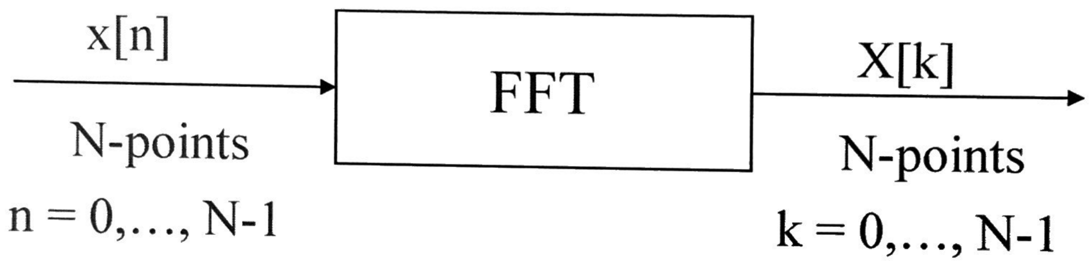
FFT(sin(4*pi*i)) returns [0 0 0 0]

FFT(cos(2*pi*i)) returns [0 2 0 2]

FFT(cos(4*pi*i)) returns [0 0 4 0]

FFT

Fast Fourier Transform



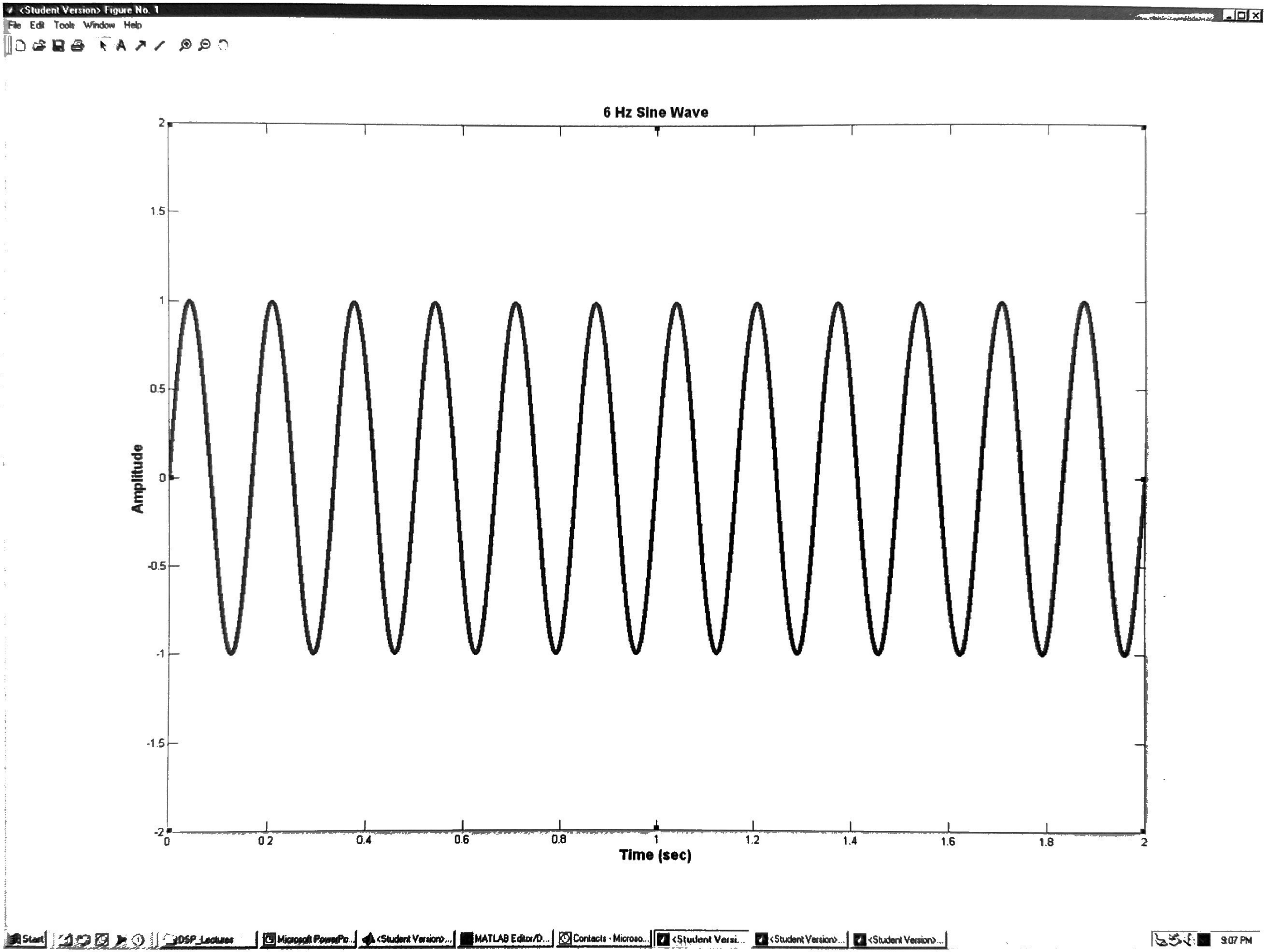
$a_0 = X[0]/N$

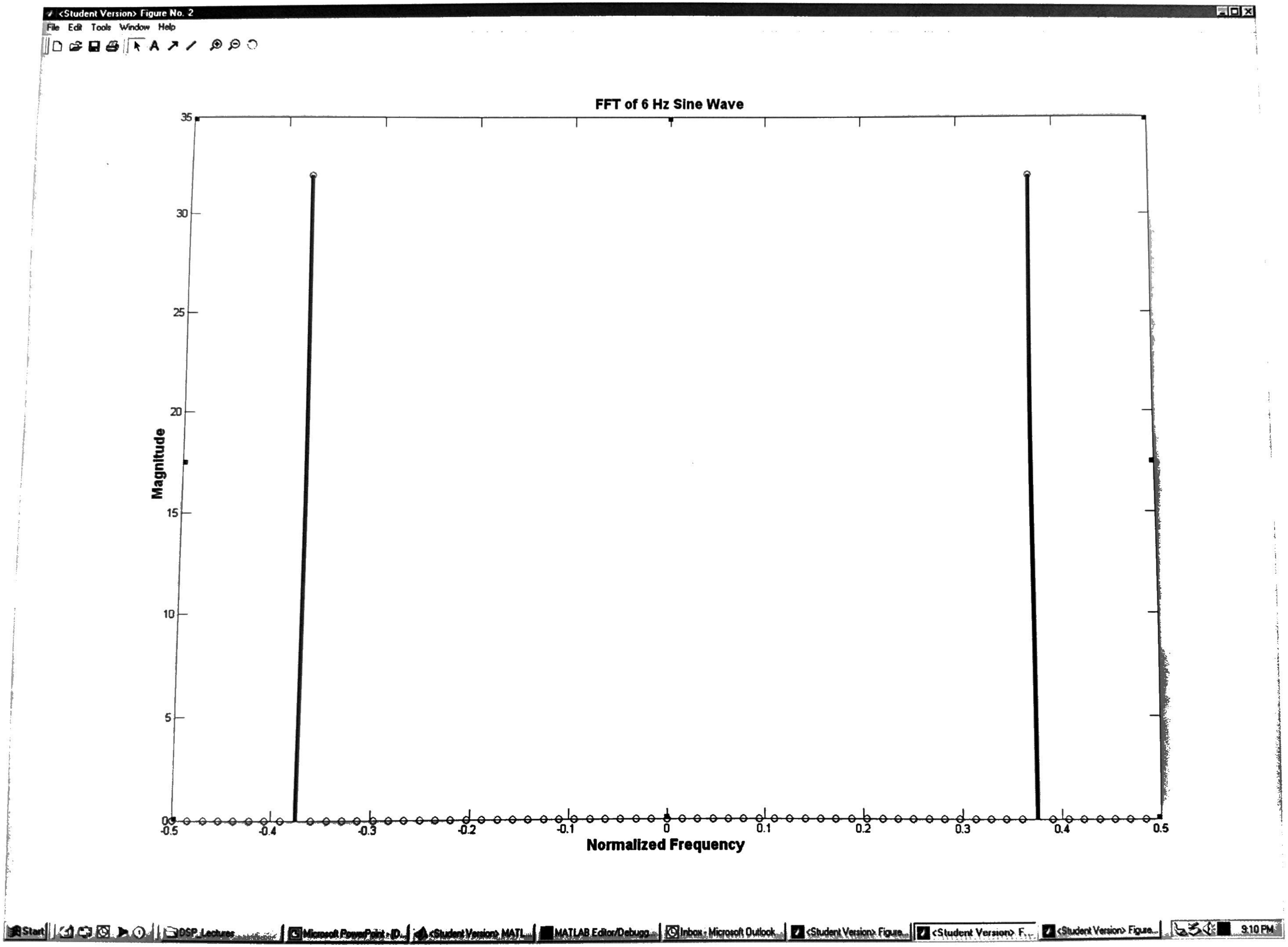
$B_k = 2*\mathrm{Re}\{X[k]/N\}$ $\quad C_k = 2*\mathrm{Im}\{X[k]/N\}$ $\quad k = 1,\ldots,N/2-1$

$B_{N/2} = \mathrm{Re}\{X[k]/N\}$ $\quad C_{N/2} = \mathrm{Im}\{X[k]/N\} = 0$ $\quad k = N/2$

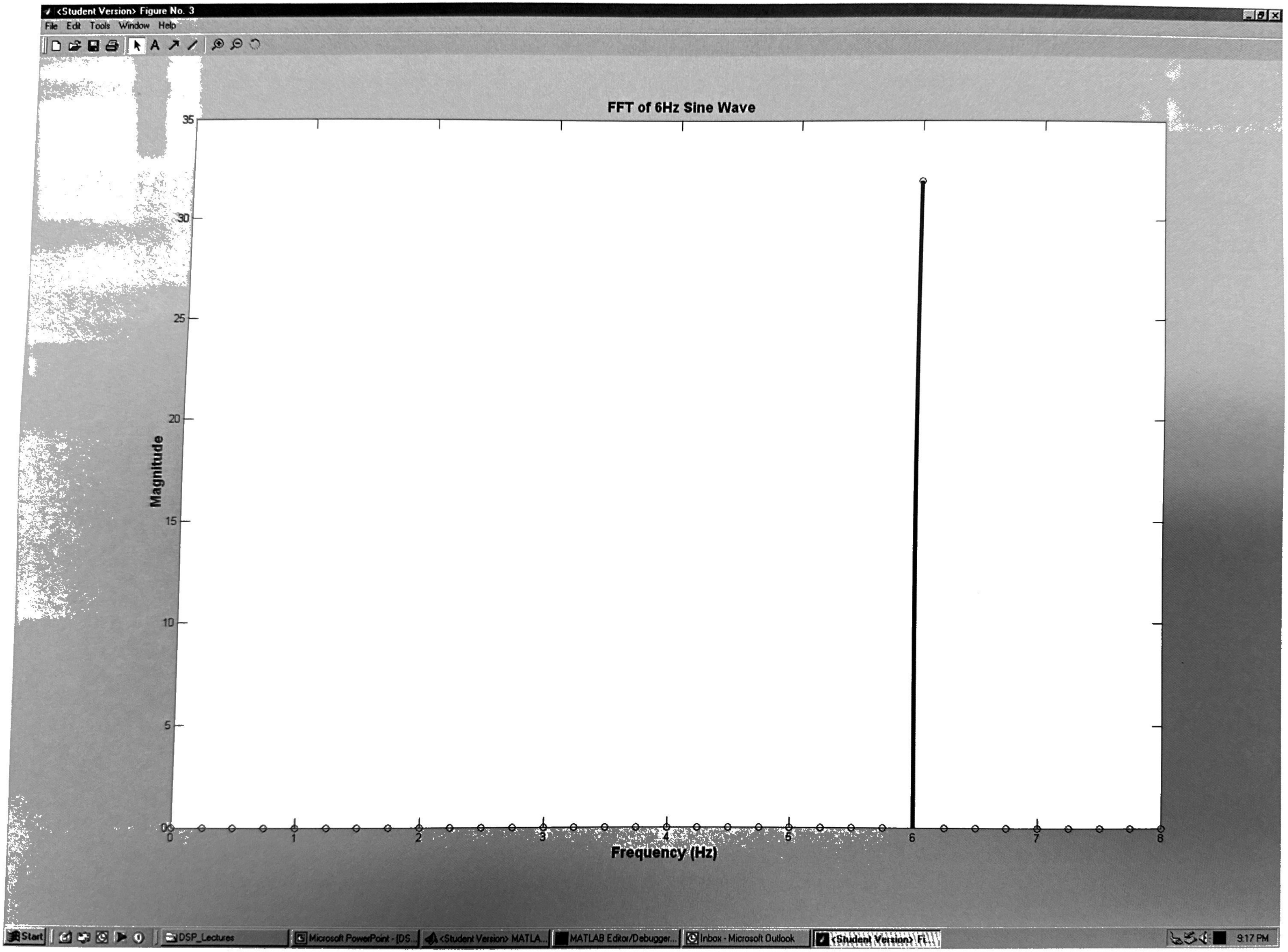
Sine Wave Examples

- FFT of 6 Hz Sine Wave will have:
  - 1 nonzero component at 6 Hz.


<Student Version> Figure No. 1
File Edit Tools Window Help
6 Hz Sine Wave
Amplitude
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (sec)
Start
DSP_Lectures
Microsoft PowerPo...
<Student Version>...
MATLAB Editor/D...
Contacts - Microso...
<Student Versi...
<Student Version>...
<Student Version>...
9:07 PM



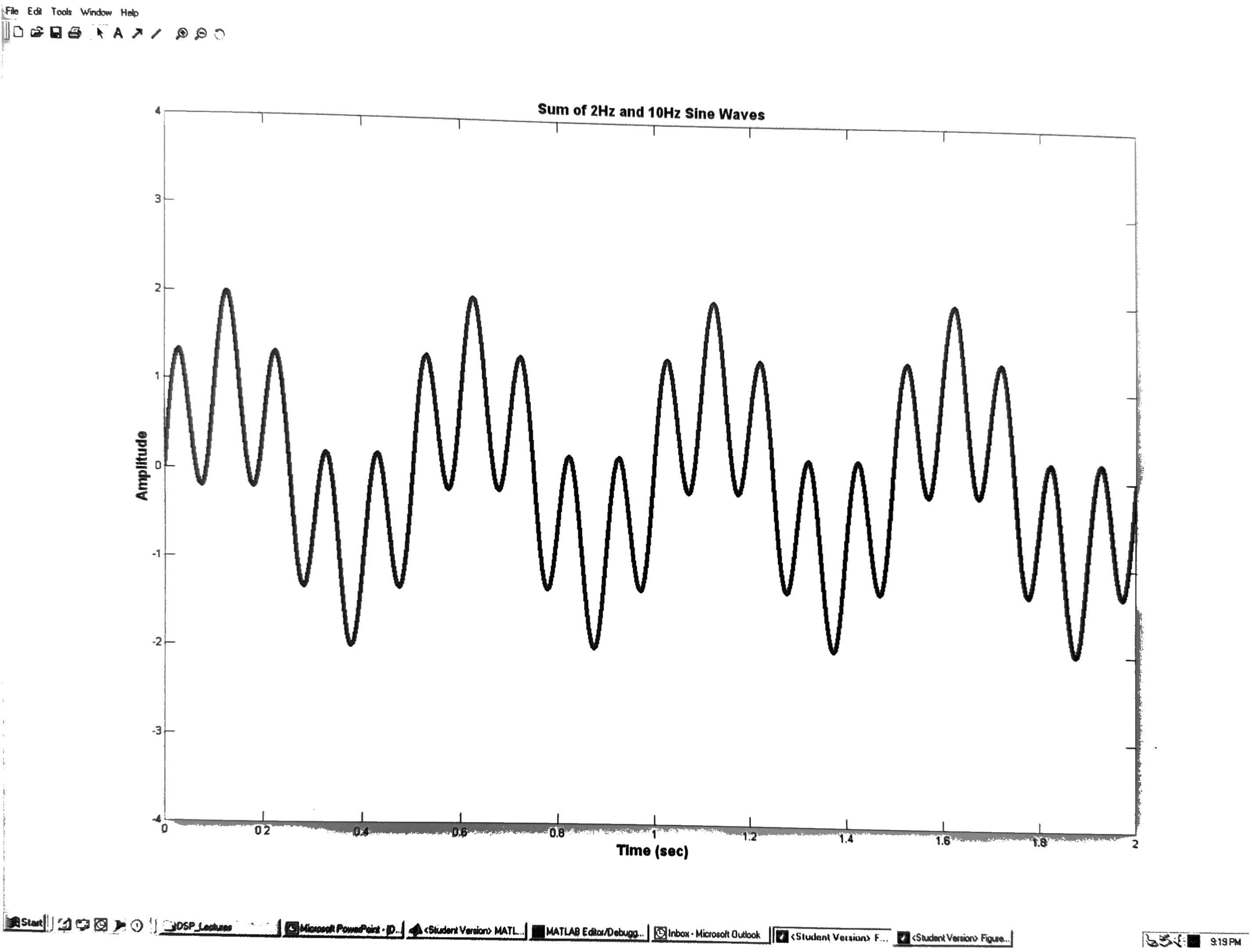
<Student Version> Figure No. 2
File Edit Tools Window Help
FFT of 6 Hz Sine Wave
Magnitude
35
30
25
20
15
10
5
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
Normalized Frequency
Start
DSP Lectures
MATLAB Editor/Debugg...
Inbox - Microsoft Outlook
9:10 PM

