

DTFS Octave Codes (2A)

Copyright (c) 2009 - 2017 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

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

This document was produced by using OpenOffice and Octave.

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

DTFS example 1

```
NF = 4;  
n = [0:NF-1]';  
x = 8*sin(pi*n/2);  
X = fft(x)/NF;
```

X =

```
0.000000 + 0.000000i  
-0.000000 - 4.000000i  
0.000000 + 0.000000i  
-0.000000 + 4.000000i
```

DTFS example 2

```
NF = 8;  
n = [0:NF-1]';  
x = cos(2*pi*n/NF);  
X = fft(x)/NF;
```

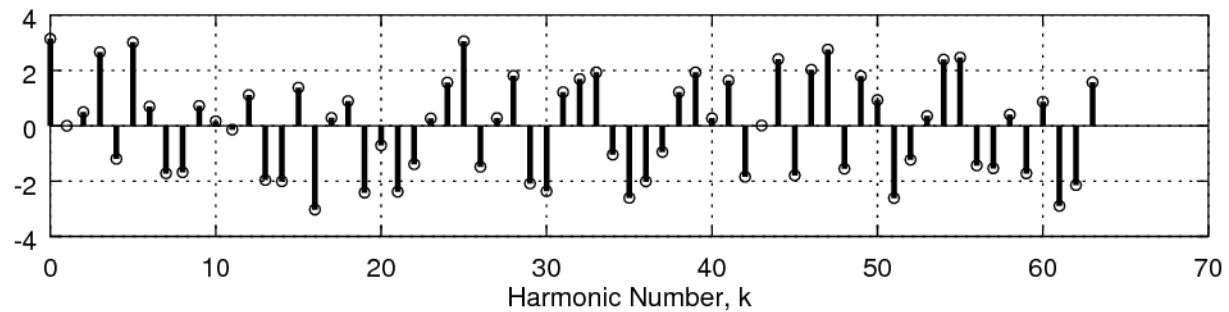
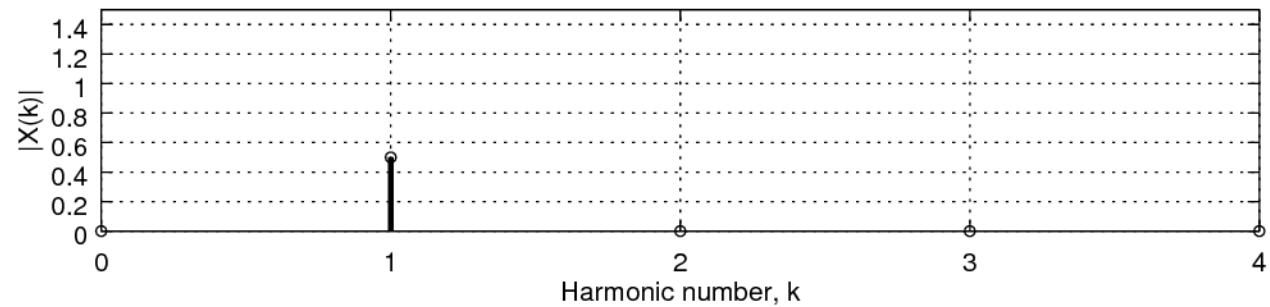
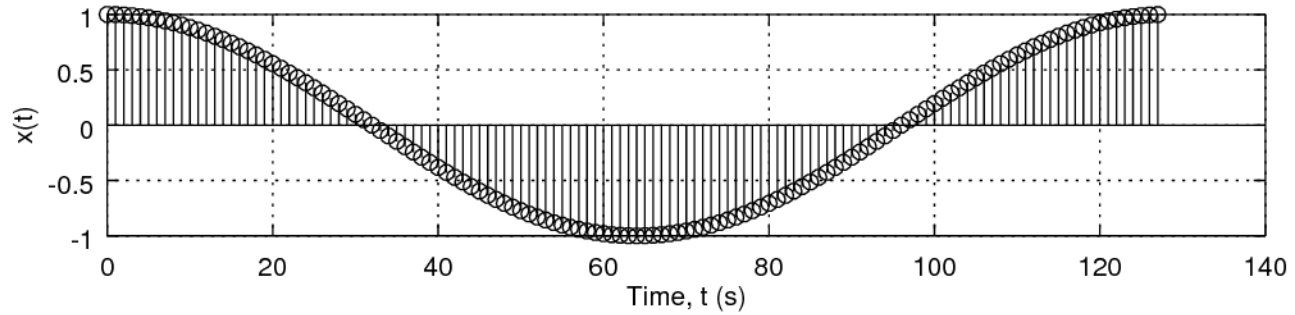
```
X =
```

```
-0.00000 + 0.00000i  
0.50000 - 0.00000i  
0.00000 + 0.00000i  
-0.00000 - 0.00000i  
0.00000 + 0.00000i  
-0.00000 + 0.00000i  
0.00000 - 0.00000i  
0.50000 + 0.00000i
```

