PROPERTIES OF THE DFT

INTRODUCTION

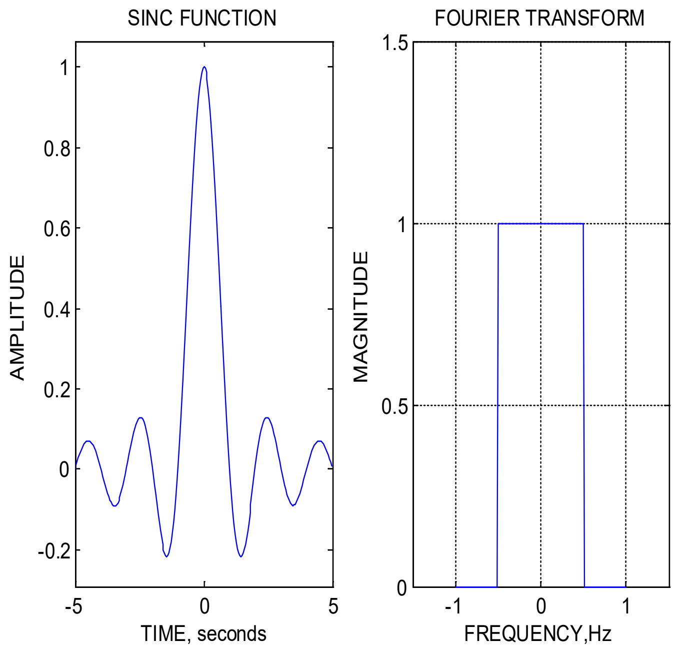
The discrete Fourier Transform (DFT) is the commonly and extensively used transform for estimating the continuous time Fourier transform (CTFT) of measured signals and discrete time data. Signals measured over time and discretized are synonymously called *time series*. As one shall see the basic characteristics of the DFT sometimes produce errors in the estimate of the CTFT. Luckily there are ways to improve the estimate very satisfactorily. This notebook will show these error sources and correction techniques by starting with a signal whose CTFT is known and easy to understand. Then it will be estimated and corrected to an acceptable level. This will establish the procedure for correctly implementing the DFT.

The function being used is the sinc function defined as

$$\text{sinc}(t) = \frac{\sin(2\pi f_0 t)}{2\pi f_0 t} \tag{1}$$

where f_0 = 0.5 Hz. The function and its CTFT are plotted in figure 1. Theoretically the sinc function is infinite in duration but only a finite duration is plotted.

```
clear; close all; format compact; whitebg('w')
syms t
f0 = 0.5; %Hz
ft = 'sin(2*pi*0.5*t)/(2*pi*0.5*t)'; % SINC FUNCTION
% an indeterminate value exists at the peak of the waveform
figure (1)
subplot(1,2,1); ezplot(ft,[-5 5])
xlabel('TIME, seconds'); ylabel('AMPLITUDE');
title('SINC FUNCTION')
FT = zeros(201,1); FT(51:151) = ones(101,1);
fsp = 0.01; f = (-100:100)*fsp;
subplot(1,2,2); plot(f,FT);
xlabel('FREQUENCY,Hz'); ylabel('MAGNITUDE');
title('FOURIER TRANSFORM')
axis([-1.5 1.5 0 1.5]); grid
```



DFT

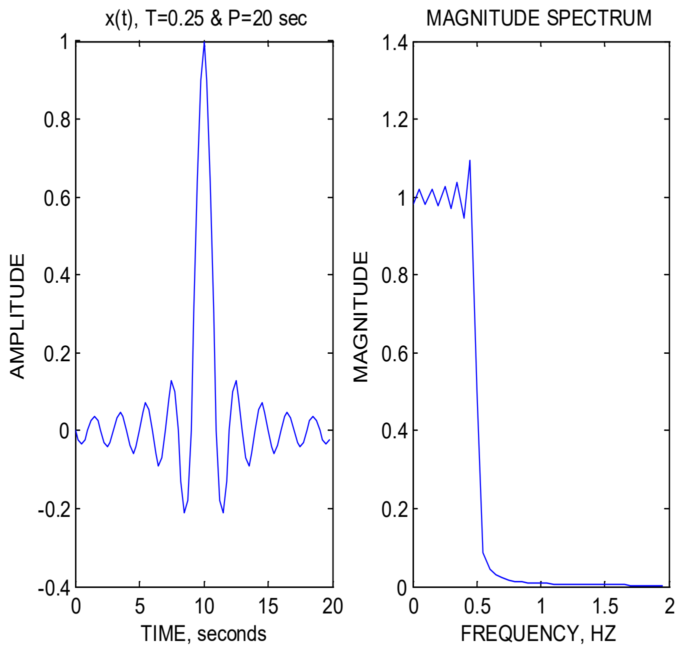
To practically implement the DFT the waveform will be shifted since the DFT operates in positive time. Remember that the DFT, *X(m)*, of a sampled signal, *x(n)*, is defined as