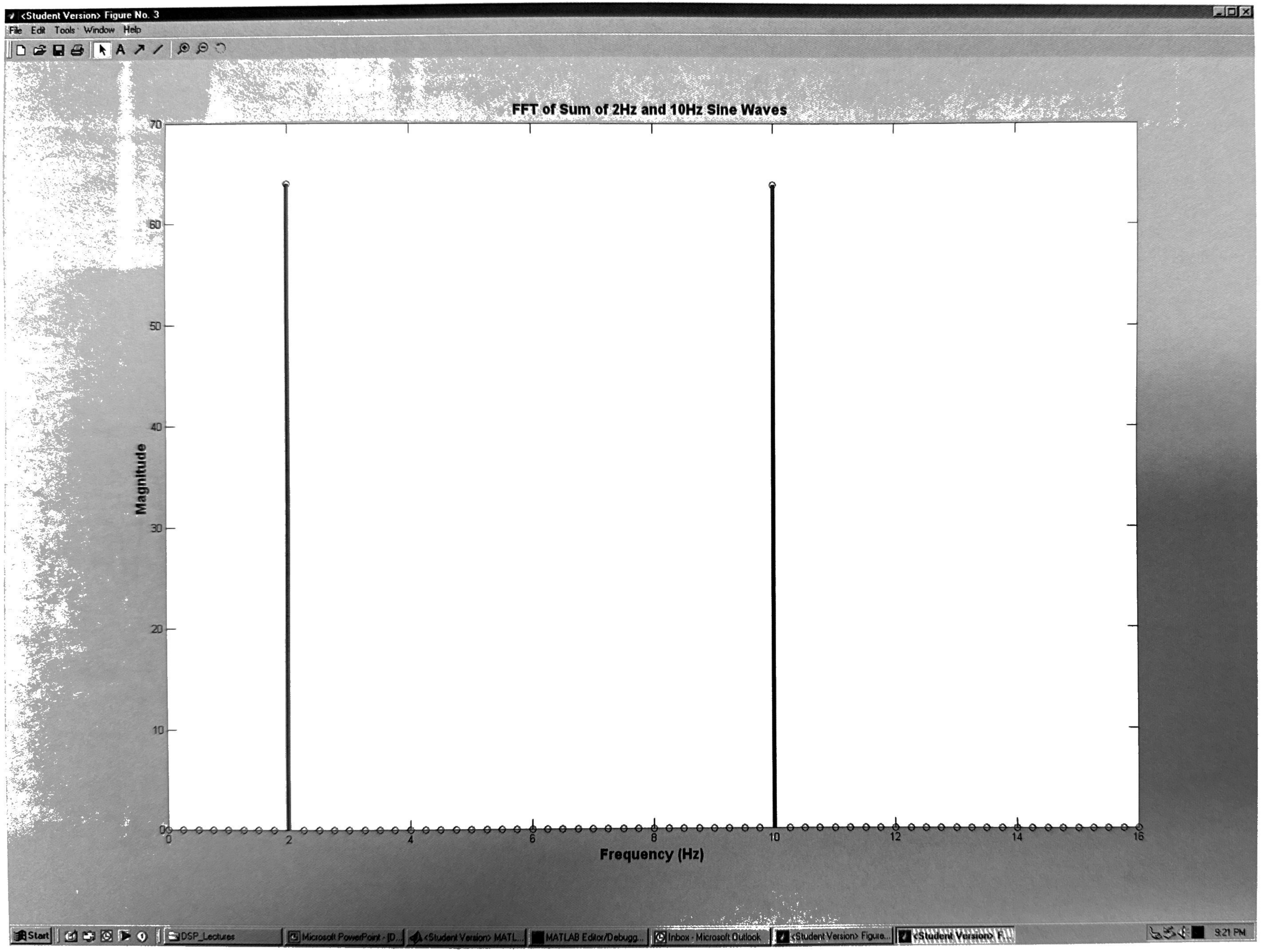


<Student Version> Figure No. 3
File Edit Tools Window Help
FFT of 6Hz Sine Wave
Magnitude
Frequency (Hz)
0 1 2 3 4 5 6 7 8
0 5 10 15 20 25 30 35
Start
DSP_Lectures
Microsoft PowerPoint - [DS...
<Student Version> MATLA...
MATLAB Editor/Debugger...
Inbox - Microsoft Outlook
<Student Version> Fi...
9:17 PM

Sine Wave Examples

- FFT of sum of sine waves at 2 Hz and 10 Hz
  - 1 nonzero component at 2 Hz.
  - 1 nonzero component at 10 Hz


File Edit Tools Window Help
Sum of 2Hz and 10Hz Sine Waves
Amplitude
Time (sec)
4
3
2
1
0
-1
-2
-3
-4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Start
DSP_Lectures
Microsoft PowerPoint - [D...
<Student Version> MATL...
MATLAB Editor/Debugg...
Inbox - Microsoft Outlook
<Student Version> F...
<Student Version> Figure...
9:19 PM


<Student Version> Figure No. 3
File Edit Tools Window Help
FFT of Sum of 2Hz and 10Hz Sine Waves
Magnitude
Frequency (Hz)
0
10
20
30
40
50
60
70
0
2
4
6
8
10
12
14
16
Start
DSP_Lectures
Microsoft PowerPoint - [D...
<Student Version> MATL...
MATLAB Editor/Debugg...
Inbox - Microsoft Outlook
<Student Version> Figure...
<Student Version> F...
9:21 PM

An Application Of Fourier Analysis

- The frequency composition of a signal given by Fourier Analysis is very useful for noise reduction.

- What are some examples of noise?

  – Additive Noise: Line Noise

    Observed Signal = Desired Signal + Noise Process

  – Convolutional Distortion: Communication Channel Noise

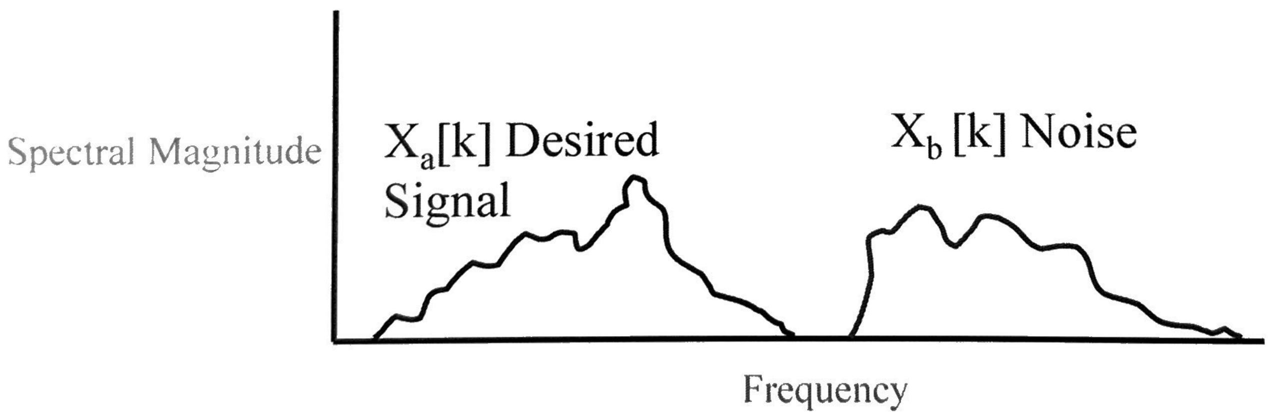
    Observed Signal = Desired Signal * Channel Impulse Response

How Fourier Analysis Helps With Additive Noise

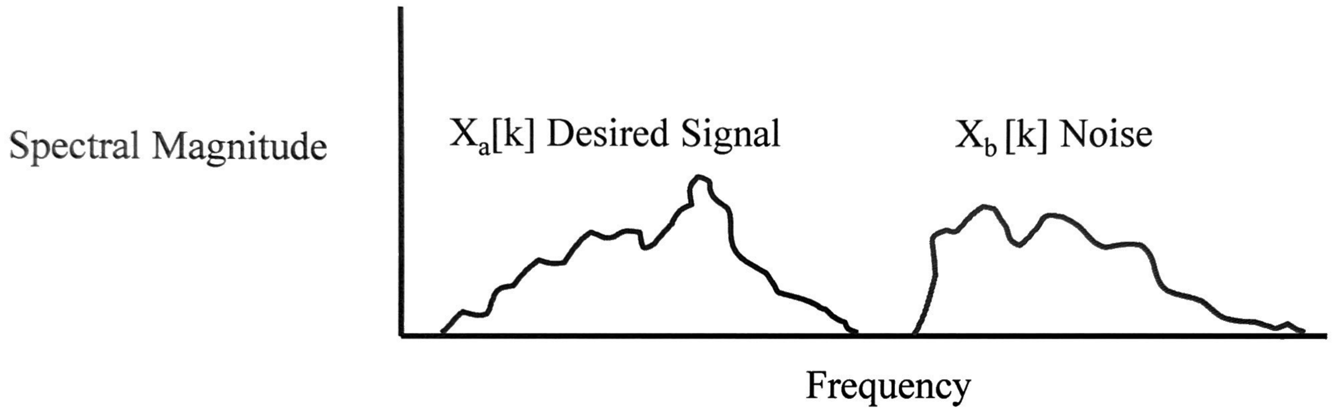
- DFT of the sum of two signals **equals** the sum of the individual DFTs:

$$x_a[n] + x_b[n] \rightarrow X_a[k] + X_b\,[k]$$

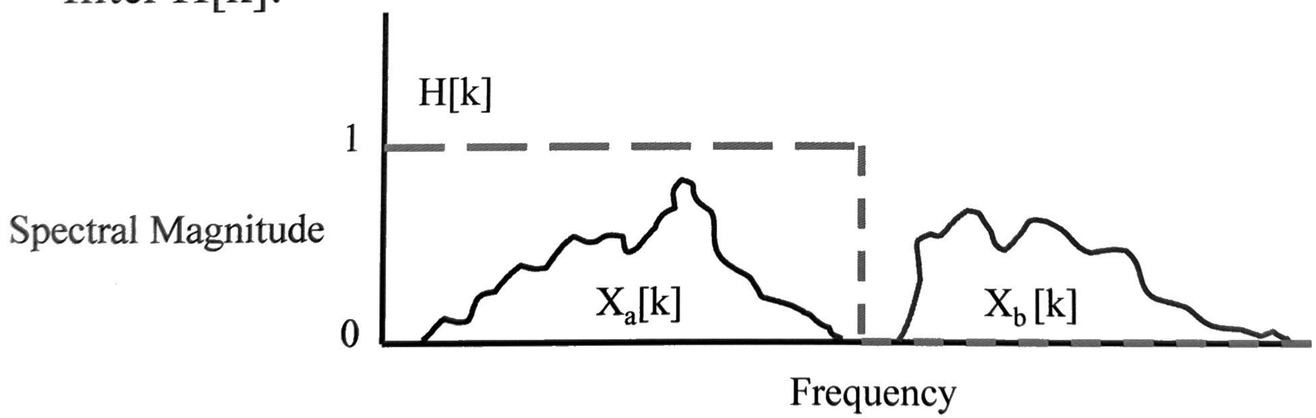
- We get lucky if $x_a[n]$ (desired signal) and $x_b[n]$ (noise) are composed of different frequencies!



How Fourier Analysis Helps



- To recover $X_a[k]$ from X_b [k], we multiply the above spectrum with a filter H[k]:



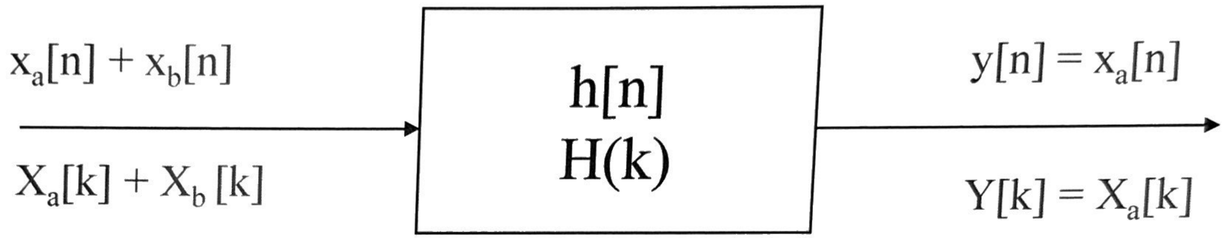
Multiplication in Frequency

- We just saw that multiplication in the frequency domain allowed us to recover $X_a[k]$ from $X_b[k]$.

- Multiplication in Frequency = Convolution in Time
- Recall that our windows of data x[n] came from an imagined periodic or repeated version $\tilde{x}[n]$. So, the convolution operation implied by multiplying DFT's is a "circular convolution":

$$X[k]\,H[k] \rightarrow \sum_{m=0}^{N-1} \tilde{x}[m]\,\tilde{h}[n-m] \quad (\text{for } 0 <= n < N-1)$$

Additive Noise Reduction
Block Diagram



Frequency Domain

(multiplication)

$$Y(k) = H[k]\, X_a[k] + \cancel{H[k]\, X_b[k]}$$

These terms would zero if the filter perfectly eliminates the noise.