DTFS example 3

```
NF =128;  
n = [0:NF-1]';  
x = cos(2*pi*n/NF);  
X = fft(x)/NF;  
  
k = [0:NF/2-1];  
  
TimeFreqPlot(n, x, k, X, NF);
```

DTFS example 3



Normalized ω_s and ω_0

```
N0 = 8;  
n = [0:N0-1];  
x = impD(n-3);  
X = fft(x)/N0;
```

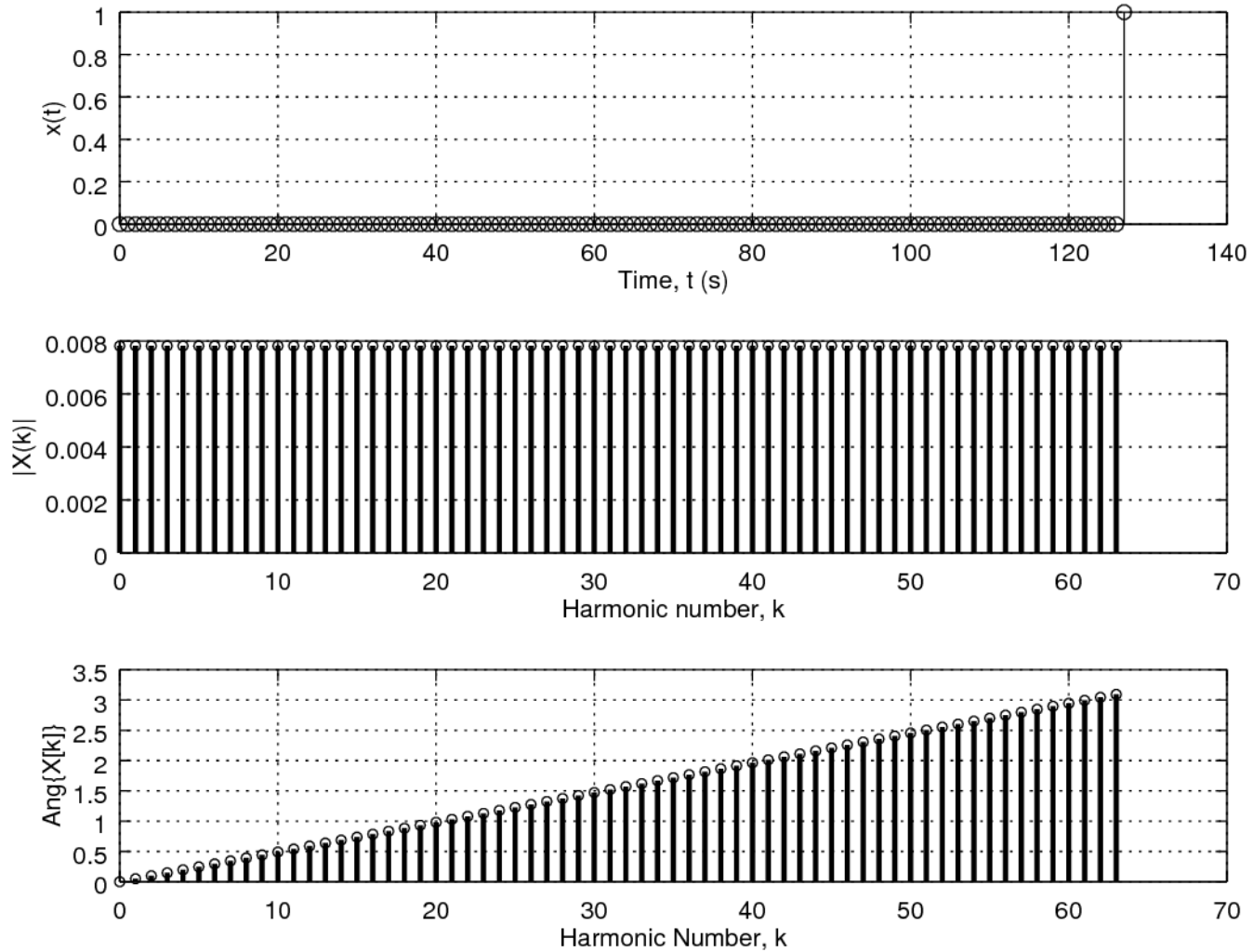
```
N0 = 10;  
K = [0:N0-1]';  
Y = impND(N0,k-2-1) + impND(N0,k-2+1) + impND(N0,k+2-1) + ...  
    impND(N0,k-2+1);  
y = ifft(Y)*N0;
```

```
N = [0:N0-1];  
Y = 4*cos(2*pi*n/10).*cos(2*pi*n/5);
```