$$X(m) = T\sum_{n=0}^{N-1} x(n)\exp(-j2\pi \tfrac{mn}{N}) \tag{2}$$

Also a finite duration of 20 seconds will be used. Let the sampling interval be 0.25 seconds. The signal and its DFT are shown in Figure 2. MATLAB calculates the summation part of equation 2 with the function 'fft'.

```
T = 0.25; t = [0:T:(20-T)]';
N = length(t);
shift = 40*ones(N,1); % time shift of 40 samples is 40T
y = sin (pi*t - pi*T*shift) ./ (pi*t - pi*T*shift); y(41)=1; % an
% indeterminate value exists at the peak of the waveform
figure(2)
subplot(1,2,1); plot(t,y)
title('x(t), T=0.25 & P=20 sec');xlabel('TIME, seconds');
ylabel('AMPLITUDE')
Y = T*fft(y); MY = abs(Y); fd = 1/(N*T);
f = [0:fd:(N/2-1)*fd]';
mag = MY(1:(N)/2); % only use the first half because the remainder is
redundant
subplot(1,2,2); plot(f,mag)
title('MAGNITUDE SPECTRUM'); xlabel('FREQUENCY, HZ');
ylabel('MAGNITUDE')

Warning: Divide by zero.
```

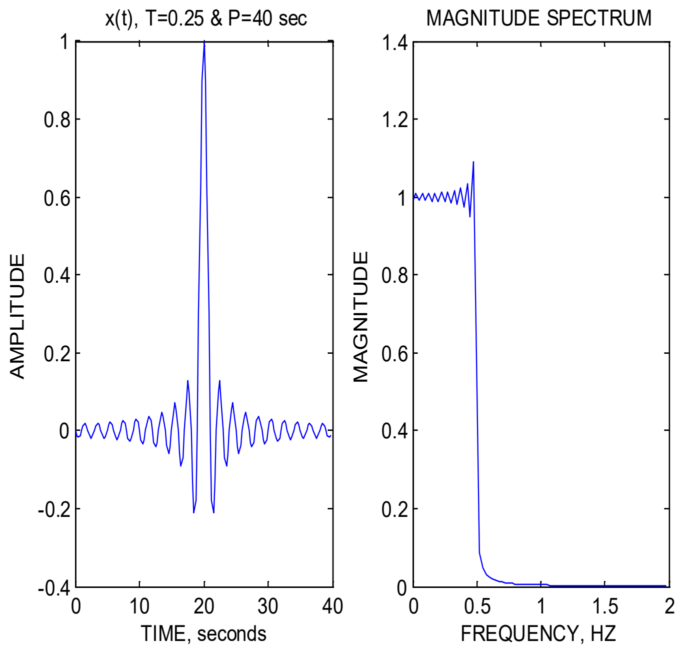


LEAKAGE ERROR

Compare visually the magnitude spectra in Figures 1 and 2. How do they differ along the frequency axis? How do they differ in the shape of the spectra? The oscillating error that is observed is called the *leakage error*. From the reading you will recall that it occurs because there is only a finite duration of signal used. Perhaps a longer signal will reduce the leakage error? Let's try it and show the results in Figure 3.

```
t1 = [0:T:40-T]'; % 40 second signal
N1 = length(t1); shift1 = 80*ones(N1,1);
y1 = sin (pi*t1 - pi*T*shift1) ./ (pi*t1 - pi*T*shift1); y1(81)=1;
figure(3)
subplot(1,2,1); plot(t1,y1)
title('x(t), T=0.25 & P=40 sec');xlabel('TIME, seconds');
ylabel('AMPLITUDE')
Y1 = T*fft(y1); MY1 = abs(Y1); fd = 1/(N1*T);
f1 = [0:fd:(N1/2-1)*fd]'; mag = MY1(1:N1/2);
subplot(1,2,2); plot(f1,mag)
title('MAGNITUDE SPECTRUM'); xlabel('FREQUENCY, HZ');
ylabel('MAGNITUDE')
```

```
Warning: Divide by zero.
```