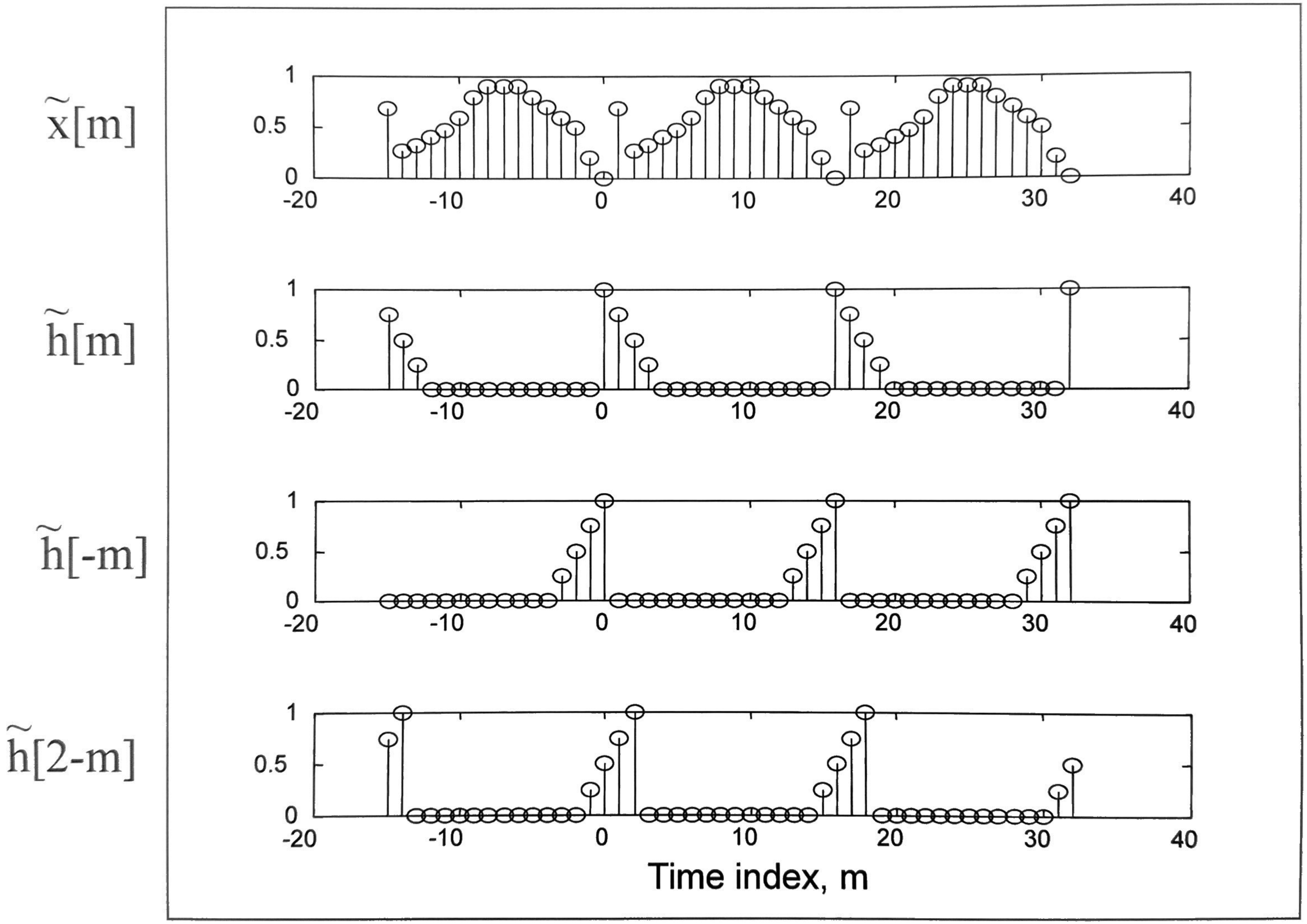
Time Domain

(circular convolution)

$$y[n] = \sum_m \tilde{x}_a[m]\, \tilde{h}(n-m) + \cancel{\sum_m \tilde{x}_b[m]\, \tilde{h}(n-m)}$$

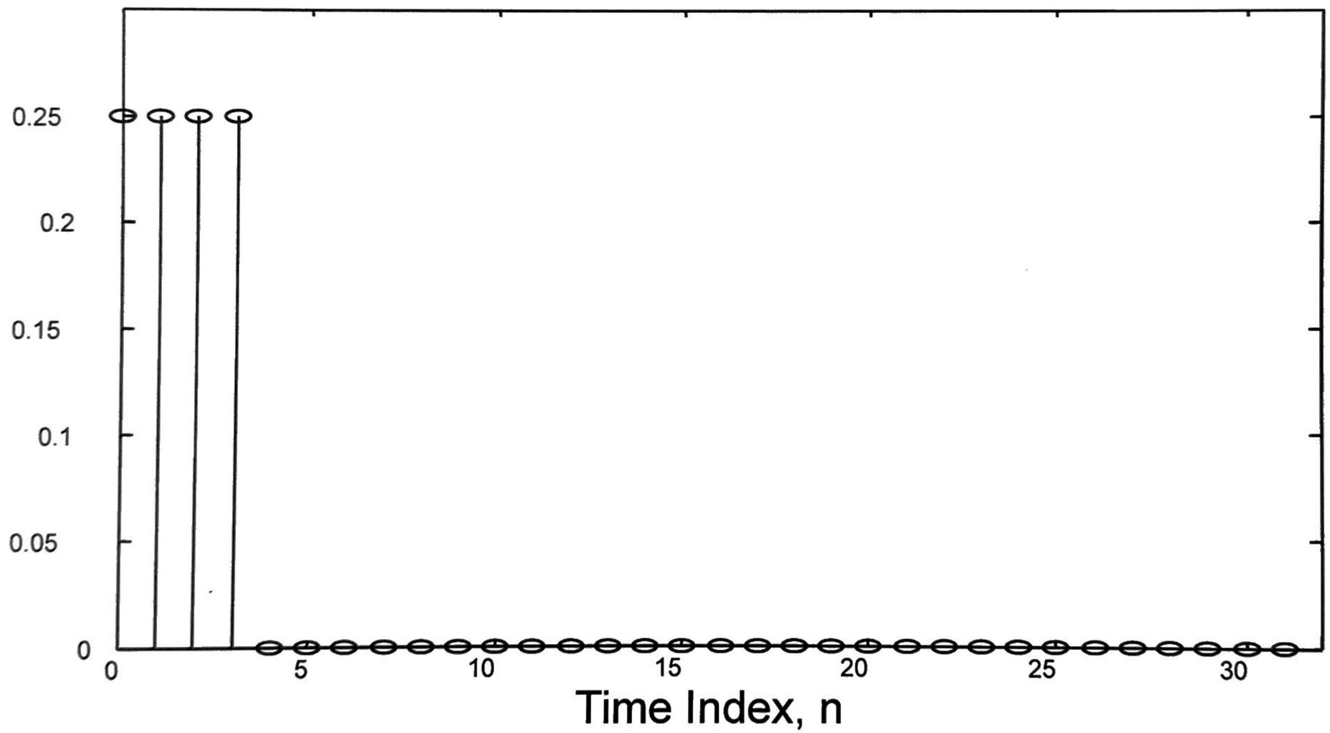
for a period, e.g., $0 <= n < N$

Components for Circular Convolution

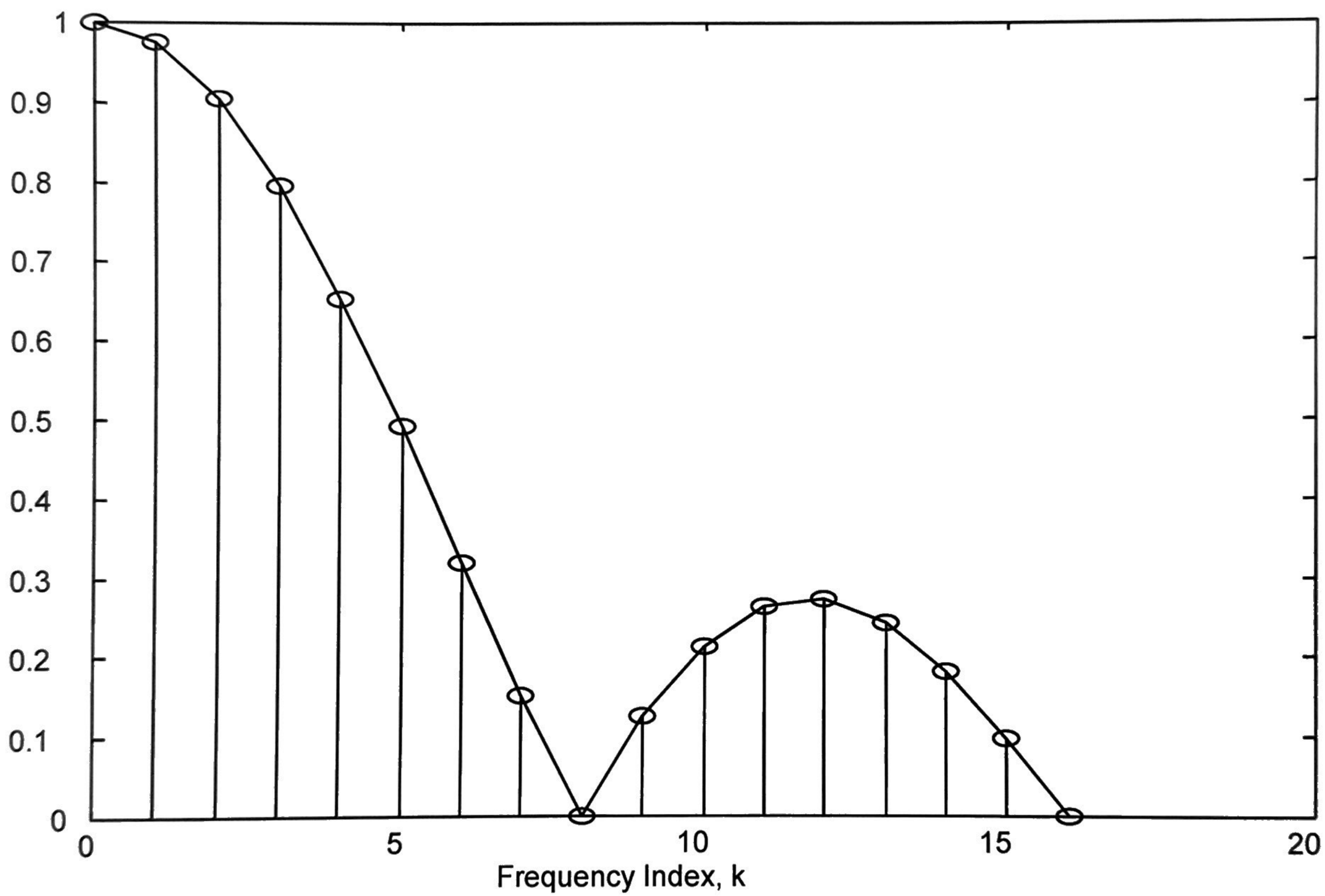


Easy Example: Low-Pass Box

Suppose we convolve a signal in time with this:



In frequency, that's like multiplying by this:



Box and Sinc are Duals!

•Convolving with a box gives a low-pass filter effect, but not a very good low pass filter effect!

•Convolving with a Sinc function in time gives a perfect low-pass box in frequency, but a filter with a global impulse response.

•Usually, we pick a compromise – neither compactly supported in frequency or in time!

