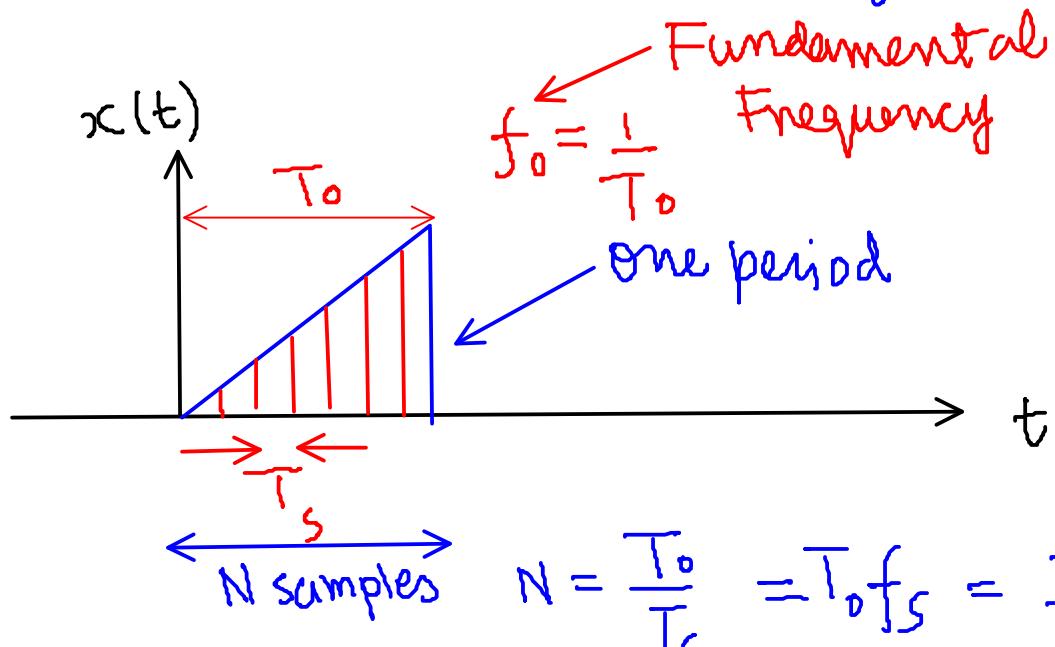


MATLAB  
Fast Fourier Transform  
Fourier Series Spectrum using FFT



spectrum  
of  $x(t)$

This is what we  