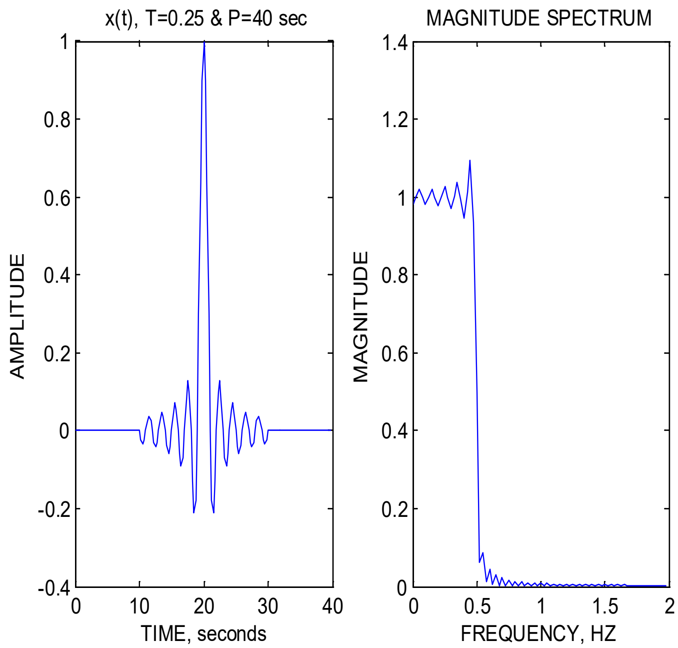


Visually compare the magnitude spectra in Figures 2 and 3. Has the leakage error been reduced? Increase the signal duration by 50%, calculate the DFT, and plot the resulting magnitude spectrum. Is this spectrum closer to the theoretical one?

ZERO PADDING

Before a correction method for leakage error is demonstrated**,** let's explore another way to increase the duration of a signal; this one is artificial. It is called *zero padding* because that is what is exactly done. In order to lengthen a signal one simply appends a vector of zeros to the signal vector. This is sometimes necessary. To demonstrate this we will take the signal in figure 2 and append and prepend zeros; the number equals the number of signal points.

```
for n=1:N1/4; y1(n)=0.; end; % prepend zeros
for n=3*N1/4+1:N1; y1(n)=0.; end; % append zeros
figure(4)
subplot(1,2,1); plot(t1,y1)
title('x(t), T=0.25 & P=40 sec');xlabel('TIME, seconds');
ylabel('AMPLITUDE')
Y1 = T*fft(y1); MY1 = abs(Y1); fd = 1/(N1*T);
f1 = [0:fd:(N1/2-1)*fd]'; mag = MY1(1:N1/2);
subplot(1,2,2); plot(f1,mag)
title('MAGNITUDE SPECTRUM'); xlabel('FREQUENCY, HZ');
ylabel('MAGNITUDE')
```



Visually compare Figures 4 and 3; visually compare Figures 4 and 2. How are they similar or different? What are the frequency spacings in the 3 figures? Has zero padding at least improved the resolution along the frequency axis?