Normalized ω_s and ω_0

```
source 'func.m'  
  
NF = 128;  
n = [0:NF-1];  
x = impD(n-127);  
X = fft(x)/NF;  
  
k = [0:NF/2-1];  
  
TimeFreqPlot(n, x, k, X, NF);
```

Normalized ω_s and ω_0



Normalized ω_s and ω_0

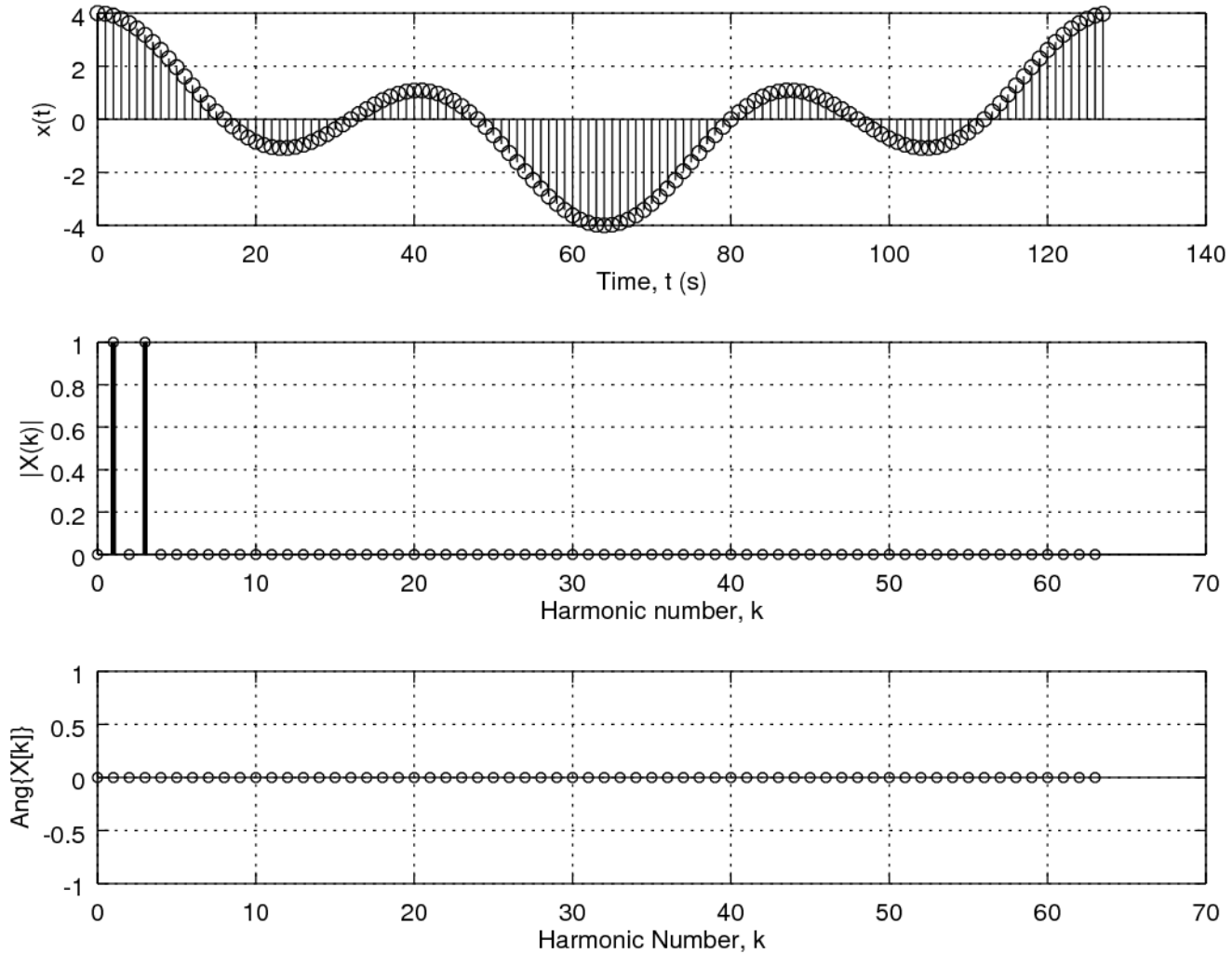
```
source 'func.m'

NF = 128;
k = [0:NF-1]';
Y = impND(NF,k-2-1) + impND(NF,k-2+1) + ...
    impND(NF,k+2-1) + impND(NF,k+2+1);
y = ifft(Y)*NF;

n = [0:NF-1]';
k = [0:NF/2-1]';
x = y;
X = Y(1:NF/2);

TimeFreqPlot(n, x, k, X, NF);
```