want

$N_h$ : number of harmonics  
to be included in FFT

$$f_s = N f_0$$

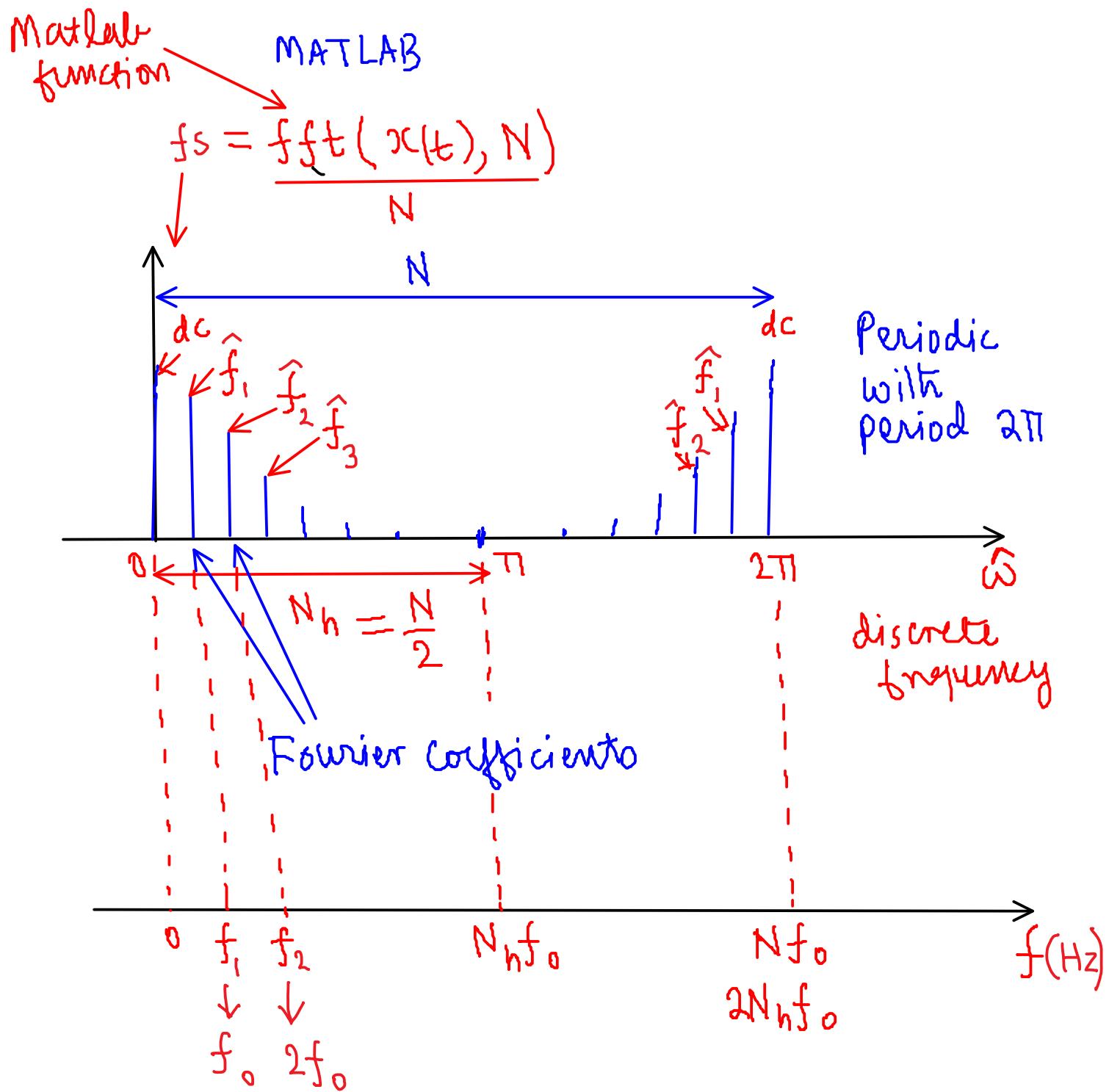
$$f_0, 2f_0, 3f_0$$

Hz