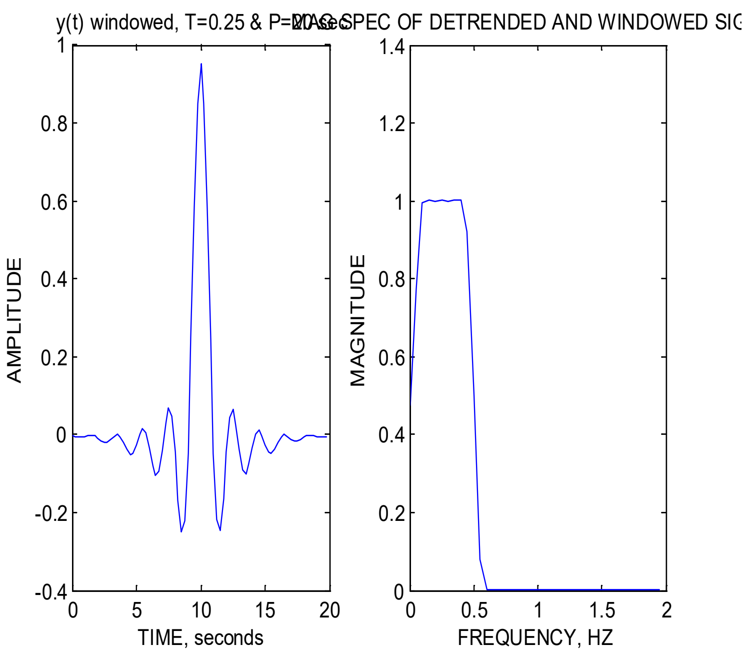
DETRENDING AND WINDOWING

In a previous section it was stated that there is a technique for reducing leakage error; it is called *windowing*. The concept is based on the following reasoning which is actually a mathematical model for forming a finite duration signal. Theoretically all signals are infinitely long; that is they are longer than the time in which we have to measure them. It is said that we 'look' at an infinitely long signal through a window and thus we only 'see' a finite duration signal. Mathematically this is equivalent to multiplying an infinitely long signal by a function called *rectangular data window*. The DFT of a truncated signal is equal to the convolution of the true spectrum and the CTFT of the rectangular data window. This happens to be the sinc function and hence the oscillatory inaccuracies. These oscillations are called *lobes*. The solution is to use another data window that has a CTFT whose lobes are smaller in magnitude. Before windowing is done to a signal another error source must be considered. This error is caused by a trend in the data. Since the DFT is actually producing a frequency model of the signal, any linear or polynomial trends must removed first because they contaminate the coefficients of the model. What is done is that a linear or polynomial model is made for the signal and then subtracted from it. This is called *detrending*. The entire procedure will now be applied to the signal in Figure 2.

```
T = 0.25; t = [0:T:20-T]';
N = length(t); shift = 40*ones(N,1);
y = sin (pi*t - pi*T*shift) ./ (pi*t - pi*T*shift); y(41)=1;
yd = detrend(y,1); %detrend signal by subtracting a linear model
figure(6)
subplot(1,2,1); plot(t,yd)
```

```
title('y(t) detrended, T=0.25 & P=20 sec');xlabel('TIME, seconds');
ylabel('AMPLITUDE')
win = hamming(N); % points of a hamming window
subplot(1,2,2); plot(t,win)
title('HAMMING WINDOW, T=0.25 & P=20 sec');xlabel('TIME, seconds');
ylabel('AMPLITUDE')
ydw = win .* yd; % windowed signal; reducing leakage error
figure(7)
subplot(1,2,1); plot(t,ydw)
title('y(t) windowed, T=0.25 & P=20 sec');xlabel('TIME, seconds');
ylabel('AMPLITUDE')
%
Y = T*fft(ydw); MY = abs(Y); fd = 1/(N*T);
f = [0:fd:(N/2-1)*fd]'; mag = MY(1:N/2);
subplot(1,2,2); plot(f,mag)
title('MAG SPEC OF DETRENDED AND WINDOWED SIGNAL');
xlabel('FREQUENCY, HZ'); ylabel('MAGNITUDE')
```

```
Warning: Divide by zero.
```



<u>Visually compare the spectrum in Figure 7 with the one in Figure 2 and Figure 1. Has the process made an improvement? Are there any shortcomings? Use another data window in place of the hamming window and plot the resulting DFT. Has it made any improvement in the process?</u>

VIBRATION SIGNAL

Measurement of vibration is important in mechanical systems because they indicate either a bad design, or a part that is worn or broken, or a deteriorating chassis securement. The frequency components are related to the type of malfunction or a broken/worn part. It is loaded and the direct DFT is calculated. From visual examination of the signal, what frequency components can you determine? It seems that strong frequency components occur at the frequencies of approximately 1, 2, 4, and 6 Hz. <u>Apply detrending and</u>

windowing properly and re-estimate the magnitude spectrum. What frequency components exist?

```
load vib.dat; T = 0.02; % seconds
N = length(vib); t = (1:N)'*T;
figure(8); subplot(2,1,1); plot(t,vib)
title('MACHINE VIBRATIONS'); xlabel('TIME,seconds');
ylabel('acceleration');
%
vibdt = detrend(vib); %removes average value
VIB = T*fft(vibdt); magvib = abs(VIB(1:N/2));
fd = 1/(N*T); f = (0:N/2-1)*fd;
subplot(2,1,2); plot(f,magvib); grid
title('MAGNITUDE SPECTRUM'); xlabel('FREQUENCY, HZ');
ylabel('MAGNITUDE')
```