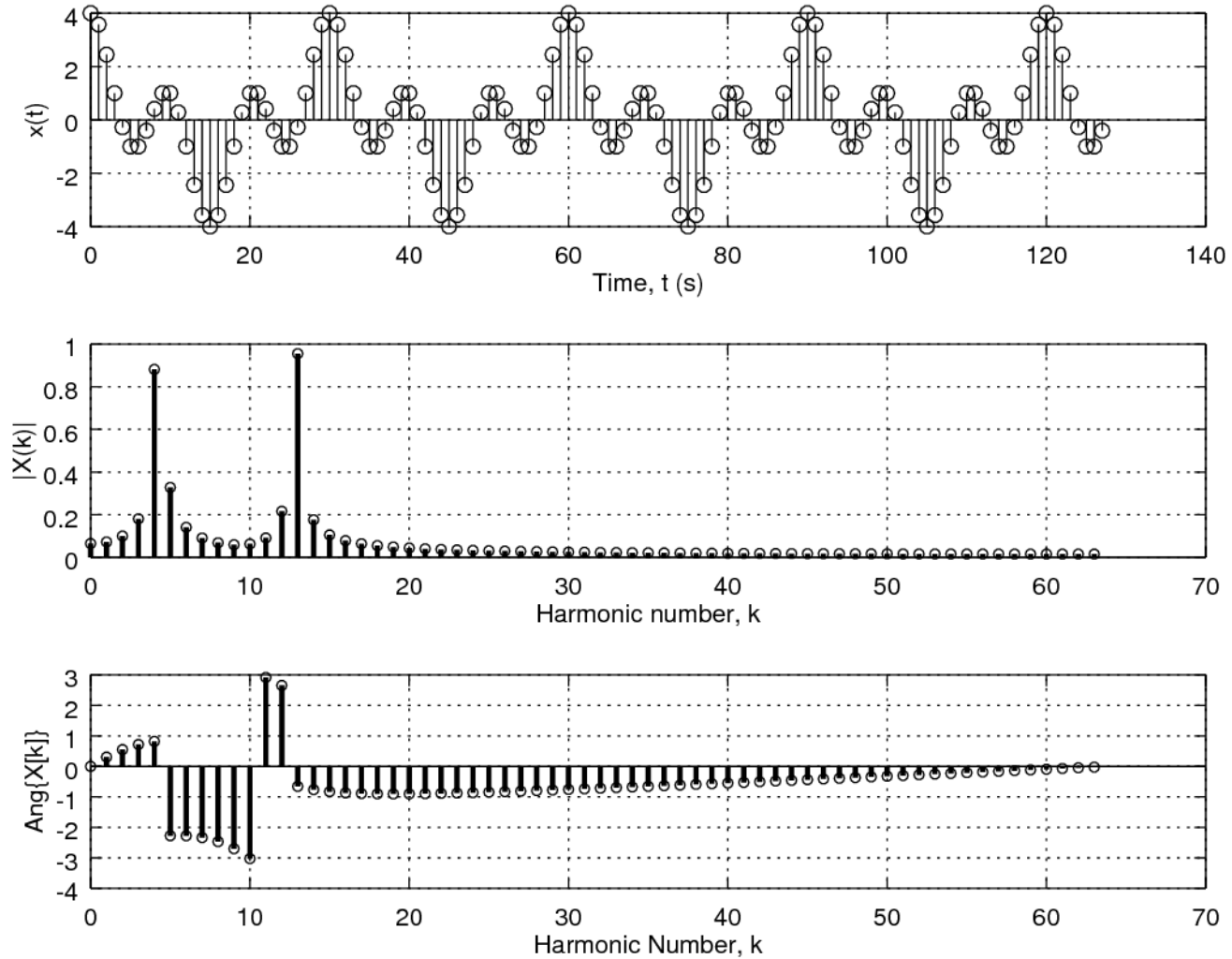
Normalized ω_s and ω_0



Normalized ω_s and ω_0

```
source 'func.m'  
  
NF = 128;  
n = [0:NF-1];  
x = 4*cos(2*pi*n/30) .* cos(2*pi*n/15);  
X = fft(x)/NF;  
  
k = [0:NF/2-1];  
  
TimeFreqPlot(n, x, k, X, NF);
```

Normalized ω_s and ω_0



Normalized ω_s and ω_0

```
N0 = 15;  
N = [0:N0-1];  
x = cos(2*pi*n/3) + impND(5,n);  
X = fft(x)/N0;  
X
```

```
N=[-10:40]'  
yp = 0*n  
for k = 0:3  
yp = yp + (0.25/(1.25 - exp(-j*2*pi*k/4))) * exp(j*2*pi*n*k);  
end
```

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