Lab 1 - Spectral Representation of Finite Length Signals

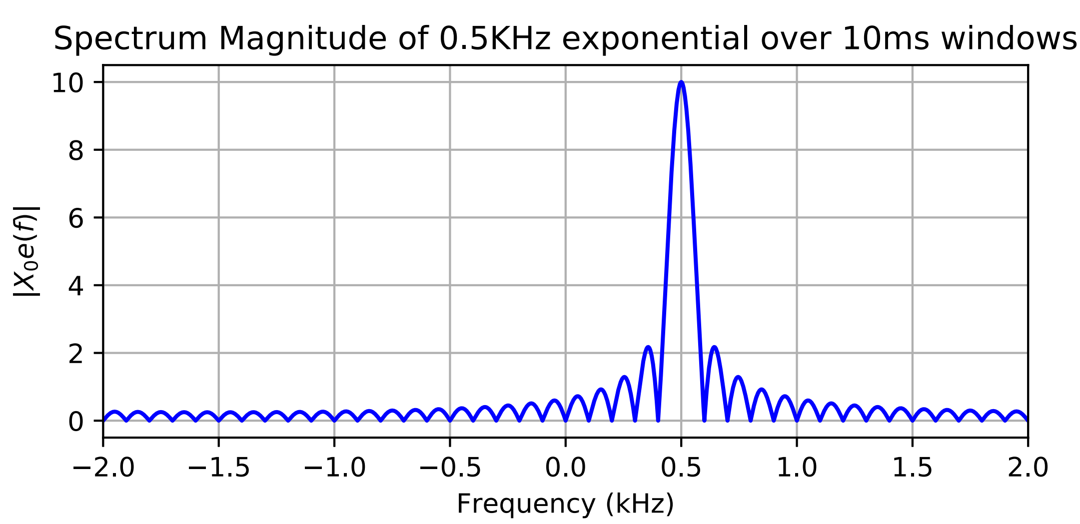
In [1]:
```
# importing necessary libraries
# Author : Jay Patel, Windowing.py
%pylab inline
import sk_dsp_comm.sigsys as ss
import scipy.signal as signal
from IPython.display import Image, SVG
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:
```
#Notebook configuration
pylab.rcParams['savefig.dpi'] = 100 # default 72
%config InlineBackend.figure_formats=['svg'] # SVG inline viewing
```

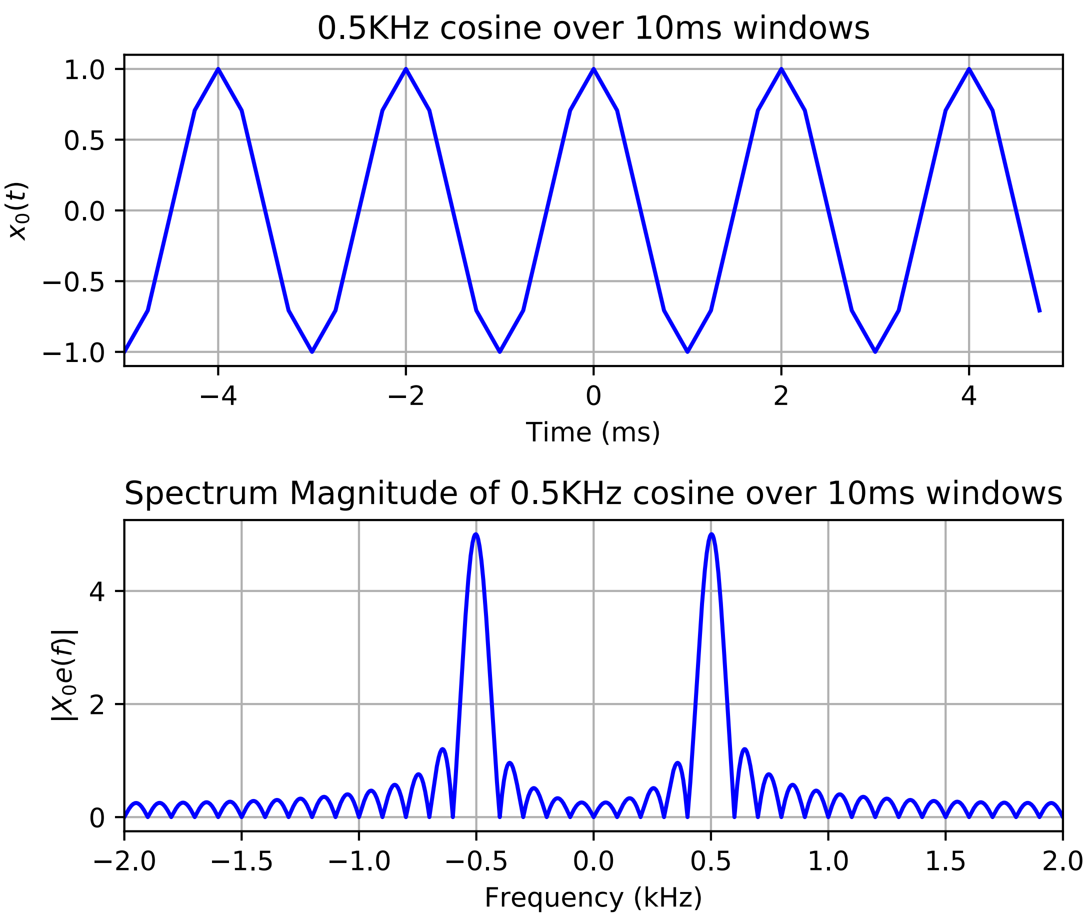
In [3]:
```
fs = 4 # sampling rate in kHz
t = arange(-5,5,1/fs)
tau = 1
f1 = 0.5; # Frequency component, in [kHz]
omega = 2*pi*f1;