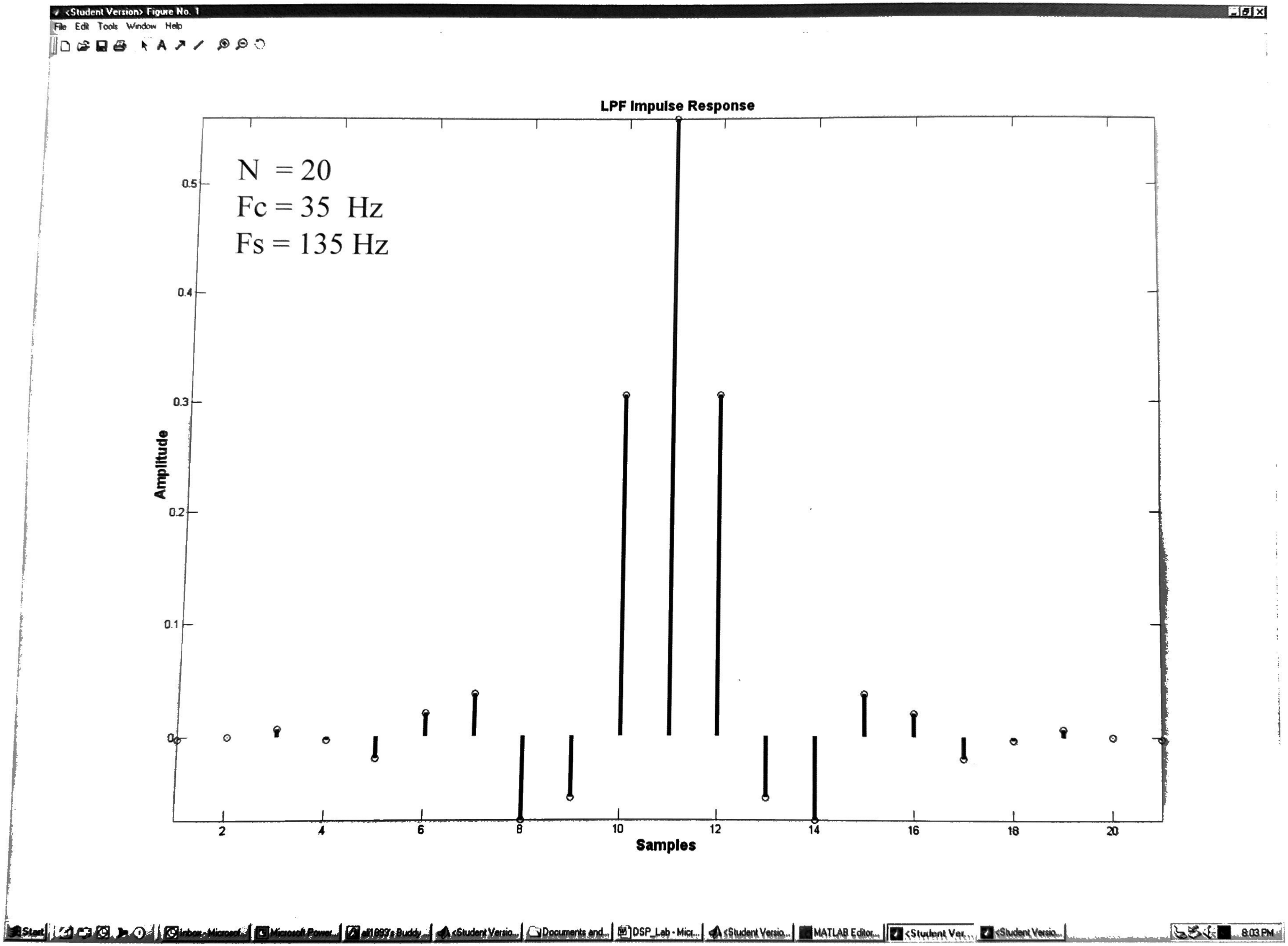
FIR Filters

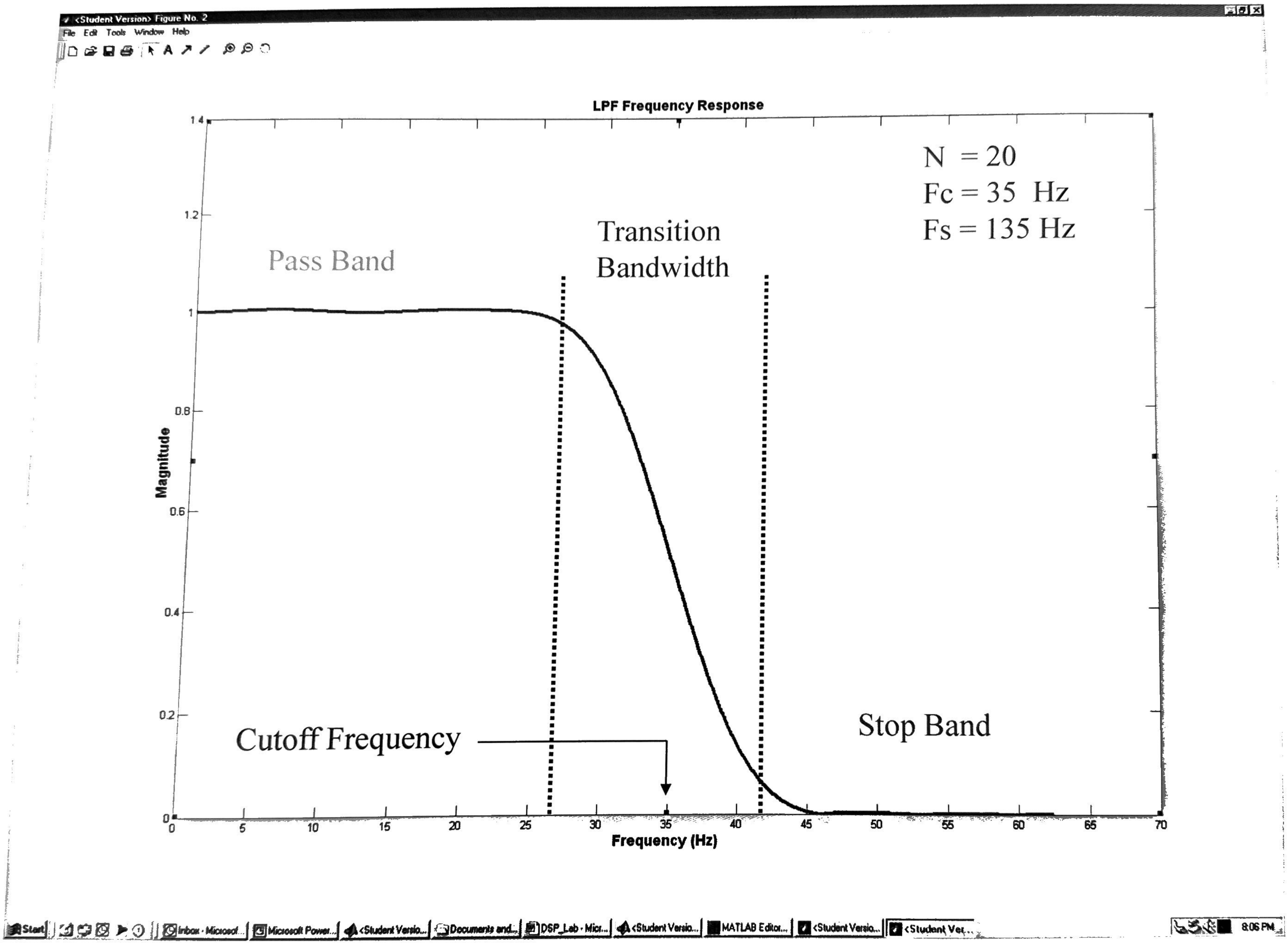
- Finite Impulse Response Filters are LTI systems that selectivly modify the frequency spectrum of an input signal.

- Design an FIR Low Pass Filter in Matlab using fir1

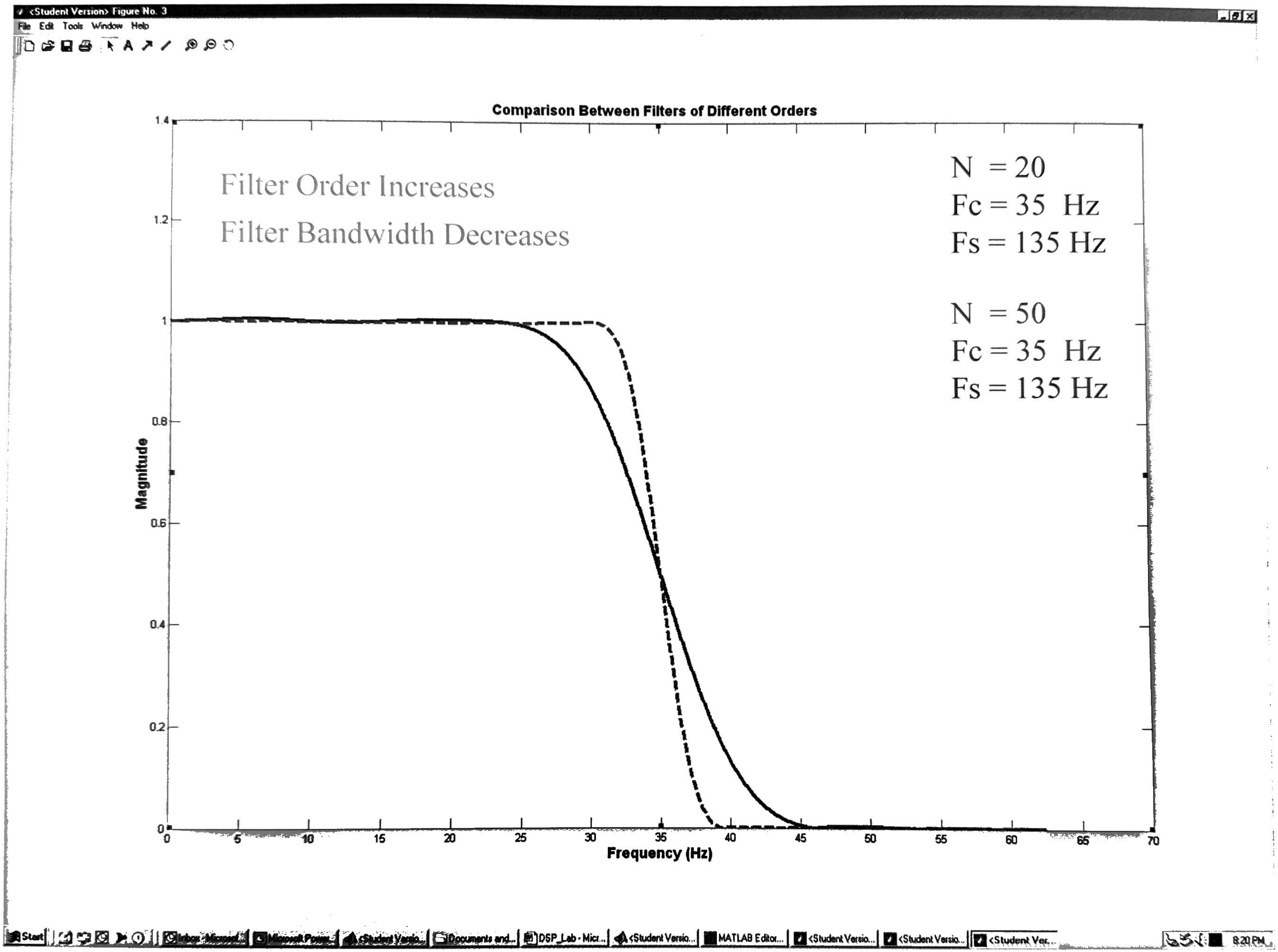
```
h = fir1(N, 2*(Fc/Fs));
```

Impulse Response   Filter Order   Cutoff Frequency


<Student Version> Figure No. 1
File Edit Tools Window Help
LPF Impulse Response
N = 20
Fc = 35 Hz
Fs = 135 Hz
Amplitude
0.5
0.4
0.3
0.2
0.1
0
2 4 6 8 10 12 14 16 18 20
Samples
Start
MATLAB Editor...
8:03 PM



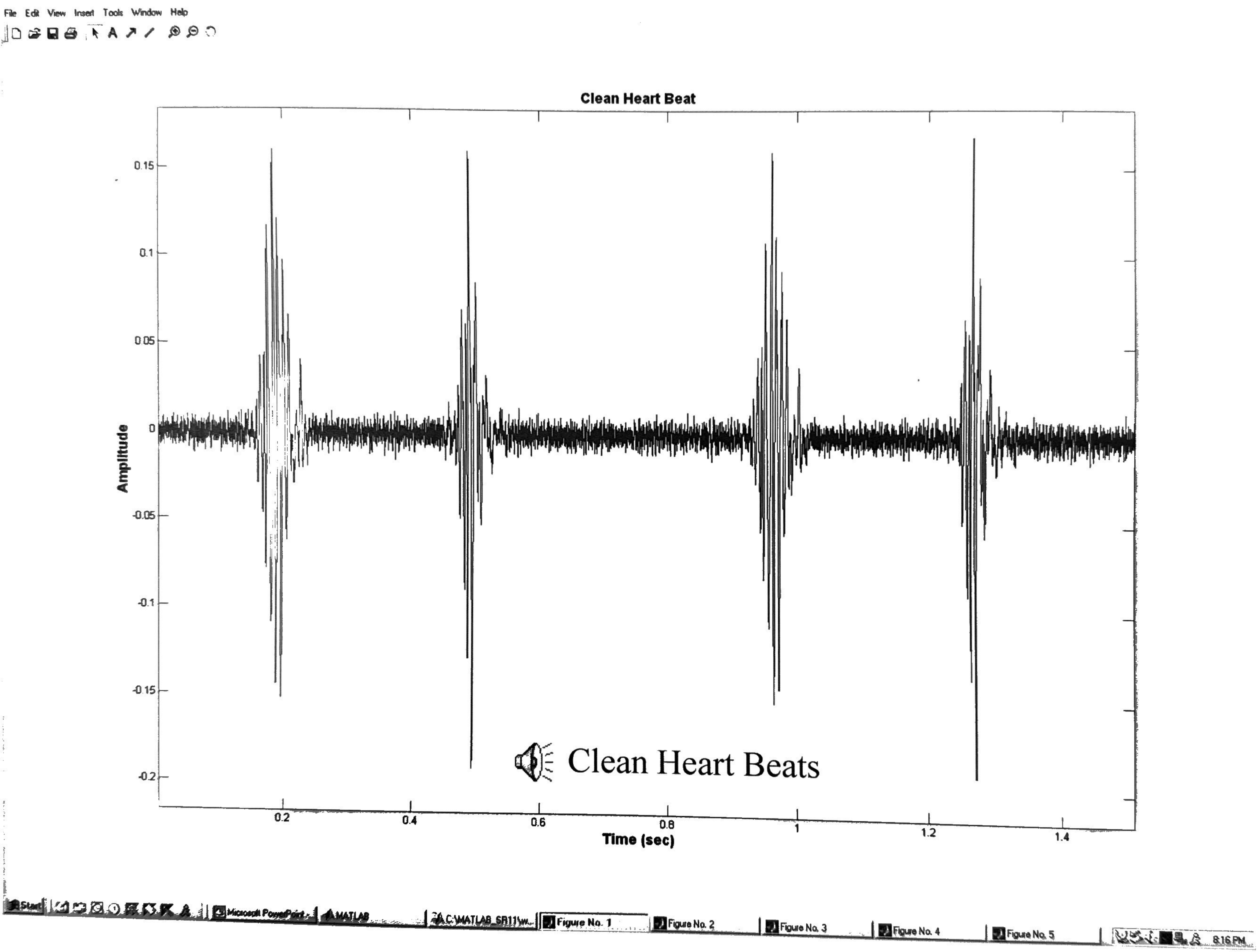
<Student Version> Figure No. 2
File Edit Tools Window Help
LPF Frequency Response
N = 20
Fc = 35 Hz
Fs = 135 Hz
Pass Band
Transition
Bandwidth
Stop Band
Cutoff Frequency
Magnitude
Frequency (Hz)

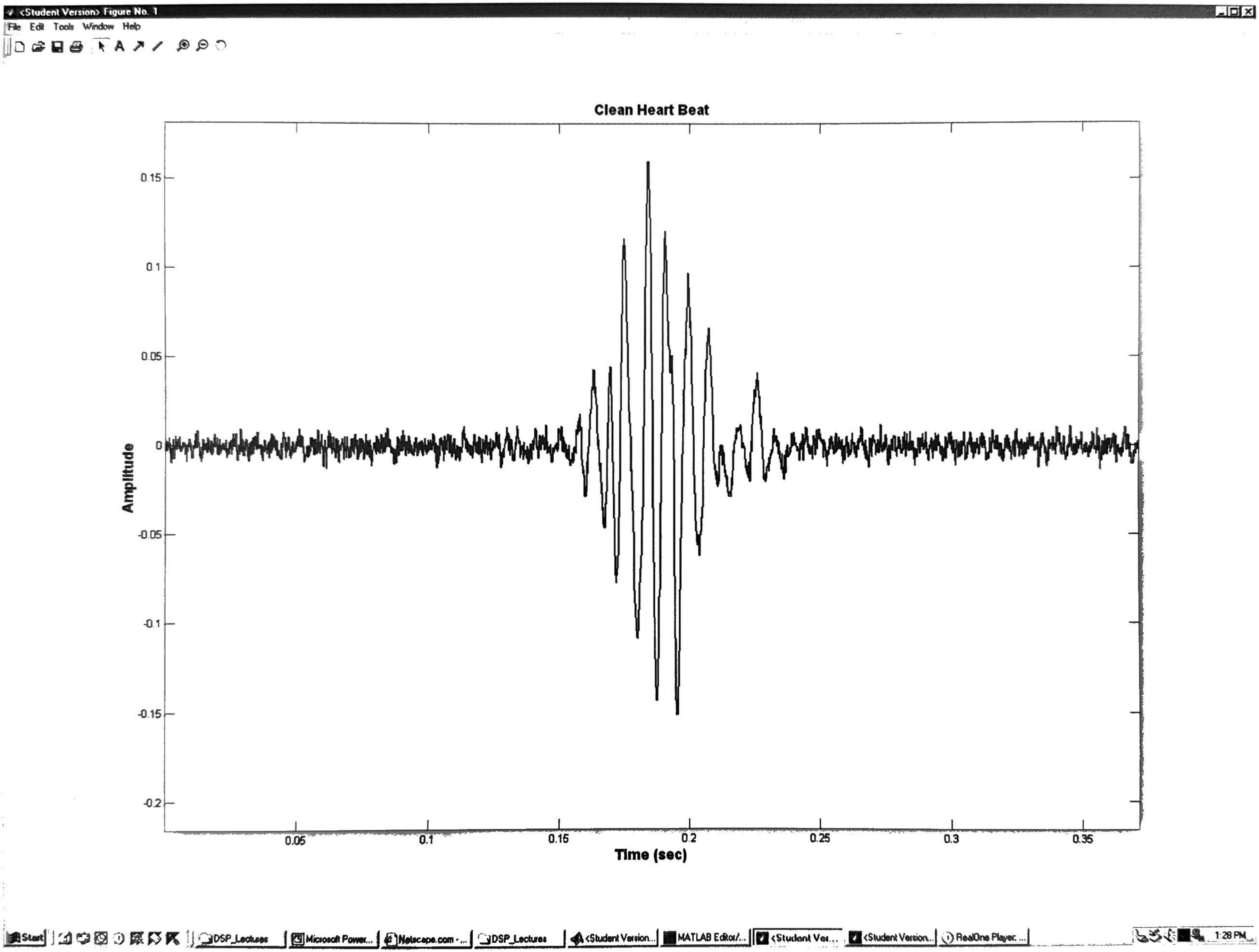


<Student Version> Figure No. 3
File Edit Tools Window Help
Comparison Between Filters of Different Orders
Filter Order Increases
Filter Bandwidth Decreases
N = 20
Fc = 35 Hz
Fs = 135 Hz
N = 50
Fc = 35 Hz
Fs = 135 Hz
Magnitude
Frequency (Hz)