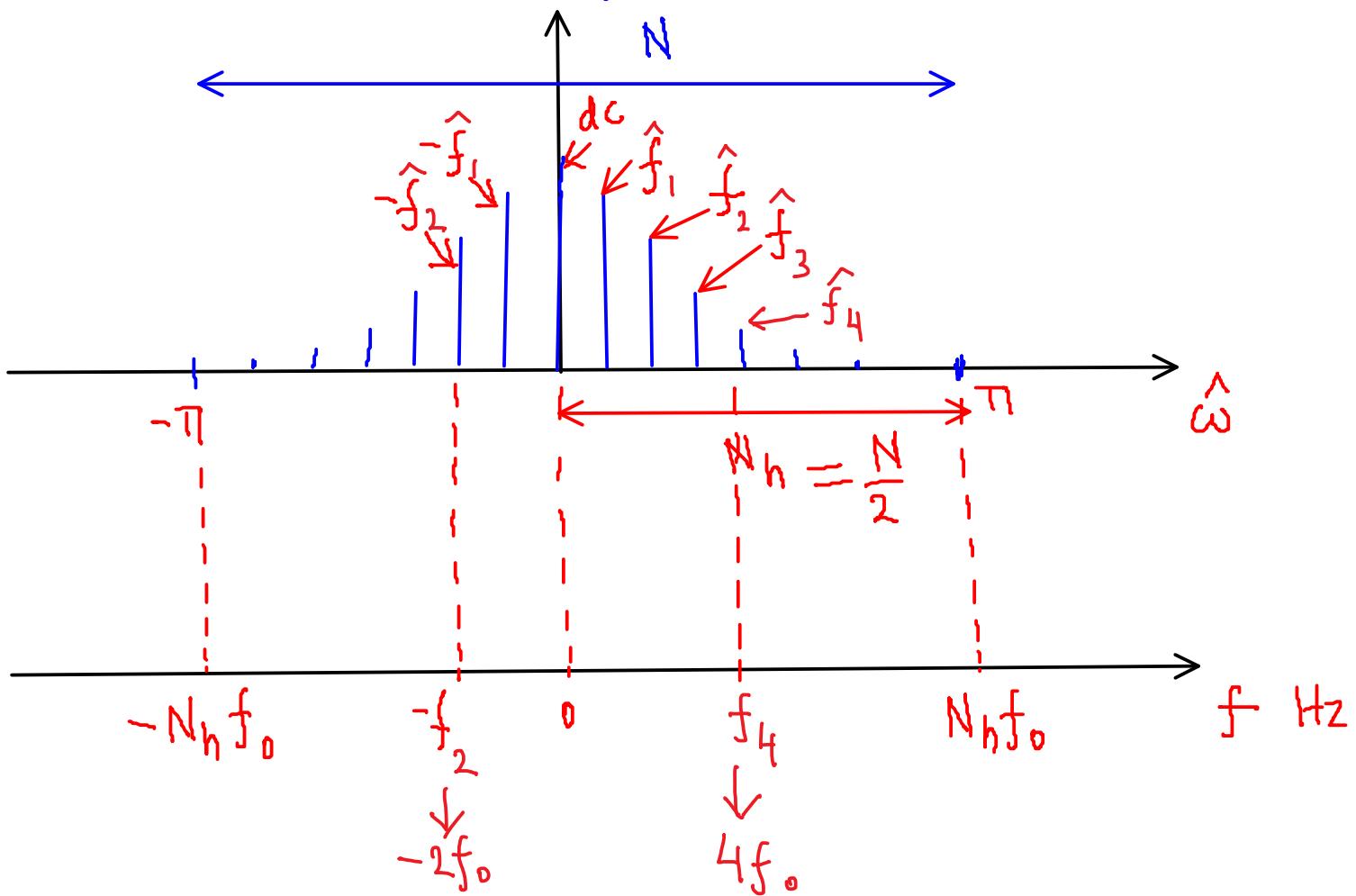
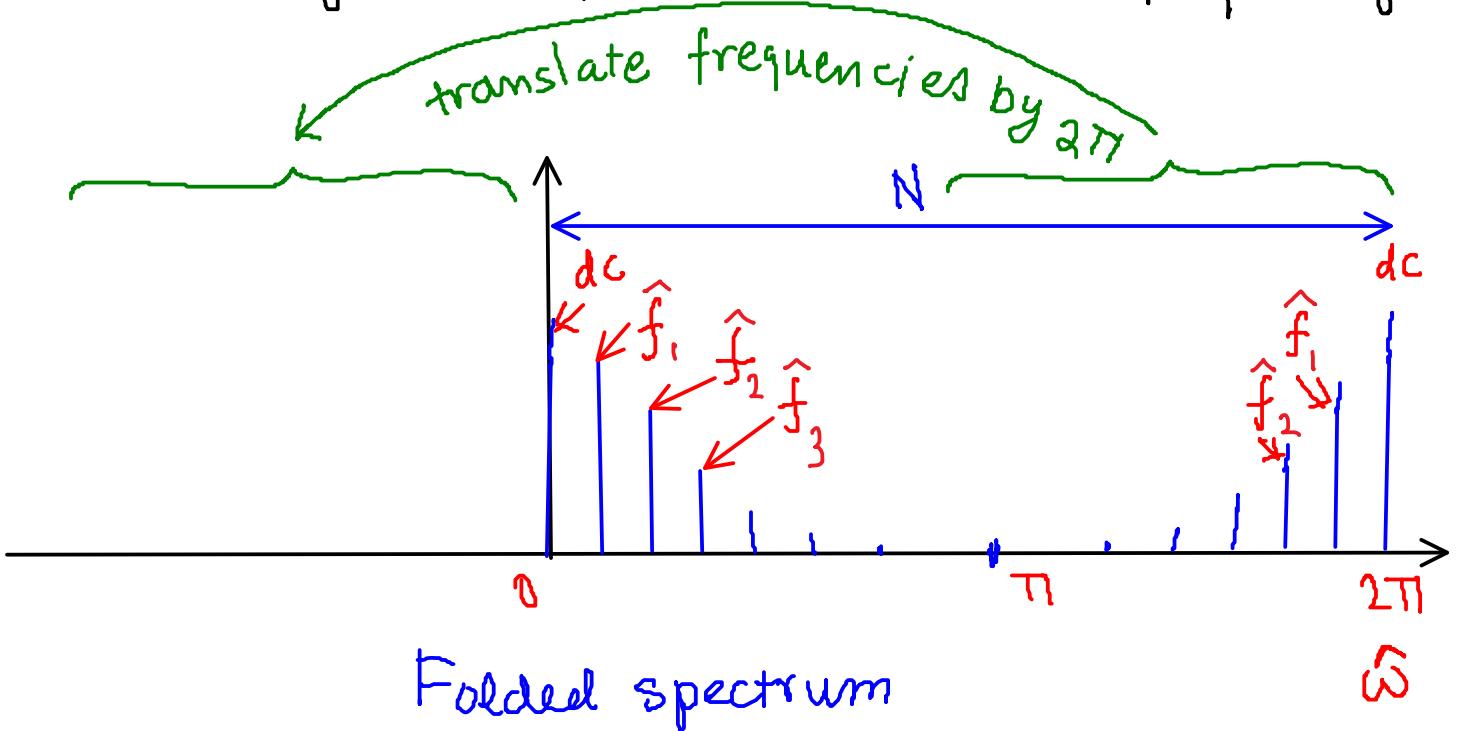
$\left. \begin{array}{l} N_h f_0 \\ \hline f_s \end{array} \right\}$  Sampling theorem