x0 = exp(1j*omega*t)
```

In [4]:
```
figure(figsize=(6,5))
subplot(211)
# plot(t,x0.real,'b')
# grid()
# ylim([-1.1,1.1])
# xlim([-5,5])
# title(r'0.5KHz complex exponential over 10ms windows')
# xlabel(r'Time (ms)')
# ylabel(r'$x_0(t)$');

# FT Exact Plot
# subplot(212)
f,X0 = ss.ft_approx(x0,t,2000)
plot(f,abs(X0),'b')
#plot(f,angle(X0))
grid()
xlim([-2,2])
title(r'Spectrum Magnitude of 0.5KHz exponential over 10ms windows')
xlabel(r'Frequency (kHz)')
ylabel(r'$|X_0e(f)|$');
tight_layout()
```



In [5]:
```
x1 = cos(omega*t);
```

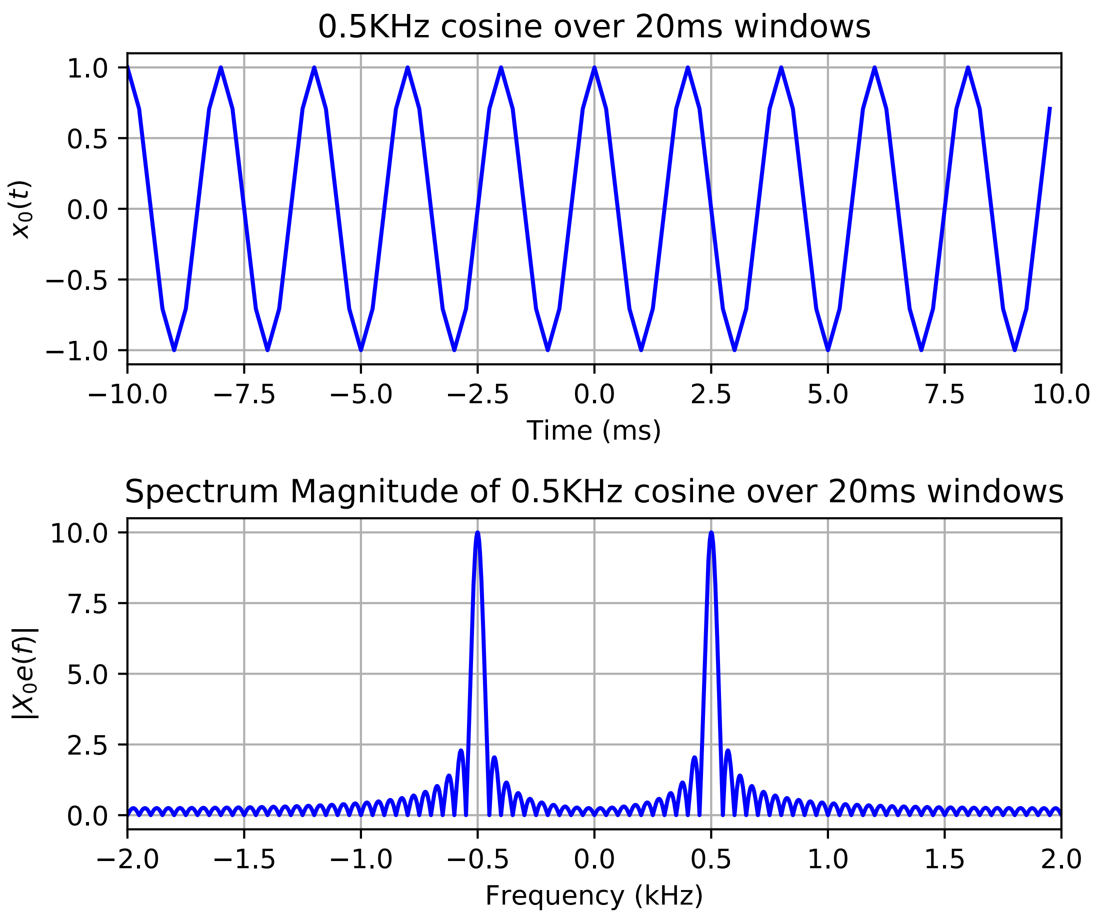
In [6]:
```
figure(figsize=(6,5))
subplot(211)
plot(t,x1.real,'b')
grid()
ylim([-1.1,1.1])
xlim([-5,5])
title(r'0.5KHz cosine over 10ms windows')
xlabel(r'Time (ms)')
ylabel(r'$x_0(t)$');