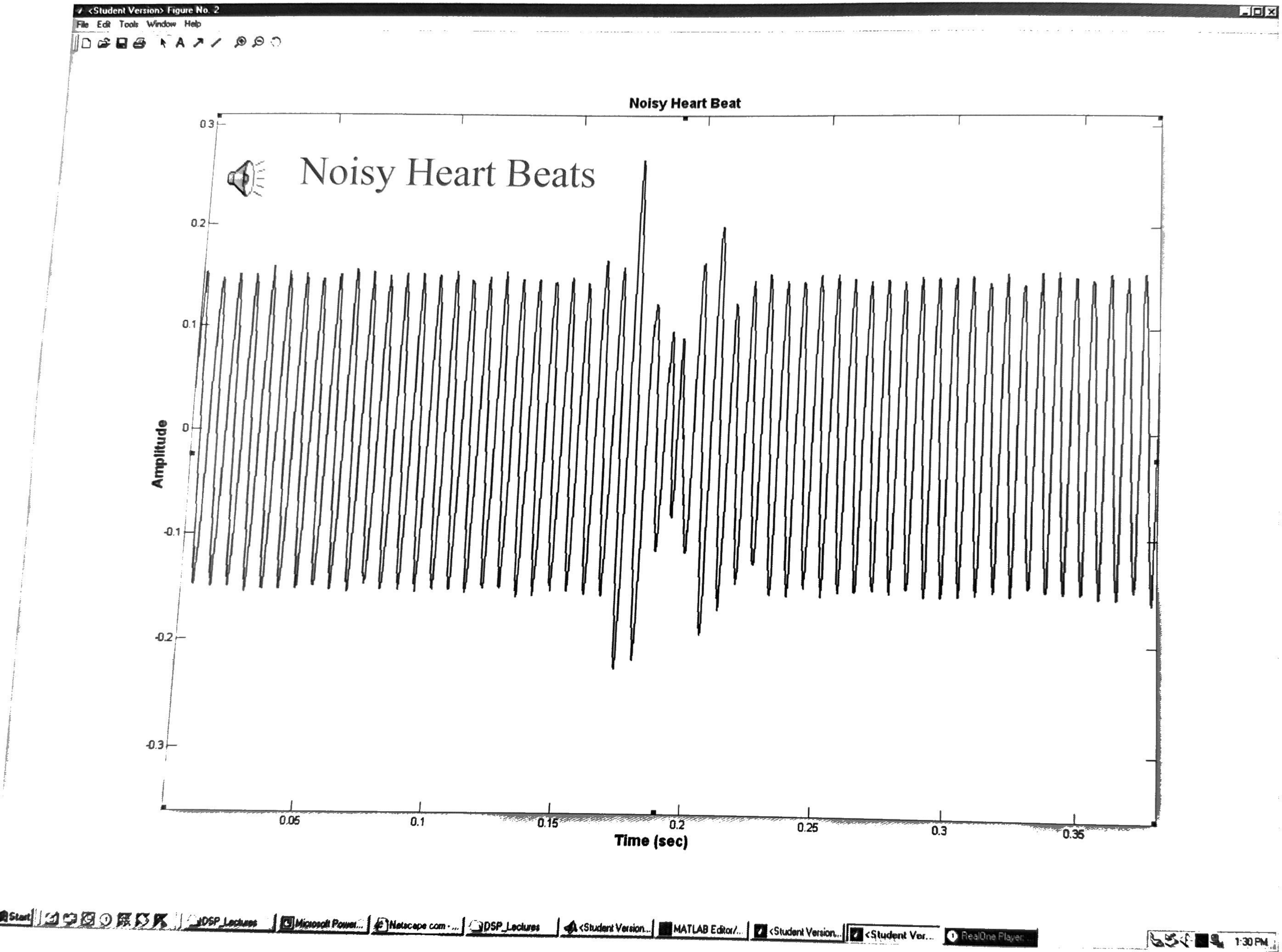
An Idealized Example

- Now we will try to remove noise composed of a single 500 Hz tone from a heart beat.

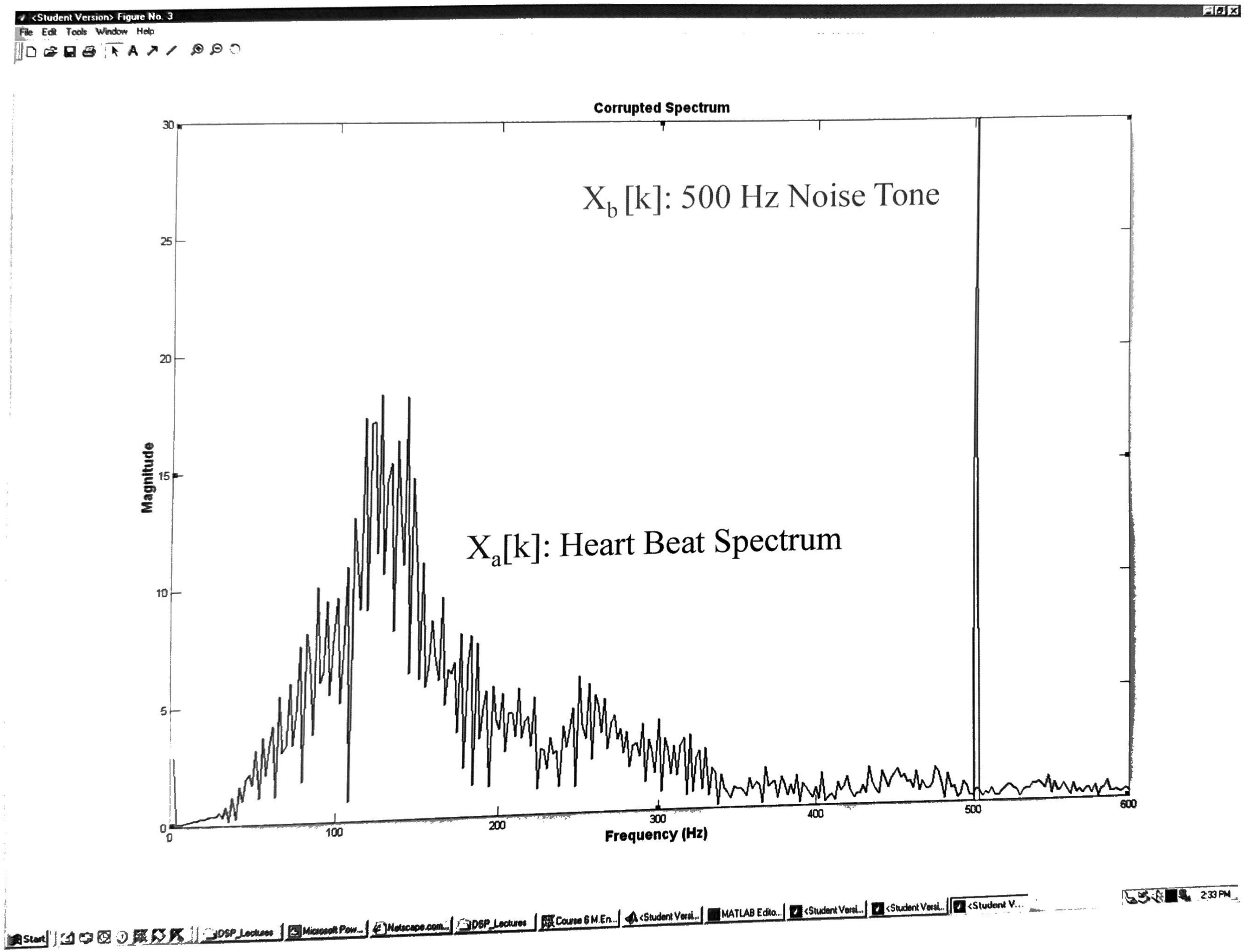

File Edit View Insert Tools Window Help
Clean Heart Beat
Amplitude
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0.2
0.4
0.6
0.8
1
1.2
1.4
Time (sec)
Clean Heart Beats
Start
Microsoft PowerPoint
MATLAB
Figure No. 1
Figure No. 2
Figure No. 3
Figure No. 4
Figure No. 5
8:16 PM



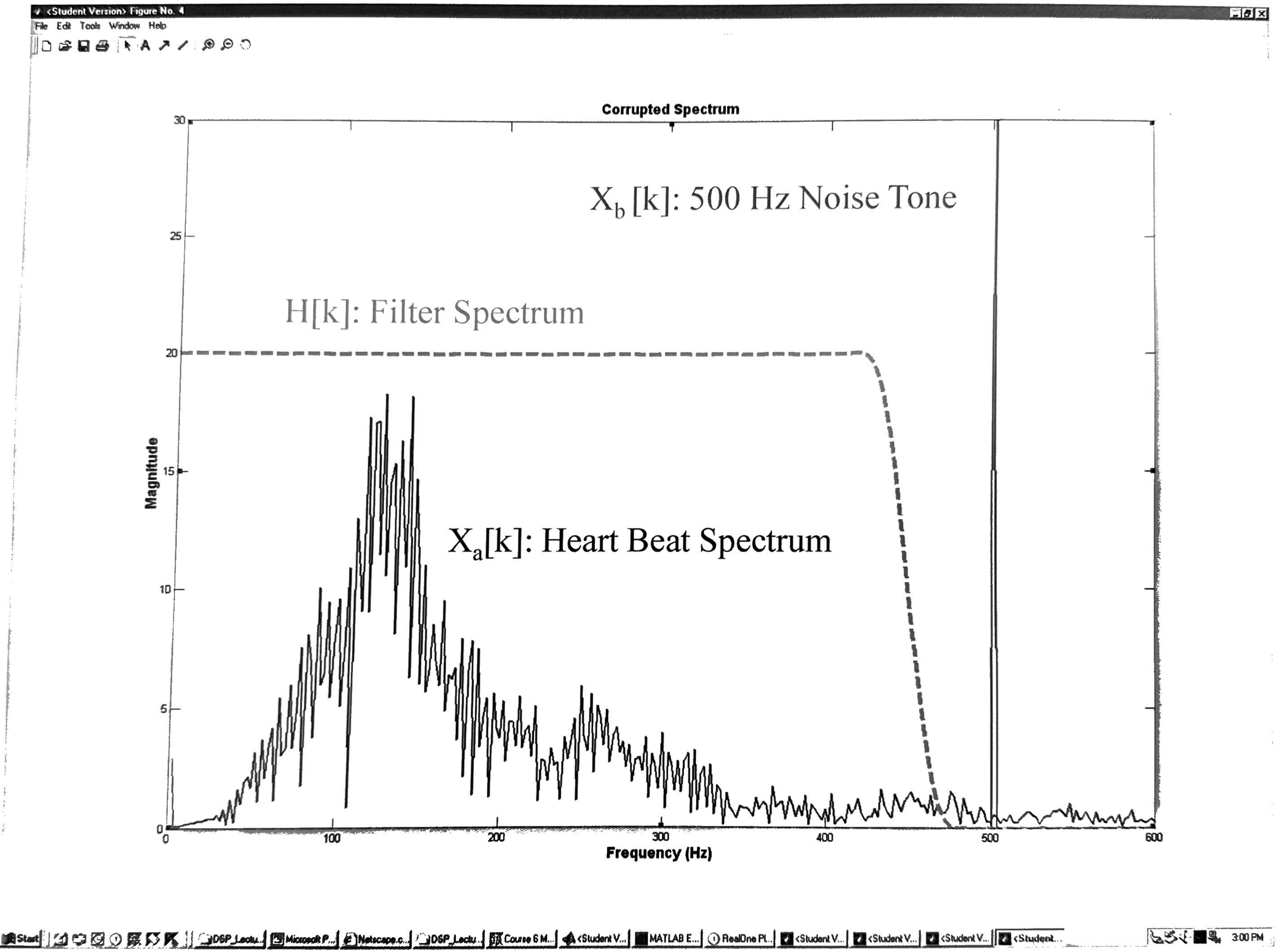
<Student Version> Figure No. 1
File Edit Tools Window Help
Clean Heart Beat
Amplitude
Time (sec)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Start
DSP_Lectures
Microsoft Power...
Netscape.com - ...
DSP_Lectures
<Student Version...
MATLAB Editor/...
<Student Ver...
<Student Version...
RealOne Player: ...
1:28 PM

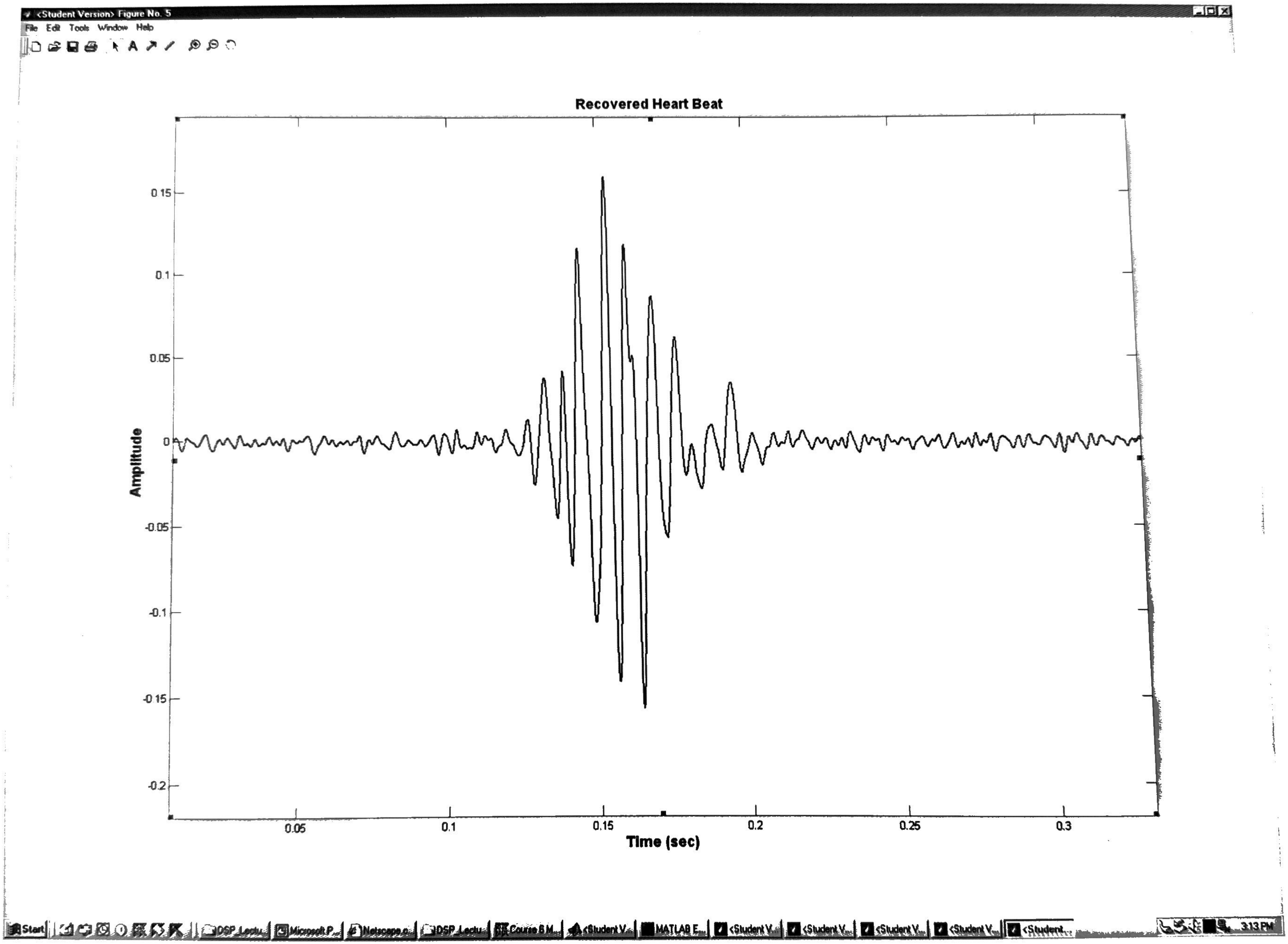


<Student Version> Figure No. 2
File Edit Tools Window Help
Noisy Heart Beat
Noisy Heart Beats
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
Amplitude
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Time (sec)
Start
DSP_Lectures
Microsoft Power...
Netscape.com - ...
DSP_Lectures
<Student Version...
MATLAB Editor/...
<Student Version...
<Student Ver...
RealOne Player...
1:30 PM