$$N_h f_0 = \frac{f_s}{2} = \frac{N f_0}{2}$$

$$N = 2N_h$$



Folding the spectrum around 0 frequency



```
%%%%%
%
% This MATLAB sample demonstrates how to obtain the
% spectrum of a periodic signal using the MATLAB
% implementation of the FFT algorithm. This sample also
% demonstrates how to reconstruct the original signal from
% the spectrum
%
%%%%%
%
% input parameters
F0 = 1e3; % fundamental frequency of the periodic signal
Nh = 80; % number of desired harmonics in the FFT spectrum

% calculated parameters
T0 = 1/F0; % fundamental period
N = 2*Nh; % samples per period
Ts = T0/N; % Sampling time

t = [0:T0:Nh]; % set the sampling times array
t(end) = []; % get rid of last point
y = 0.5*(1+square(2*pi*F0*t)); % construct the periodic function with
% F0 fundamental period sampled at times
% in the array t
% If y(t) has a discontinuity at t0 then set the value of
% y(t0) = (y(t0+)+y(t0-))/2
y(1)=0.5;
y(Nh+1)=0.5;

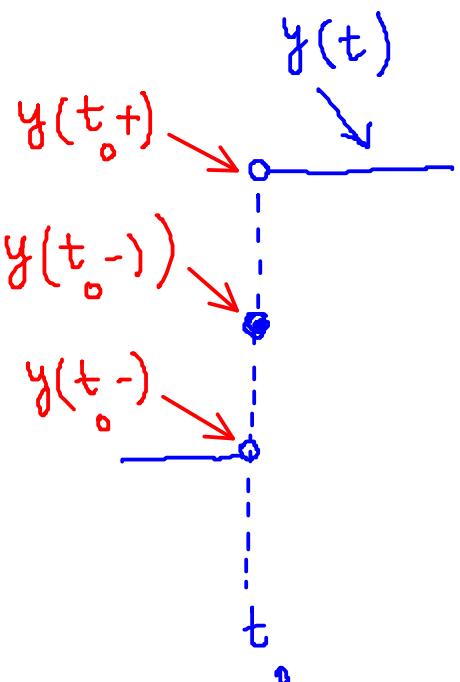
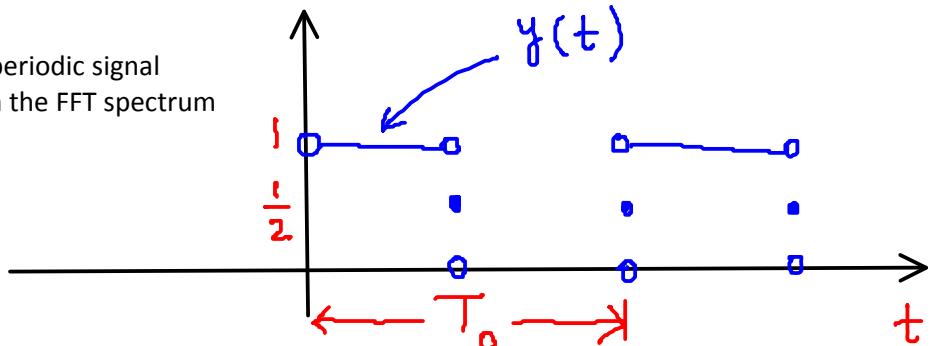
% Use fft to determine the spectrum of y(t)
fs = fft(y,N)/N;

%Fold the frequency axis around zero frequency
index = -Nh:Nh-1;
fs = [ fs(Nh+1:end) fs(1:Nh)];;

fs_abs = abs(fs); % magnitude spectrum
fs_angle = angle(fs); % phase spectrum

% Eliminate points from the phase spectrum that have negligible
% value in the magnitude spectrum
```

This sample uses  
50% duty cycle  
square wave as  
an example



... continued on next page ...

