Discrete Fourier Transform

Given an N-point time series \( x(n) \), the Z transform is given by

\[
X(z) = \sum_{n=0}^{N-1} x(n)z^{-n} = \sum_{n=1}^{N} x(n)z^{-(n-1)}
\]

The Discrete Fourier Transform (DFT) is given by sampling the Z-transform at \( N \) points equally spaced around the unit circle, as shown. This is accomplished by defining a frequency index \( k \) by

\[
z = e^{j(2\pi/N)(k-1)},
\]

where \( k \) runs from 1 to \( N \). The Z-transform now becomes a function of \( k \) so that the DFT is written

\[
X(k) = \sum_{n=1}^{N} x(n) e^{-j2\pi(k-1)(n-1)/N}, \quad k = 1,2,...,N
\]

The inverse DFT is given by

\[
x(n) = \frac{1}{N} \sum_{k=1}^{N} X(k)e^{j2\pi(k-1)(n-1)/N}, \quad n = 1,2,...,N
\]

The DFT is periodic with period \( N \). The inverse DFT yields a time function that is periodic with period \( N \).

The DFT of the delta function is the constant function

\[
DFT(\delta(n)) = 1.
\]

Scaling the Frequency Axis

To plot frequency in radians, the frequency axis should be scaled to run from 0 to \( 2\pi \). To accomplish this, for plotting one uses the frequency variable
\[ w = \frac{2\pi}{N} (k - 1). \]

The FFT first harmonic frequency is given by

\[ \Omega = \frac{2\pi}{NT} \]

with \( T \) the sampling period. More discussion will be made later about sampling and relating the continuous and discrete time and frequency scales.

**Example 1- Digital Filter**

Let \( x(n) \) be generated by

\[ x(k + 1) = \beta \cdot x(k), \quad x(1) = 1 \]

Then \( x(n) = x(1) \beta^{n-1} \).

The plots for \( x(n) \), \text{mag}(X(k))\), and \text{angle}(X(k))\) are given below.

MATLAB m file

```
function [ki,x]=expon(N,beta);
x(1)= 1; ki(1)=1 ;
for k=1:N-1
    ki(k+1)=k+1;
    x(k+1)=beta*x(k);
end
>> plot(ki,x)
```

\[ \text{>
} \text{> dft=fft(x);}
\]

\[ \text{>
} \text{> loglog(ki,abs(dft))}
\]

\[ \text{>
} \text{> semilogx(ki,angle(dft))}
\]
Example 2- FFT of Time Window

Let \( x(n)= 1,1,1,1,1,1,1,1,0,0,0 \ldots \) as shown, with eight ones followed by zeros.

Take first the fft of this \( x(n) \) with length 32--

\[
>> x=[1 1 1 1 1 1 1 1]'
\]
\[
>> y=fft(x,32)
\]
\[
>> k=1:32
\]
\[
>> plot(k,abs(y))
\]

Note that the resolution of the FFT is not good. To fix this, pad with extra zeros at the end–

\[
>> y=fft(x,512)
\]
\[
>> k=1:512
\]
\[
>> w=2*\pi*(k-1)/512
\]
\[
>> plot(w,abs(y))
\]

See how much better the resolution is now. Note that we have scaled the frequency to run from 0 to \( 2\pi \) rads.

Windowed DFT

Most signals of interest have time-varying characteristics, including time-varying spectra. The best-known example is speech. Also of interest are financial market stock prices. Therefore, one usually computes a windowed DFT at time \( N \) using only the past \( m \) values of the signal

\[
X(k, N) = \sum_{n=N-m+1}^{N} x(n) e^{-j2\pi(k-1)(n-1)/N}.
\]

Now, the DFT is a function of two variables, both the time index \( N \) and the frequency \( k \). It must be plotted as a 3-D plot.
In some applications one does not consider the variation in time for every sample. Instead, one chops the time index into bins of length m and performs the DFT only for values of \( N = im \) with \( i \) an integer. This is common, e.g., in speech processing.

**Properties of DFT**

The DFT enjoys several nice properties detailed elsewhere. Some of them are given here.

**Parseval’s Theorem** states that

\[
\sum_{n=1}^{N} x^2(n) = \frac{1}{N} \sum_{k=1}^{N} |X^2(k)|
\]

**The Convolution Theorem** States that the transform of the convolution

\[
x \ast y(n) = \sum_{i=1}^{N} x_i y_{n-i}
\]

is given by the product of the transforms \( X(k)Y(k) \).

**The correlation theorem** says that the transform of the correlation

\[
R_{xy}(n) = \frac{1}{N} \sum_{i=1}^{N} x_i y_{i+n}^*
\]

is given by the product \( X(k)Y^*(k)/N \), with superscript * denoting complex conjugate transpose. Note that normally we shall deal with real sequences in the time domain.

A special case of this theorem says that the transform of the autocorrelation

\[
R_x(n) = \frac{1}{N} \sum_{i=1}^{N} x_i x_{i+n}^*
\]

is given by

\[
\Phi_x(k) = \frac{1}{N} X(k)X^*(k) = \frac{1}{N}|X(k)|^2,
\]

which is know as the power spectral density (PSD). It is often easier to see details in the frequency response using the PSD than using \( X(k) \).
Example 3- Using FFT to Extract Frequency Component Information
(From MATLAB reference book under ‘fft’)
A time signal with frequency components at 50 Hz and 120 Hz and added random noise is given by

```matlab
>> t=0:0.001:0.6;
>> x=sin(2*pi*50*t) + sin(2*pi*120*t);
>> y=x + 2*randn(size(t));
```

The plot of y(t) shows that the noise makes it impossible to determine the frequency content of the signal.

To analyze the spectrum, take the dft of the first 512 samples using

```matlab
>> Y=fft(y,512);
```

The magnitude spectrum is plotted. Also plotted for comparison is the spectrum of the uncorrupted x(t).

```matlab
>> plot(abs(X))
```

Mag Spectrum of uncorrupted signal

```matlab
>> plot(abs(Y))
```

Mag Spectrum of noise corrupted signal
The frequency must be scaled to be meaningful. Note from \( t=0:0.001:0.6; \) that the sample time is \( T=0.001 \) sec. The index \( k \) varies from 1 to 512, while the frequency \( f \) varies from 0 to the sampling frequency \( f_s = \frac{1}{T} \).

Therefore a meaningful frequency axis is given by

\[
 f = \frac{f_s}{N} (k - 1) = \frac{1}{NT}(k - 1).
\]

Redo the plot using

\[
 f=1000*(0:255)/512
\]
\[
 >> \text{plot}(f,\text{abs}(Y(1:256)))
\]

Only the first half of the spectrum is needed since it is symmetric.

Note the presence of spikes at 50 Hz and 120 Hz. The peaks are better detected using the power spectral density, which gives the energy at each frequency.

\[
 >> \text{Py}=Y.*\text{conj}(Y)/512;
\]
\[
 >> \text{plot}(f,\text{Py}(1:256))
\]
Discrete Fourier Transform -

Time-varying DFT using window (using MATLAB FFT)
One second buffer DFT of the speech at a refreshing rate of one second

Onset of gear tooth wear

Resulting load imbalance

Intermittent incipient bearing outer race fault

DFT for CBM