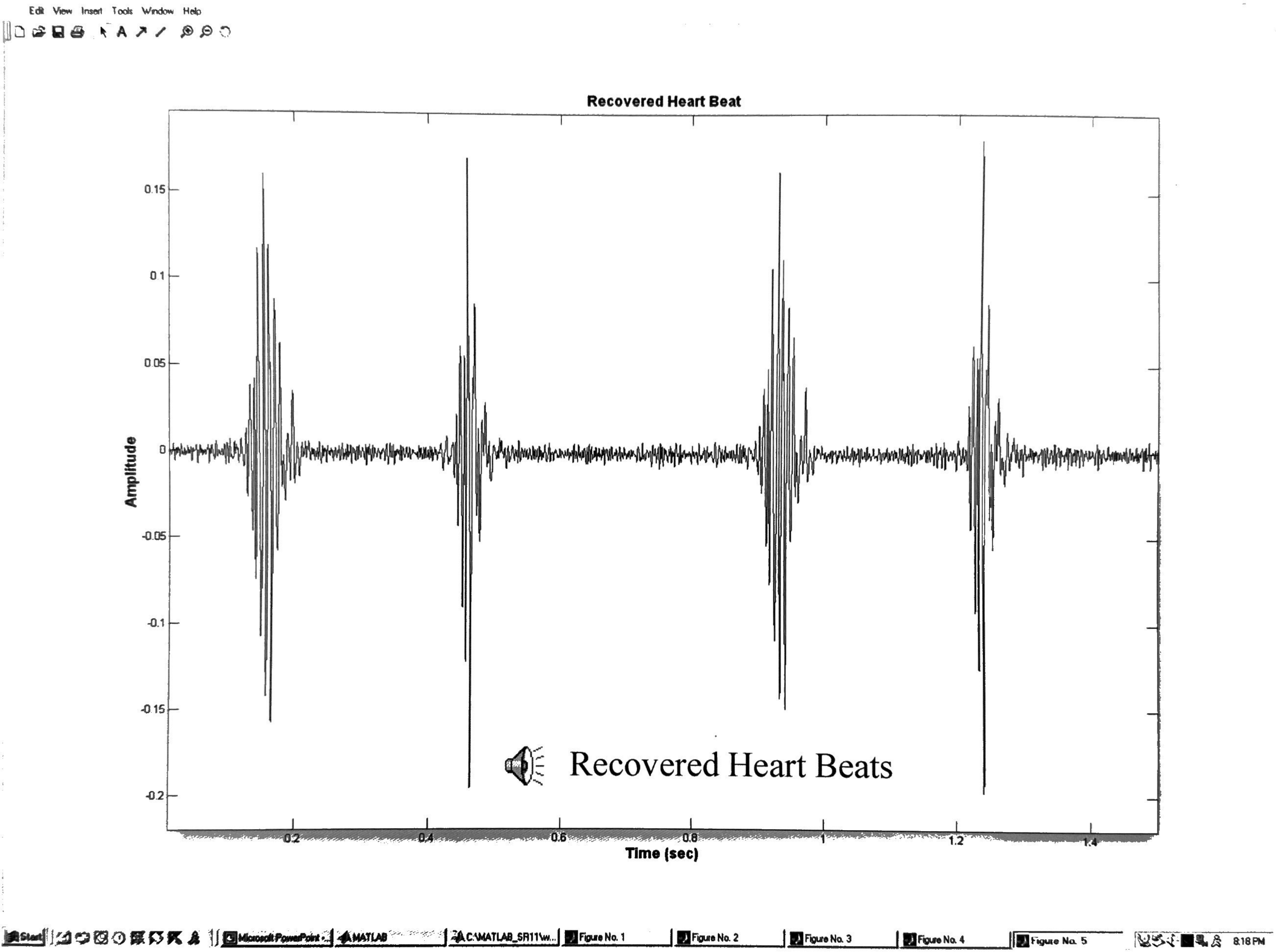


<Student Version> Figure No. 3
File Edit Tools Window Help
Corrupted Spectrum
X_b [k]: 500 Hz Noise Tone
X_a[k]: Heart Beat Spectrum
Magnitude
Frequency (Hz)
0
5
10
15
20
25
30
100
200
300
400
500
600
Start
DSP_Lectures
Microsoft Pow...
Netscape.com...
Course 6 M.En...
MATLAB Edito...
2:33 PM


<Student Version> Figure No. 4
File Edit Tools Window Help
Corrupted Spectrum
X_b [k]: 500 Hz Noise Tone
H[k]: Filter Spectrum
X_a[k]: Heart Beat Spectrum
Magnitude
Frequency (Hz)
0
5
10
15
20
25
30
0
100
200
300
400
500
600
Start
DSP_Lectu...
Microsoft P...
Netscape.c...
DSP_Lectu...
Course 6 M...
<Student V...
MATLAB E...
RealOne Pl...
<Student V...
<Student V...
<Student V...
<Student...
3:00 PM



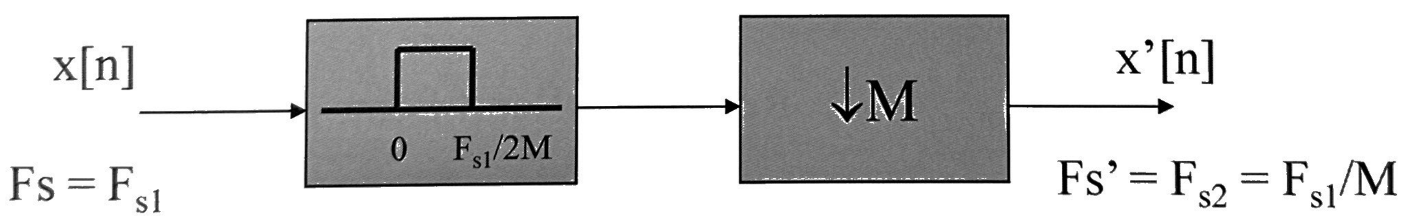
<Student Version> Figure No. 5
File Edit Tools Window Help
Recovered Heart Beat
Amplitude
Time (sec)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0.05
0.1
0.15
0.2
0.25
0.3
Start
MATLAB E...
3:13 PM



Edit View Insert Tools Window Help
Recovered Heart Beat
Amplitude
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0.2
0.4
0.6
0.8
1
1.2
1.4
Time (sec)
Recovered Heart Beats
Start
Microsoft PowerPoint
MATLAB
Figure No. 1
Figure No. 2
Figure No. 3
Figure No. 4
Figure No. 5
8:18 PM

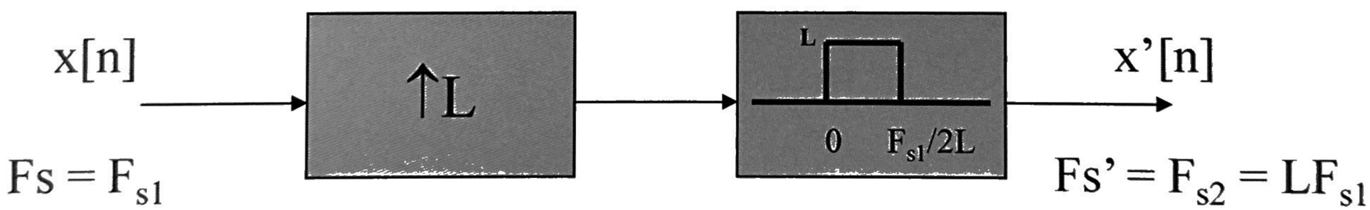
Downsampling by a factor of M

- If a signal has been sampled at a rate F_{s1}, it can be downsampled by a factor of M to a new sampling frequency F_{s2} using the following procedure
  - Band-limit the signal to half the new sampling rate: $F_{s2}/2$
  - Throw out M-1 samples for every sample.



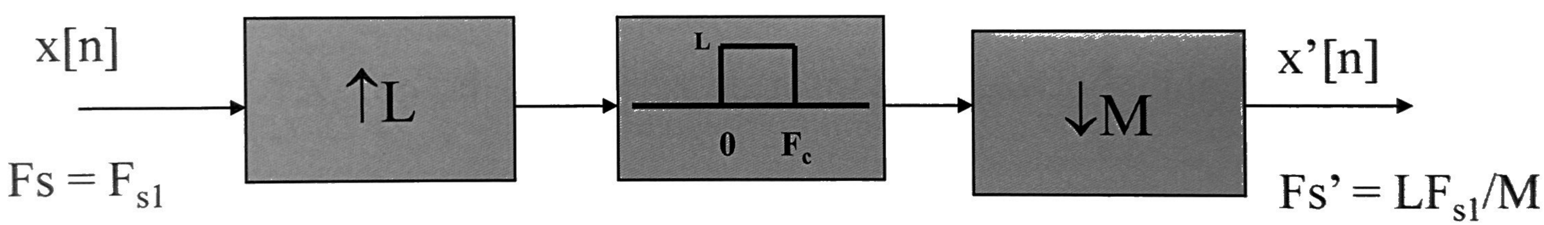
Upsampling by a factor of L

- If a signal has been sampled at a rate F_{s1}, it can be upsampled by a factor of L to a new sampling frequency F_{s2} using the following procedure
  - Insert L-1 zeros after every sample
  - Interpolate between the inserted zeros and the non-zero samples.



Changing the Sampling Frequency by Factor of L/M

- Upsample FIRST by a factor of L.
- Downsample SECOND by a factor of M.



$$F_c = \min \{ F_{s1}/2M , F_{s1}/2L \}$$

Background

Class of discrete, nonlinear filters which incorporate a sorting element, e.g.

The Median Filter

Capable of suppressing impulse noise

Preserves underlying edges

Smooths large outlyers even when signal and noise spectra overlap

Used extensively in image signal processing

Further Considerations

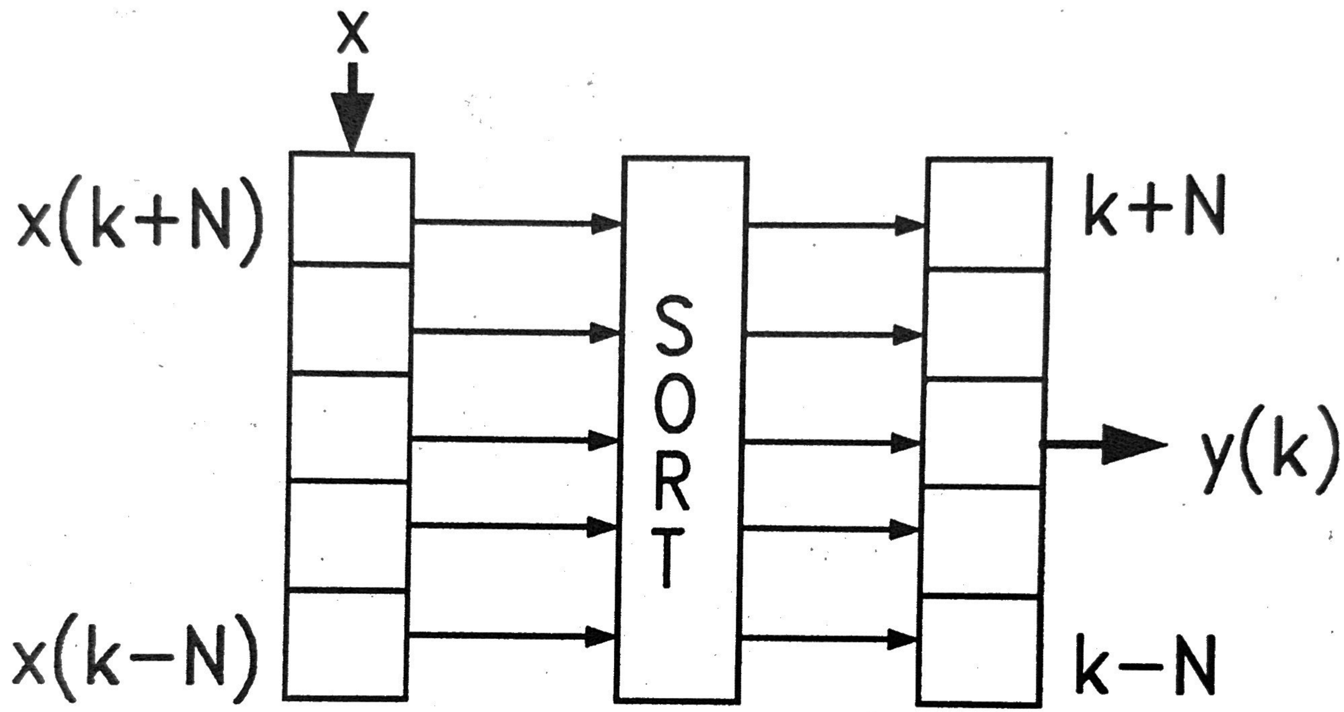
Median and Median–type filters are especially valuable when:

The probability distribution of the noise is unknown or heavily–tailed (e.g. Cauchy noise).

Signal and noise spectra overlap.

The underlying waveform has sharp edges.

