Systems & Signals 414

Practical 2

D.S. van der Westhuizen - 17158680

In [1]:

```
#All the necessary imports
%matplotlib inline
import pylab as pl
pl.style.use('bmh') #pretty plots
pl.rcParams['figure.figsize'] = (13, 2)
import numpy as np
import scipy.fftpack

#pl.figure()
#pl.title(r"$f_s \ldots$ Hz") #Try not to have
#extremely long lines in your file, like this one please
#pl.ylabel("...")
#pl.xlabel("...")
#pl.stem(np.abs(np.arange(-5,5)));

def zpad(signal, pad_left_right):
    return np.pad(signal, pad_left_right, 'constant', constant_values=0)
```

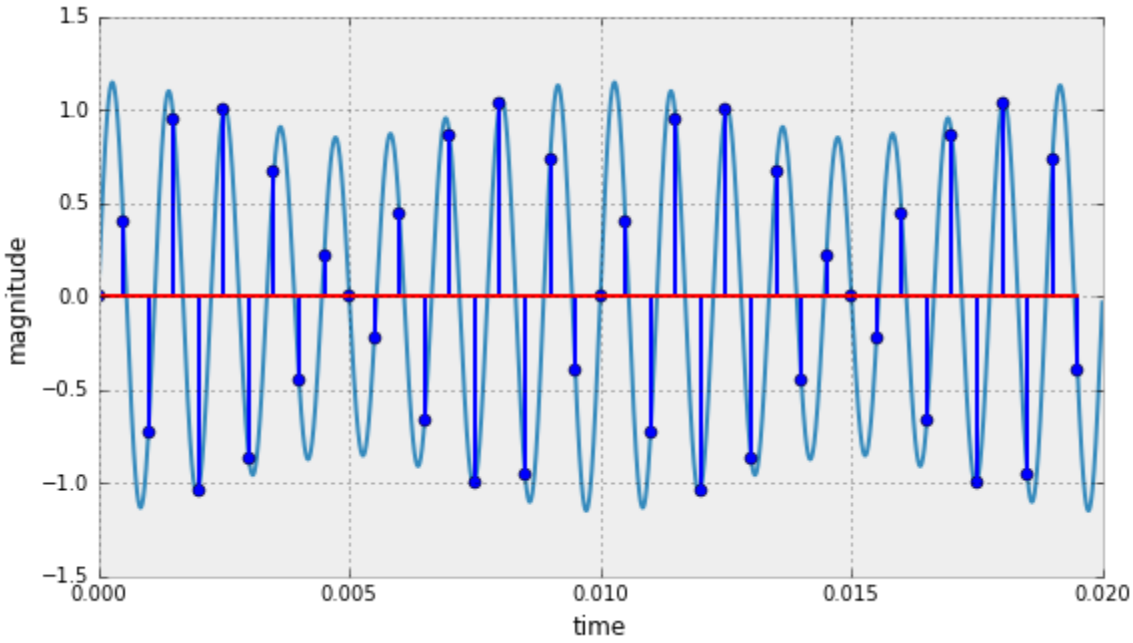
Section 1

In [2]:

```
%matplotlib inline
pl.figure(figsize=(9,5))
pl.ylabel("magnitude")
pl.xlabel("time")
t = np.linspace(0,0.02,4000,False)
x = np.sin(900*2*np.pi*t)+ 0.15*np.sin(800*2*np.pi*t)
pl.plot(t,x);
t2 = np.linspace(0,0.02,40,False)
x2 = np.sin(900*2*np.pi*t2)+ 0.15*np.sin(800*2*np.pi*t2)
pl.stem(t2,x2)
```

Out[2]:

<Container object of 3 artists>

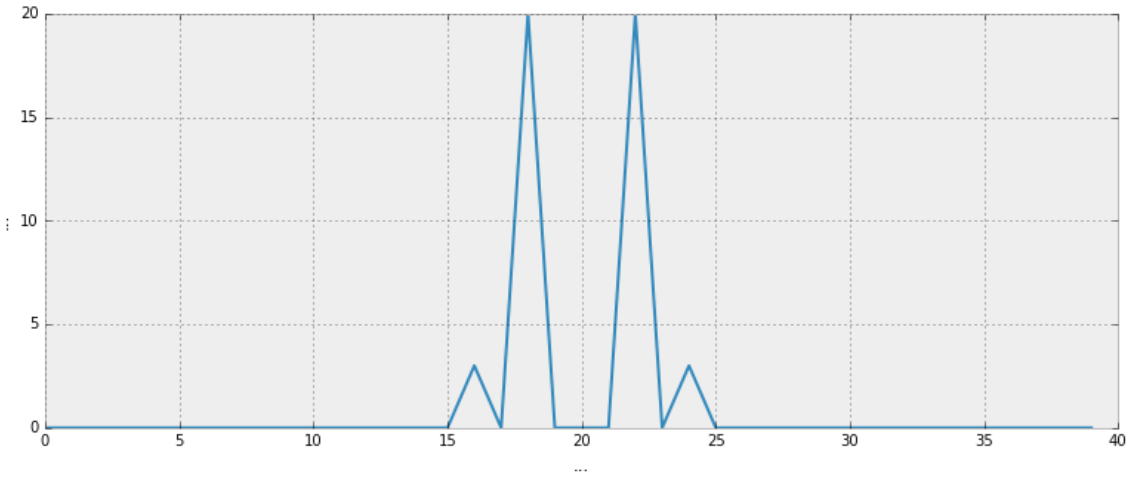


In [3]:

```
%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("...")
pl.xlabel("...")
X2 = np.fft.fft(x2)
pl.plot(np.abs(X2))
```