# FT Exact Plot
f,X1 = ss.ft_approx(x1,t,2000)
subplot(212)
plot(f,abs(X1),'b')
#plot(f,angle(X0))
grid()
xlim([-2,2])
title(r'Spectrum Magnitude of 0.5KHz cosine over 10ms windows')
xlabel(r'Frequency (kHz)')
ylabel(r'$|X_0e(f)|$');
tight_layout()
```



In [7]:
```
t = arange(-10,10,1/fs)
```

In [8]:
```
x2 = cos(omega*t);
```

In [9]:
```
figure(figsize=(6,5))
subplot(211)
plot(t,x2.real,'b')
grid()
ylim([-1.1,1.1])
xlim([-10,10])
title(r'0.5KHz cosine over 20ms windows')
xlabel(r'Time (ms)')
ylabel(r'$x_0(t)$');

# FT Exact Plot
f,X2 = ss.ft_approx(x2,t,2000)
subplot(212)
plot(f,abs(X2),'b')
#plot(f,angle(X0))
grid()
xlim([-2,2])
title(r'Spectrum Magnitude of 0.5KHz cosine over 20ms windows')
xlabel(r'Frequency (kHz)')
ylabel(r'$|X_0e(f)|$');
tight_layout()
```



In [10]:
```
fs = 4 # sampling rate in kHz
W = 5
t = arange(-5,5,1/fs)
x4 = W/pi*sinc(W/pi*t)
figure(figsize=(6,2))
plot(t,x4,'b')
grid()
# ylim([-1.1,1.1])
xlim([-5,5])
title(r'Time Domain: $x_4(t),\ W = 5$ Hz')
xlabel(r'Time (s)')
ylabel(r'$x_4(t)$');
f,X4 = ss.ft_approx(x4,t,2000)
figure(figsize=(6,2))
plot(f,abs(X4),'b')
grid()
title(r'Frequency Domain: $X_4(f)$')
xlim([-1,1])
xlabel(r'Frequency (Hz)')
ylabel(r'$|X_4(f)|$');
figure(figsize=(6,2))
plot(f,20*log10(abs(X4)),'b')
grid()
title(r'Frequency Domain: $X_4(f)$ in dB')
ylim([-50,5])
xlim([-1,1])
xlabel(r'Frequency (Hz)')
ylabel(r'$|X_4(f)|$ (dB)');
```