Utility Functions Octave Codes (0B)

Young Won Lim
2/2/18

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

Based on
M.J. Roberts, Fundamentals of Signals and Systems

Time and Frequency Domain Plot

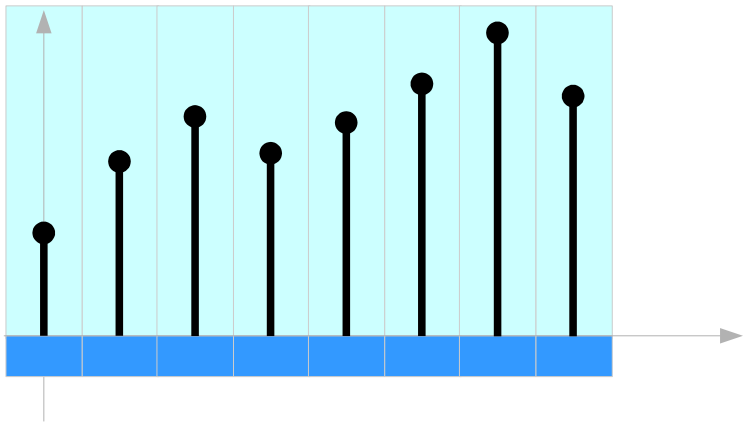
x(1:NF) Time domain signal

n(1:NF) Time index vector

X(1:NF) Frequency domain spectrum

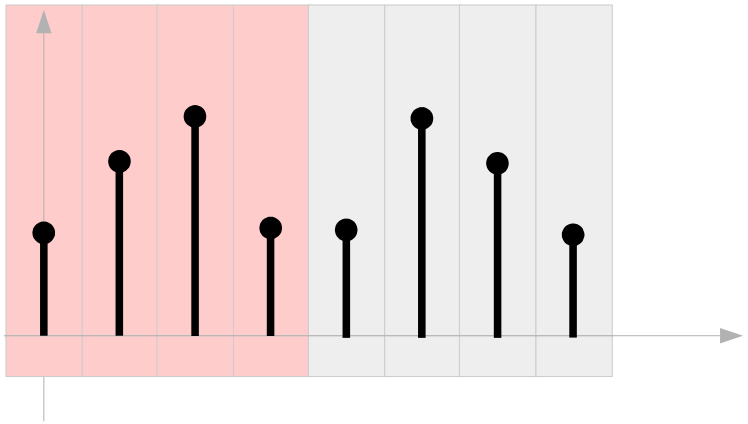
k(1:NF) Frequency index vector

Time and Frequency Domain Plot



x(1:NF) **x**(1:NF)

n(1:NF) **n**(1:NF)



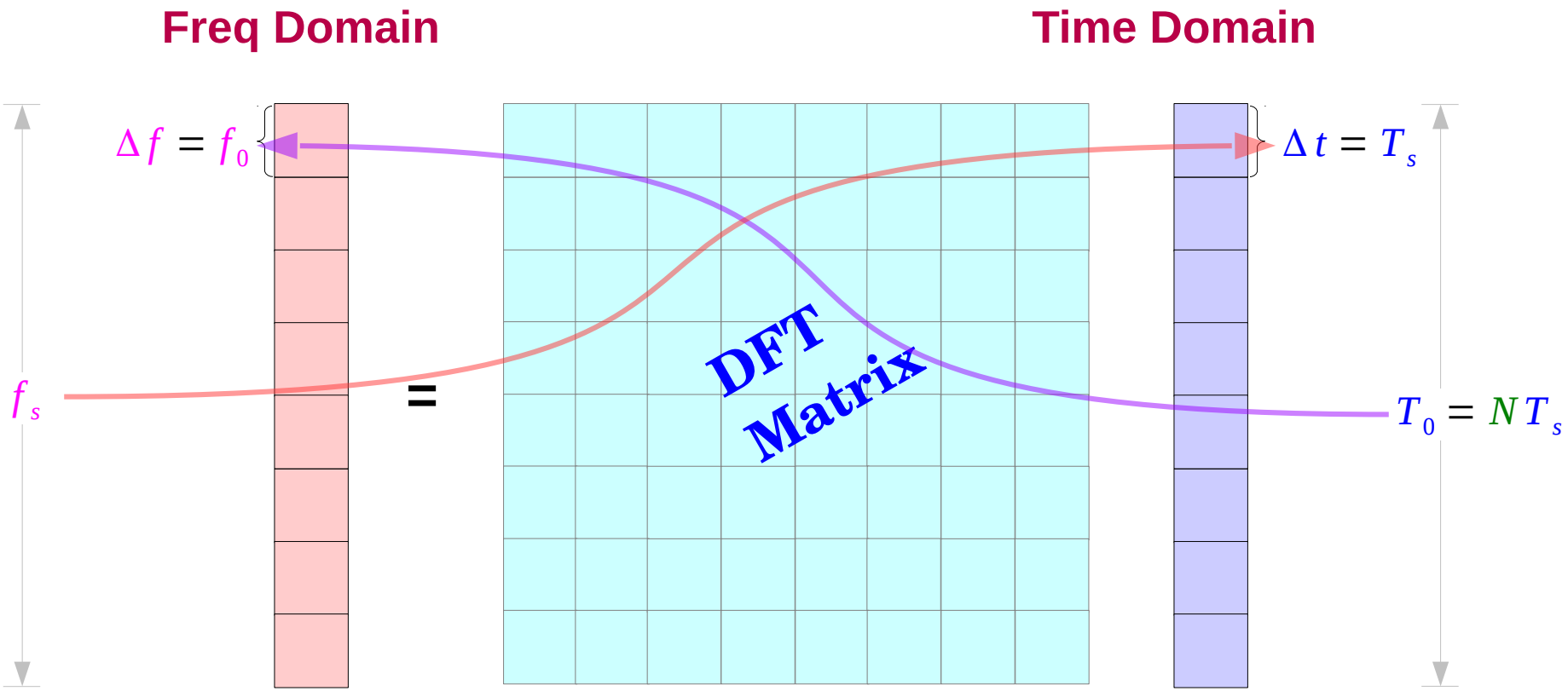
X(1:NF) **X**(1:NF/2)