## Lecture 31 - Page 5

```
k=0;
for n = 1:size(fs_angle,2)
    if fs_abs(n) > 1e-13
        k=k+1;
        index2(k) = index(n);
        fs_angle_rad(k) = fs_angle(n);
    end
end
fs_angle_deg = fs_angle_rad*180/pi; % phase spectrum in degrees

%-----
%      plot lollipop plot of magnitude spectrum
subplot(2,1,1)
stem(index,fs_abs);
Title('Square Wave') % edit this line to match your periodic wave
xlabel('Harmonic Number')
ylabel('magnitude')

%-----
%      plot lollipop plot of phase spectrum
% subplot(2,1,2)
% stem(index2,fs_angle_deg);
% xlabel('Harmonic Number')
% ylabel('phase (deg)')

%-----
% Reconstruct the original signal using the spectrum
% obtained by using FFT
% Nhr is the number of harmonics to include
%           in the reconstructed signal
% Nr is the number of time ticks per period to plot
% the reconstructed signal
%
% Nperiod is the number of periods to plot

Nhr = 79; % adjust this as needed
Nr = 500; % adjust this as needed
Nperiod = 2; % adjust as needed

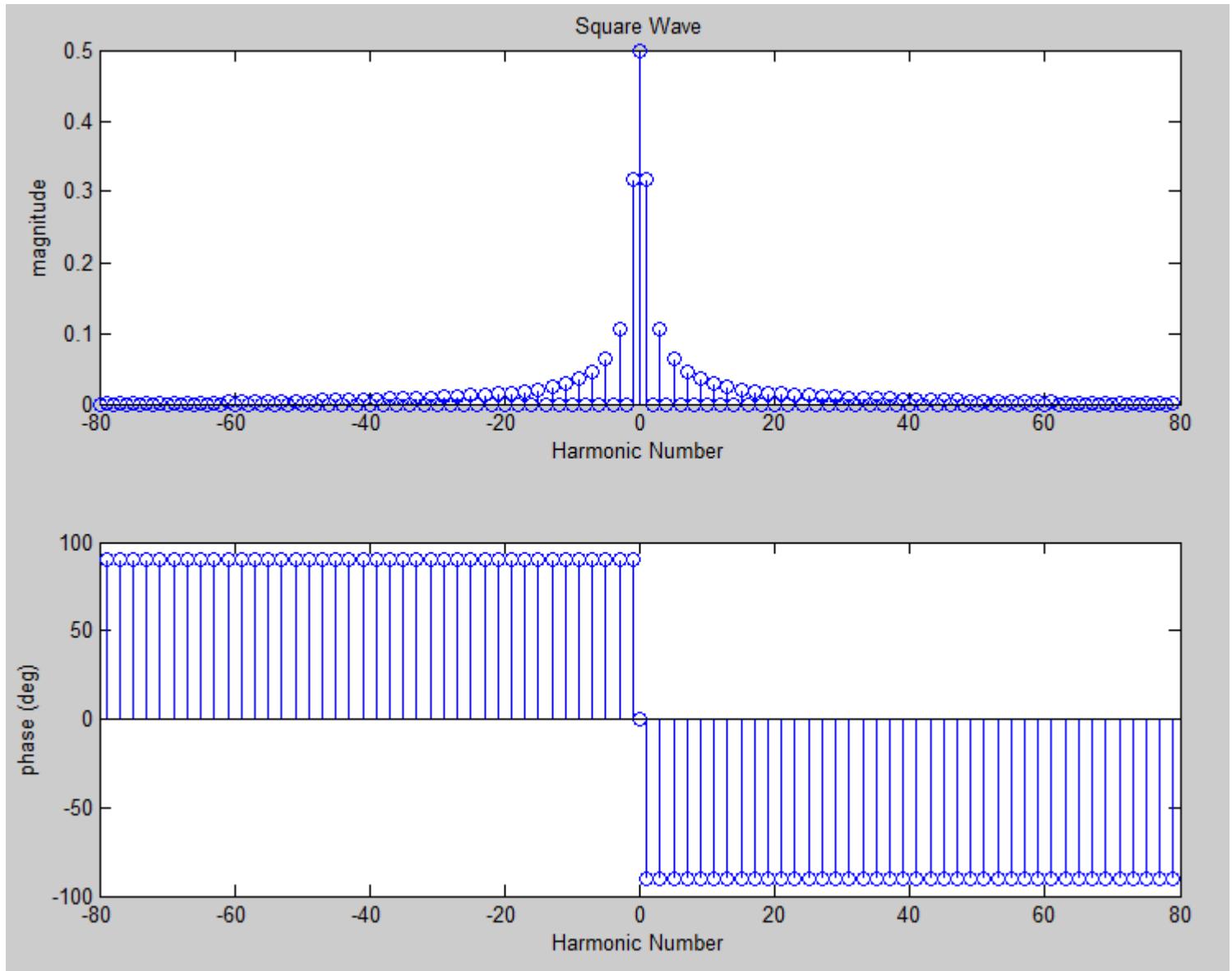
Tr = Nperiod*T0;
Ts = T0/Nr;
tr = [0: Ts: Tr];

% sum the harmonics from the spectrum
yr = fs(Nh+1);
for m = 1:Nhr
    yr = yr + fs(Nh+1-m)*exp(-j*2*pi*m*F0*tr) + ...
        fs(Nh+m+1)* exp(j*2*pi*m*F0*tr);
end

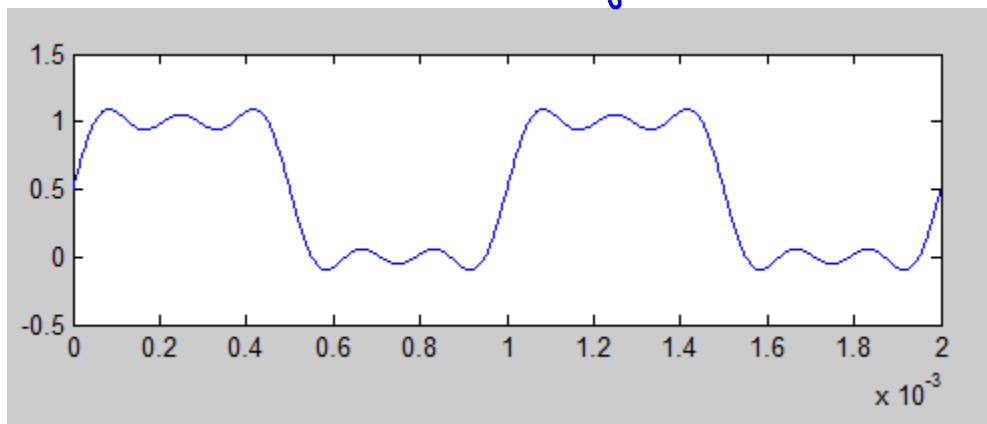
subplot(2,1,2)
plot(tr,yr); % plot the reconstructed signal
```