The Median Filter

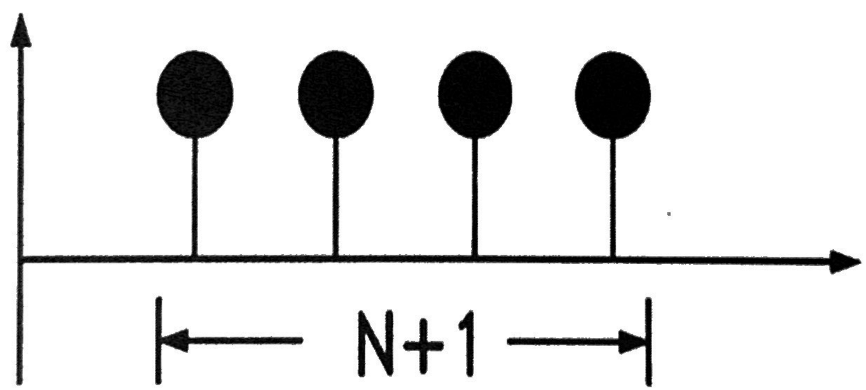


Filter Size	= N (a positive integer)
Window Size	= 2N+1
Delay	= N

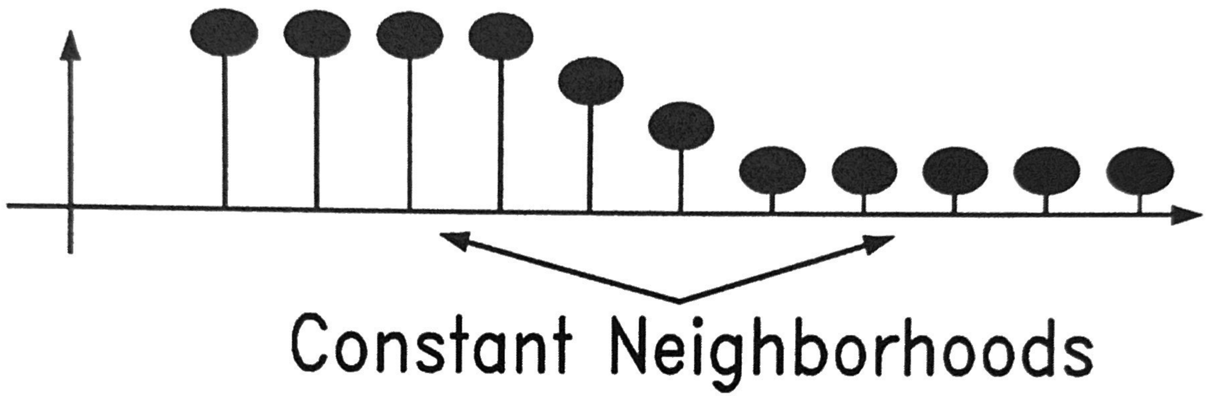
Geometric Analysis of the Median Filter

Definitions (for a filter of size N):

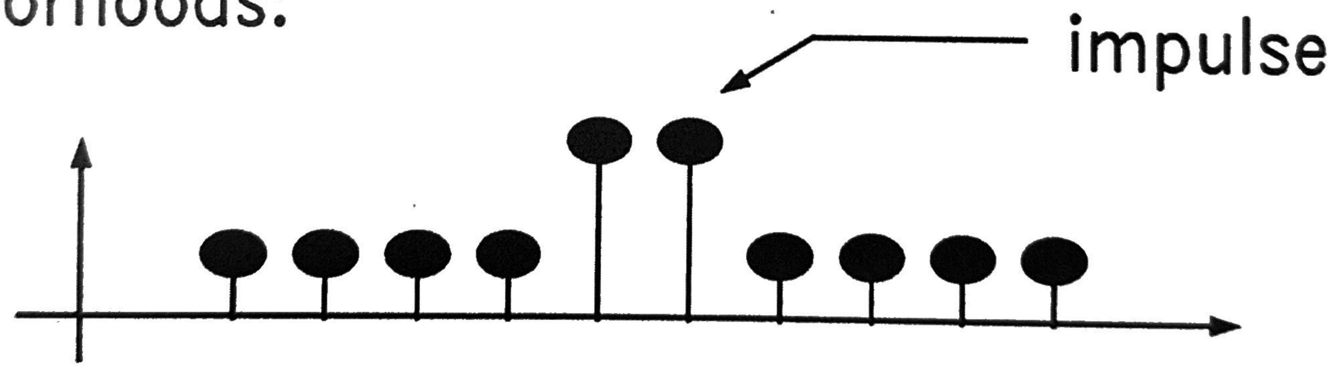
Constant Neighborhood- A section of at least N+1 consecutive, identically valued points:



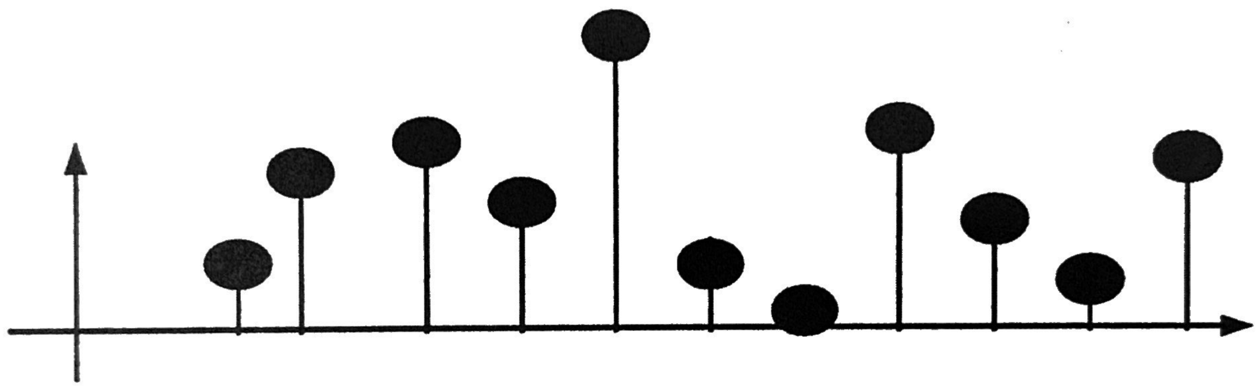
Edge– A monotonically rising or falling region between two constant neigborhoods:



Impulse– A section of one to N points surrounded by identically valued constant neighborhoods:



Oscillation– Any section that is not part of a constant neighborhood, an edge, or an impulse:



Root– A signal unaffected by a median filter of size N.

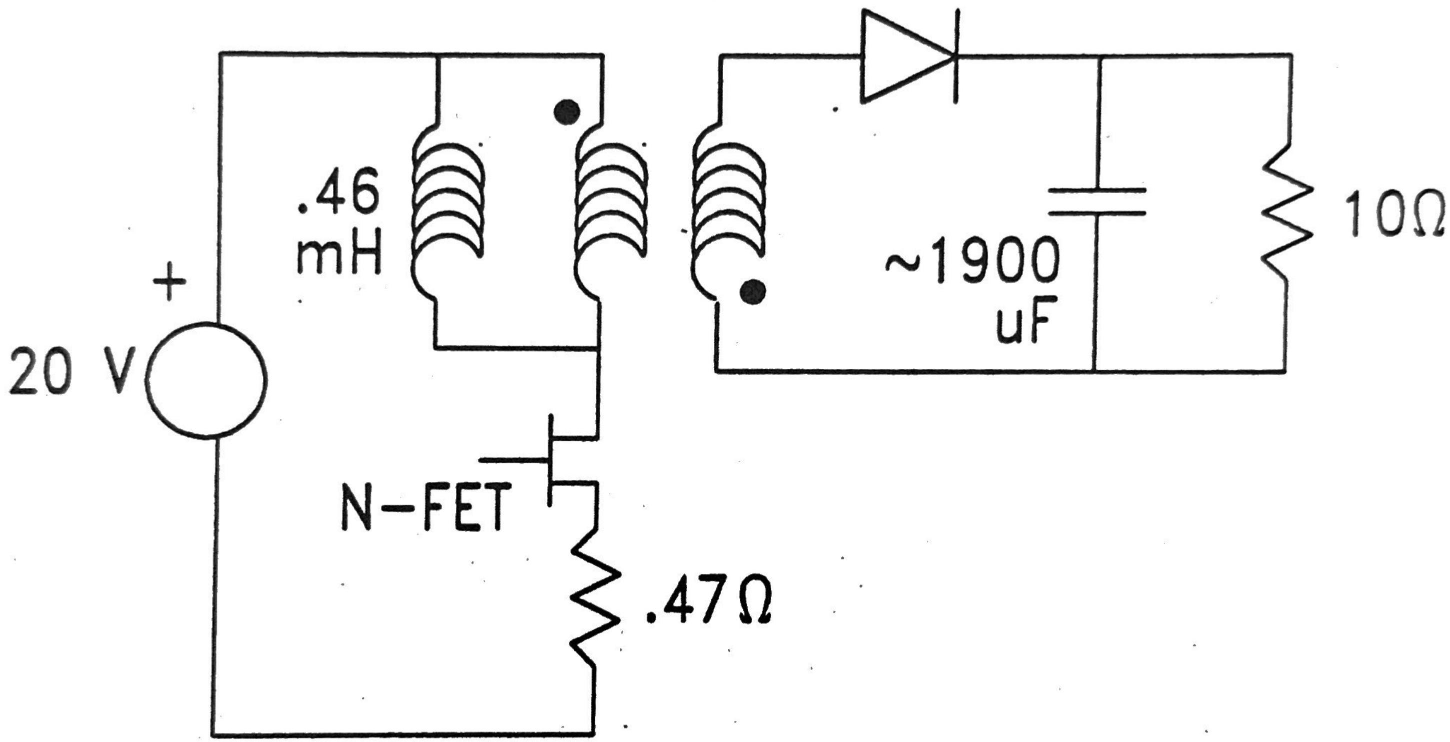
Some Properties of the Median Filter

Impulses are eliminated after a single pass of the median filter.

A root signal consists only of edges and constant neighborhoods.

In a root signal, increasing and decreasing regions must be separated by a constant neighborhood (at least $N+1$ constant points, *the geometric bandwidth*).