k(1:NF) **k**(1:NF/2)

Time and Frequency Domain Plot

	mode	a:b	'full'	'half'
x(1:NF)	Time domain signal	**x**(a:b)	**x**(1:NF)	**x**(1:NF)
n(1:NF)	Time index vector	**n**(a:b)	**n**(1:NF)	**n**(1:NF)
X(1:NF)	Frequency domain spectrum	**X**(a:b)	**X**(1:NF)	**X**(1:NF/2)
k(1:NF)	Frequency index vector	**k**(a:b)	**k**(1:NF)	**k**(1:NF/2)

Using Sampling Frequency and Time



$$f_0 = \frac{1}{T_0} = \frac{1}{N T_s} \qquad f_s \geq \frac{2}{T_0} \qquad f_s \geq 2 f_0 \qquad T_s = \frac{1}{f_s}$$

Time and Frequency Domain Plot

```
subplot(3,1,1);
p = stem(n(trange), x(trange), 'k'); grid on;
xlabel('Time, t (s)'); ylabel('x(t)');

subplot(3,1,2);
p = stem(k(frange), abs(X(frange)), 'k');
set(p,'LineWidth',2,'MarkerSize',4); grid on;
xlabel('Harmonic number, k');
ylabel('|X(k)|');

subplot(3,1,3);
p = stem(k(frange), angle(X(frange)), 'k');
set(p,'LineWidth',2,'MarkerSize',4); grid on;
xlabel('Harmonic Number, k');
ylabel('Ang{X[k]}');

endfunction
```

Time and Frequency Domain Plot

```
# Prevent Octave from thinking that this
# is a function file:
1;

#-----------------------------------------
# n : time index vector
# x : time domain signal vector
# k : frequency index vector
# X : frequency domain signal vector
#-----------------------------------------
function TimeFreqPlot(n, x, k, X, mode)
```

Time and Frequency Domain Plot

```
NF = length(n);

if (strmatch(typeinfo(mode), "range"))
  trange = mode; frange = mode;
else
  if (strmatch(mode, "half"))
    trange = 1:NF; frange = 1:NF/2;
  elseif (strmatch(mode, "full"))
    trange = 1:NF; frange = 1:NF;
    fs = k(2)+k(N)
    X = fftshift(X); k = fftshift(k);
    k(1:NF/2) = k(1:NF/2) - fs;
  else
    trange = 1:NF; frange = 1:NF;
  endif
endif
```

Time and Frequency Domain Plot

```
subplot(3,1,1);
p = stem(n(trange), x(trange), 'k'); grid on;
xlabel('Time, t (s)'); ylabel('x(t)');

subplot(3,1,2);
p = stem(k(frange), abs(X(frange)), 'k');
set(p,'LineWidth',2,'MarkerSize',4); grid on;
xlabel('Harmonic number, k');
ylabel('|X(k)|');

subplot(3,1,3);
p = stem(k(frange), angle(X(frange)), 'k');
set(p,'LineWidth',2,'MarkerSize',4); grid on;
xlabel('Harmonic Number, k');
ylabel('Ang{X[k]}');

endfunction
```

wavplot

```
function wavplot(fname)

[x, fs, nbits] = wavread(fname);
nn = length(x);
printf('fs = %f \n', fs);
printf('nn = %d \n', nn);

n = (0 : nn-1) / fs ;
k = fs * (0 : nn-1) / nn ;
X = fft( x );

source "../0.util.octave/util.m"
#TimeFreqPlot(n, x, k, X, "half")
#TimeFreqPlot(n, x, k, X, (1:1024)+1024*0.5)
#TimeFreqPlot(n, x, k, X, (1*1024:2*1024))
#TimeFreqPlot(n, x, k, X, (35200-0.5*1024:35200+0.5*1024))
TimeFreqPlot(n, x, k, X, (2*1024:3*1024))


endfunction
```

References

[1] http://en.wikipedia.org/
[2] J.H. McClellan, et al., Signal Processing First, Pearson Prentice Hall, 2003
[3] M.J. Roberts, Fundamentals of Signals and Systems
[4] S.J. Orfanidis, Introduction to Signal Processing
[5] K. Shin, et al., Fundamentals of Signal Processing for Sound and Vibration Engineerings

[6] A “graphical interpretation” of the DFT and FFT, by Steve Mann