## Output from sample program

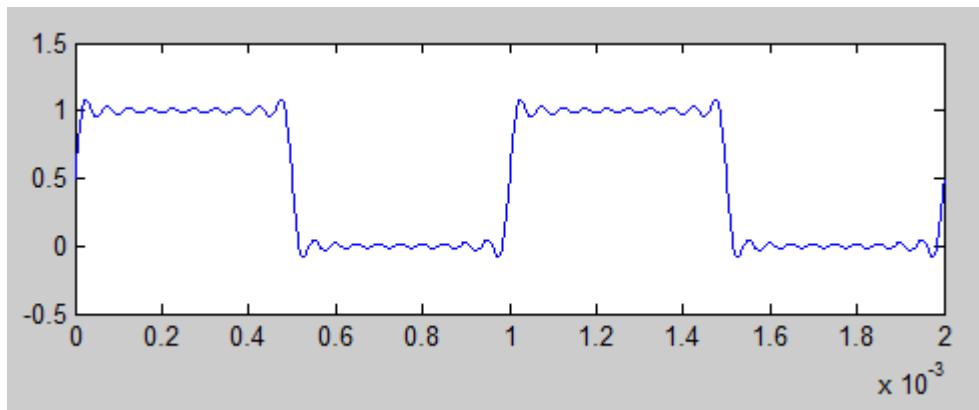
Fundamental frequency =  $F_0 = 1000$  Hz, Desired # of harmonics  $N_h = 80$



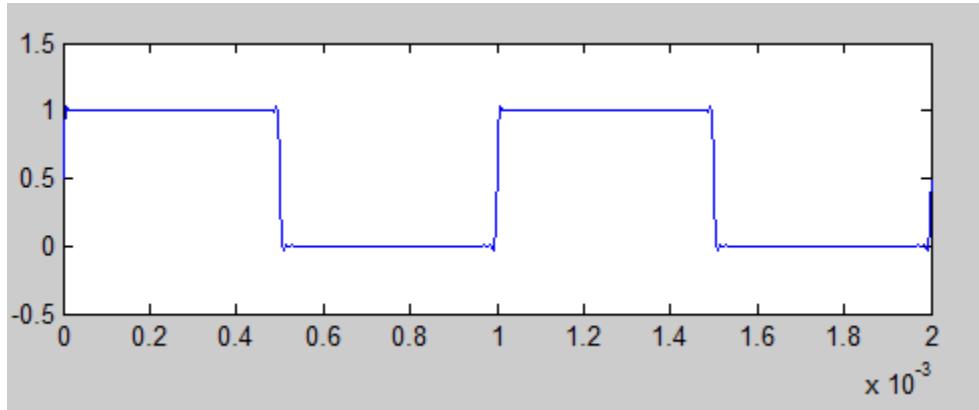
reconstructed signal



Number of Harmonics (Nhr) = 5, Nr = 500



Number of Harmonics (Nhr)= 20, Nr = 500



Number of Harmonics(Nhr) = 79, Nr = 500