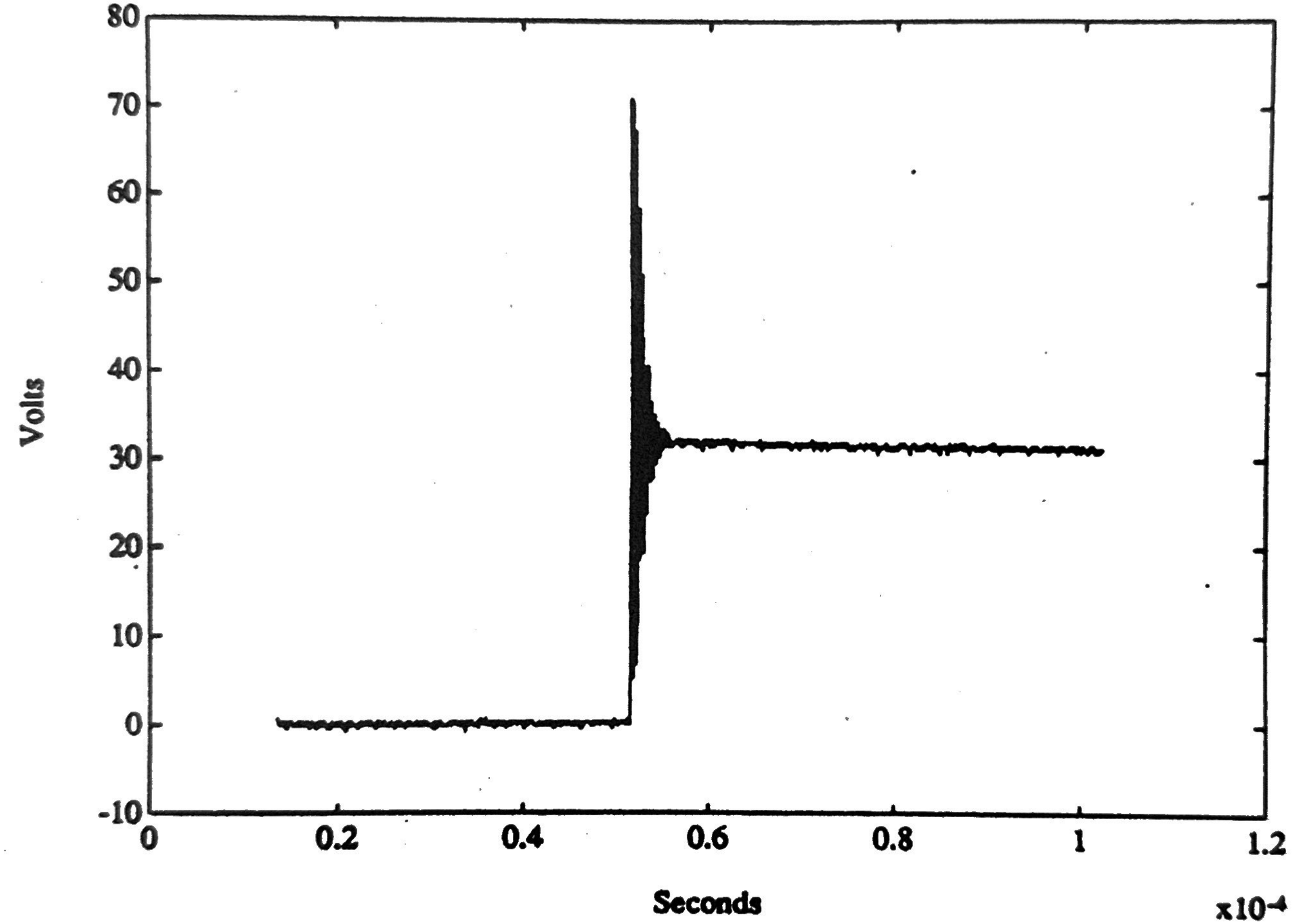
Flyback Converter



Switching Frequency = 5kHz

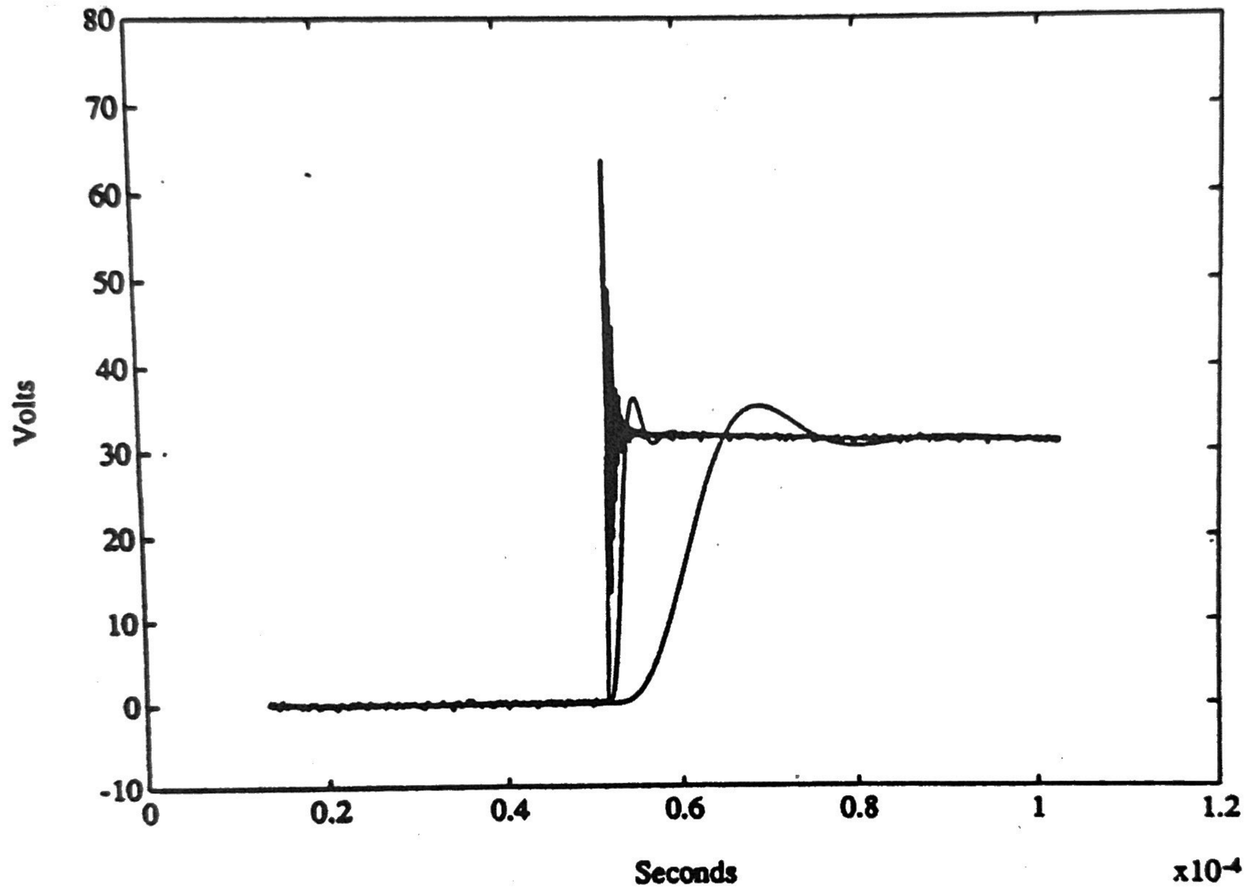
Output Power = 10 Watts

Switch Voltage During Turn-Off



Sample Rate = 10 MHz

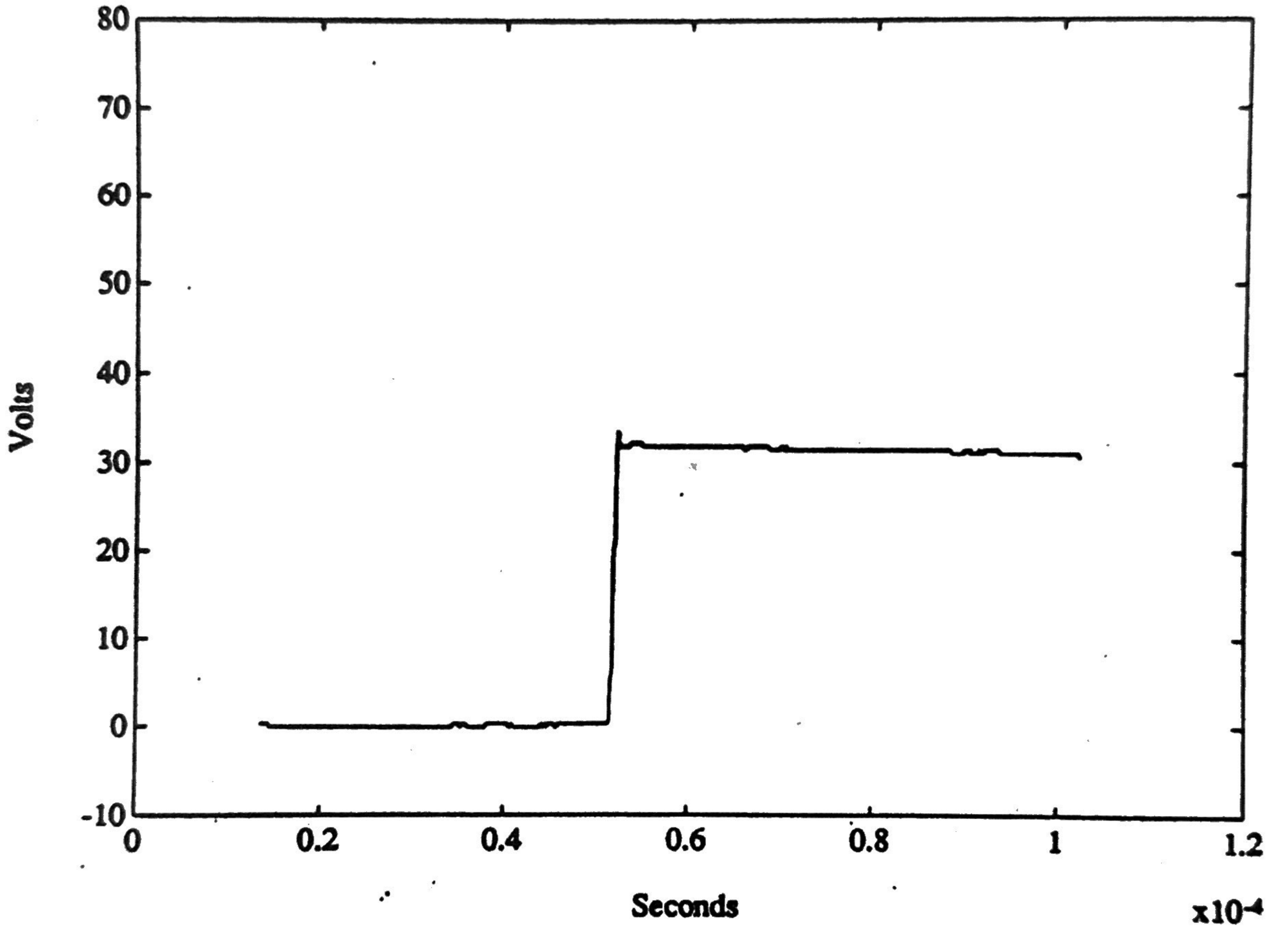
Lowpass Filtered Switch Voltage



Fourth-Order Butterworth Filters with cutoff frequencies at 0.5w, 0.05w, 0.01w.

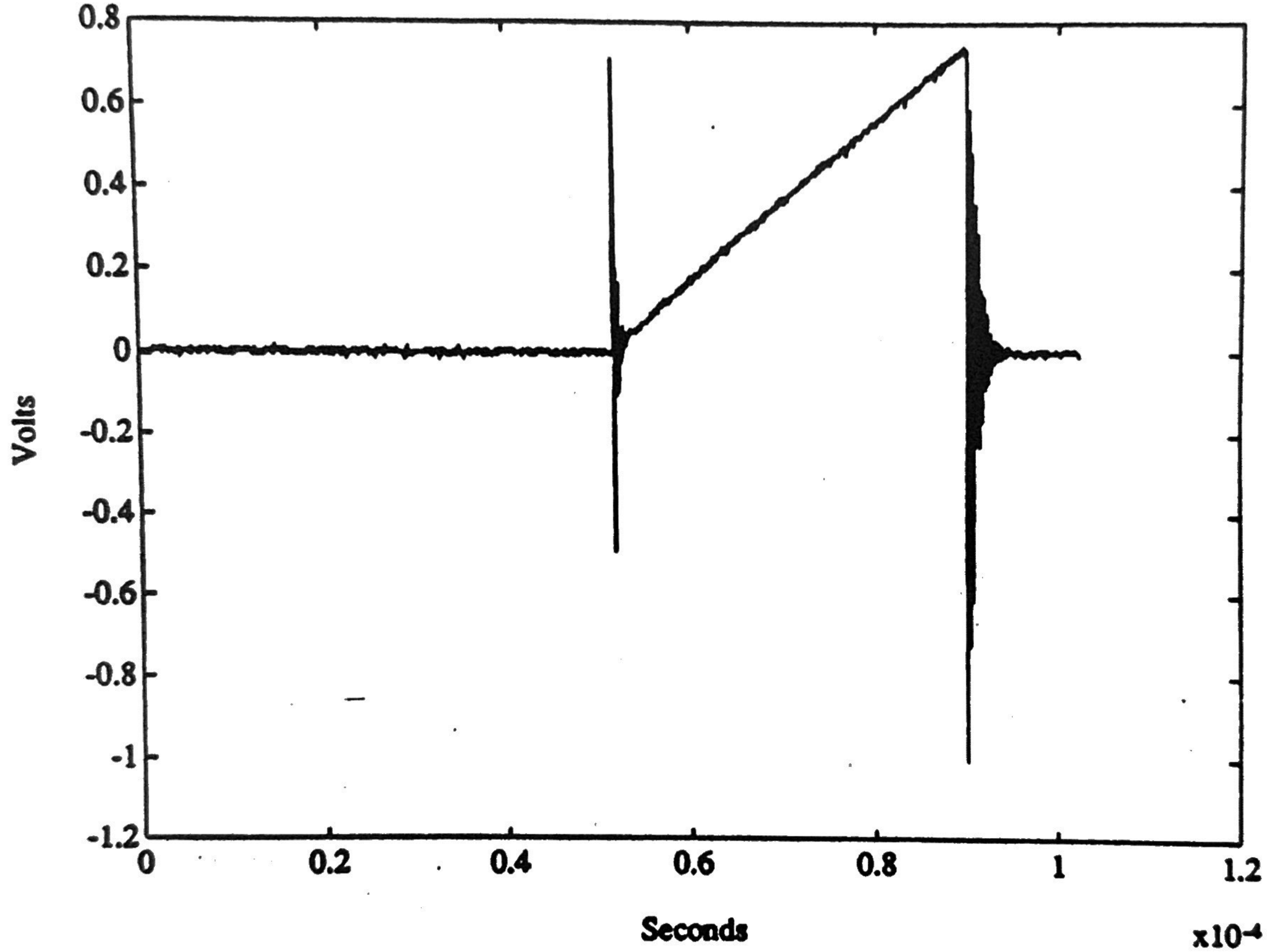
$w = 2 \times \pi \times 5e7$ rads/sec

Median Filtered Switch Voltage



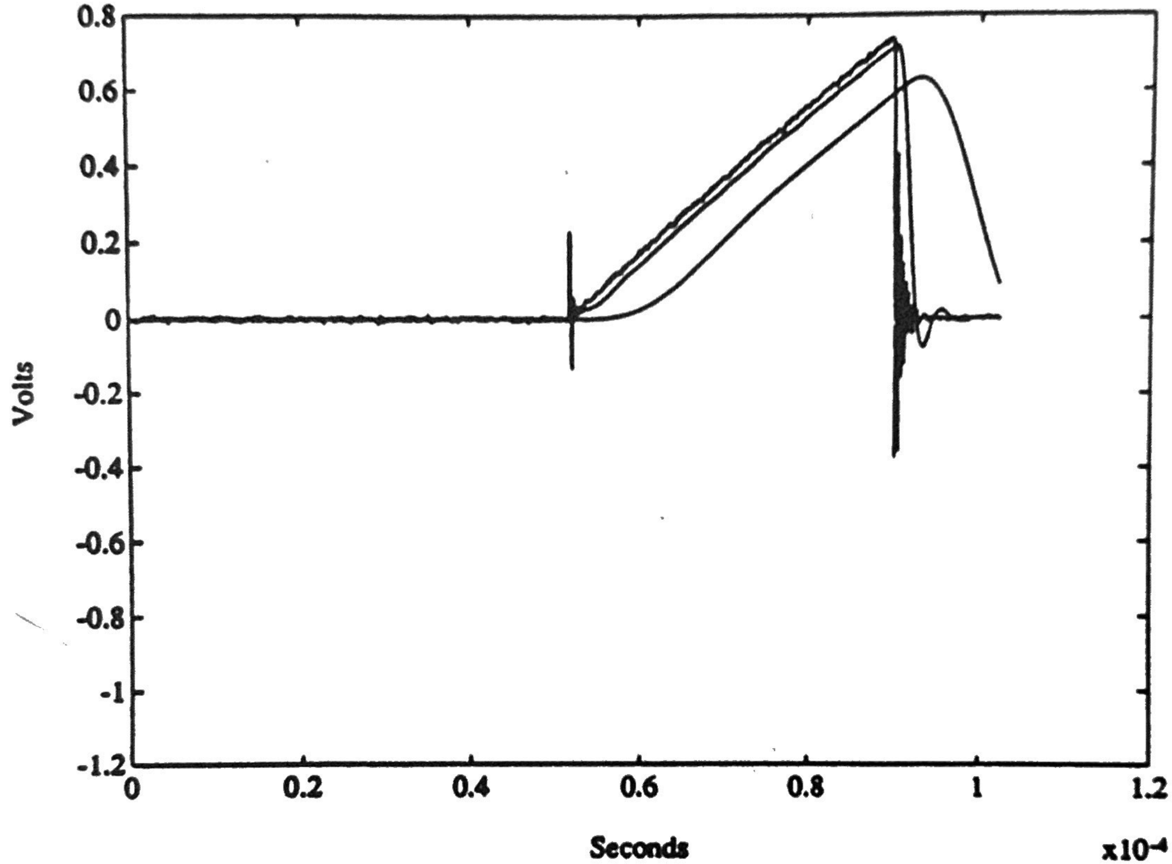
Filter Size N = 8

Voltage Across the Switch Current Sense Resistor



Sample Rate = 10 MHz

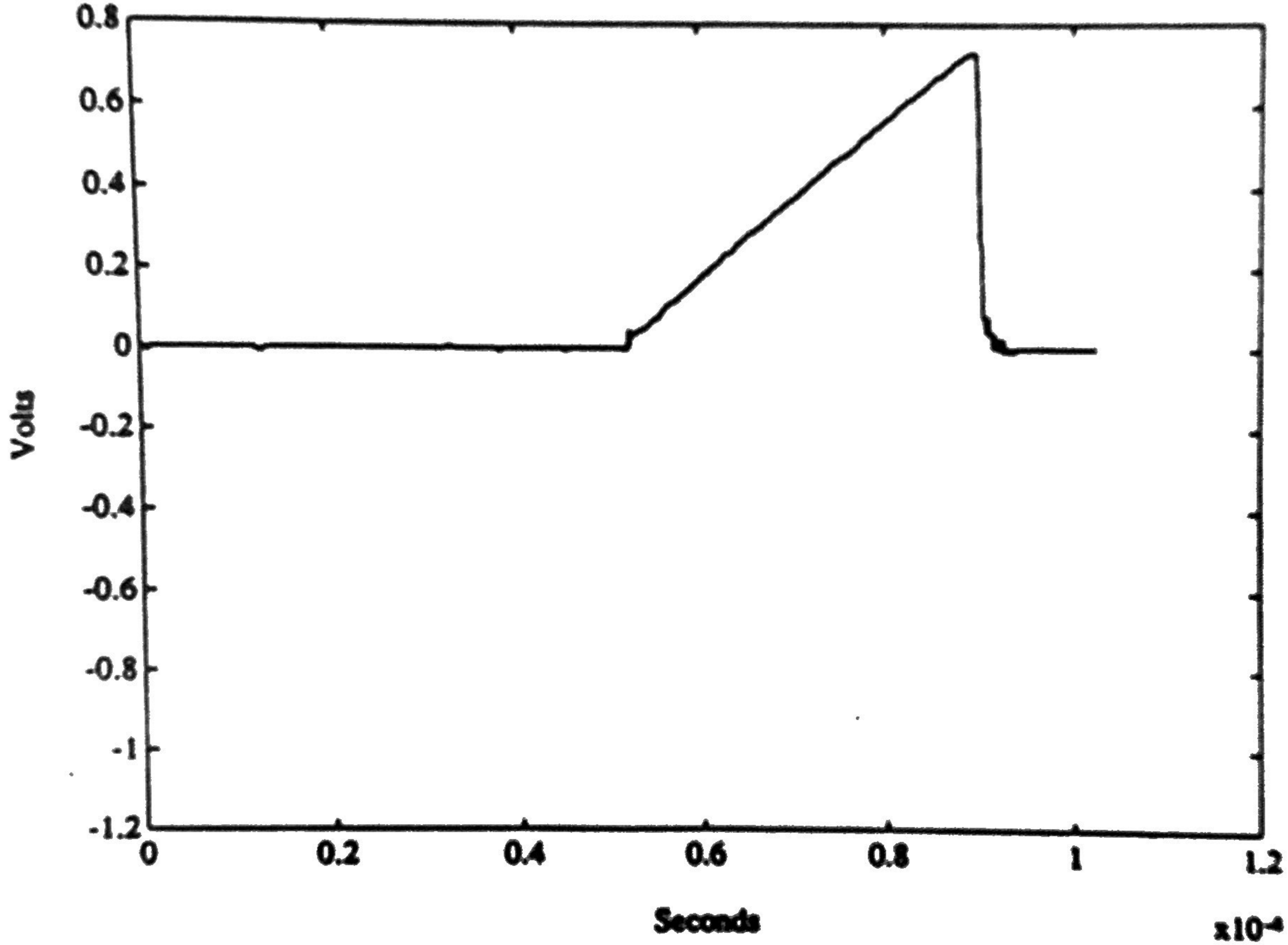
Lowpass Filtered Current Sense Voltage



Fourth-Order Butterworth Filters with cutoff frequencies at 0.5w, 0.05w, 0.01w.

$$w = 2 \times \pi \times 5\text{ e}7 \text{ rads/sec}$$

Median Filtered Current Sense Voltage



Filter Size N = 5

Design Considerations

Median filters are nonlinear, shaping the local form or shape of signals. Examination of spectral behavior is of restricted value.

Trade-off: For greater smoothing, pick N large so that more signal structures appear as impulses and oscillations, which will be removed and reduced.

To preserve an underlying signal, the smallest structure of interest must be at least N+1 points long (the geometric bandwidth).