Out[3]:

```
[<matplotlib.lines.Line2D at 0x74c3ef0>]
```



The longer spike: 18/0.02 = 900Hz

The shorter spike: 16/0.02 = 820Hz

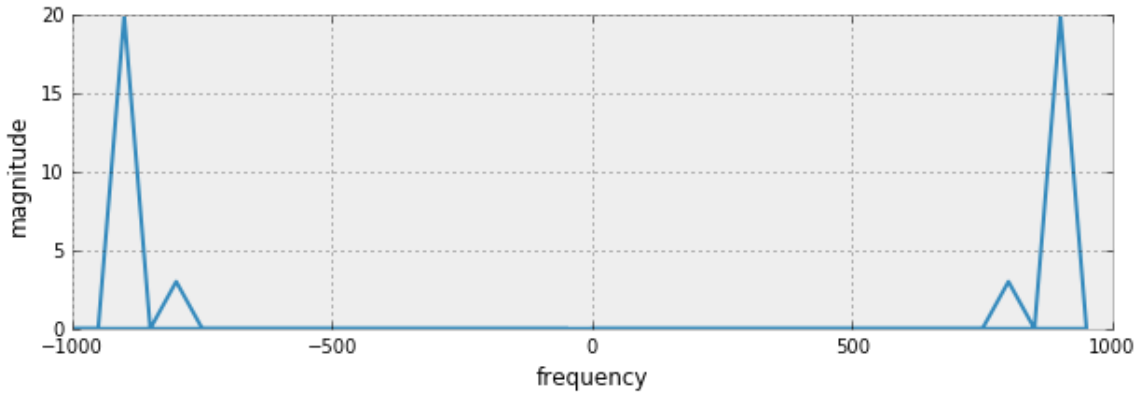
The frequency graph below confirms it.

In [4]:

```
%matplotlib inline
pl.figure(figsize=(10,3))
pl.ylabel("magnitude")
pl.xlabel("frequency")
freq = np.fft.fftfreq(len(x2), t2[1] - t2[0])
pl.plot(freq,np.abs(X2))
```

Out[4]:

```
[<matplotlib.lines.Line2D at 0x752e668>]
```



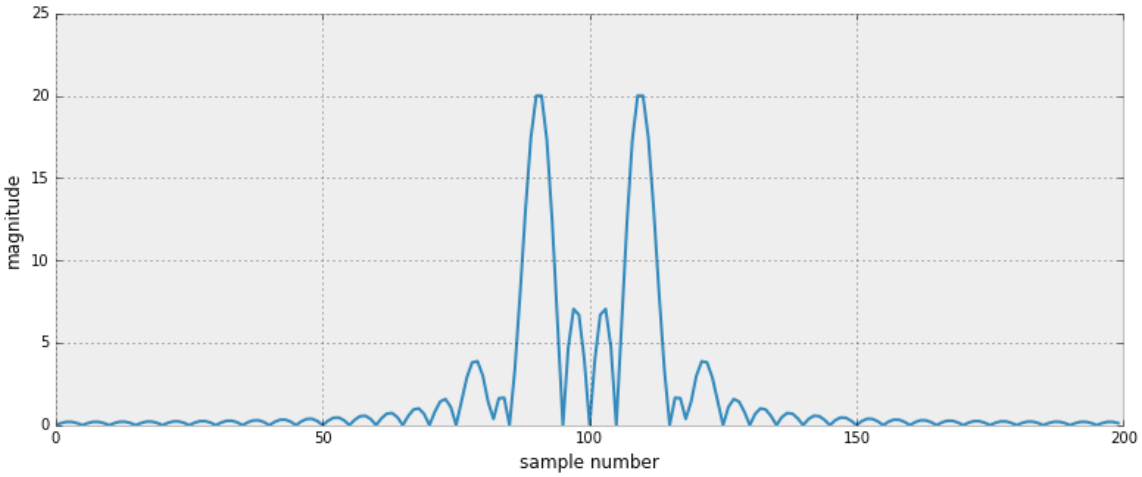
In [5]:

```
x3 = zpad(x2,(0,160))

%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
X3 = np.fft.fft(x3)
pl.plot(np.abs(X3))
```

Out[5]:

```
[<matplotlib.lines.Line2D at 0x7583cf8>]
```



The original FFT has been interpolated. The zero padding causes the new sampled signal to be the equivalent of a windowing over the nonzero region (a region over which the original signal has been multiplied by block function. The FFT of this new signal is a sinc function convoluted with the original sampled signal's FFT. This is what is portrayed in the graph above.

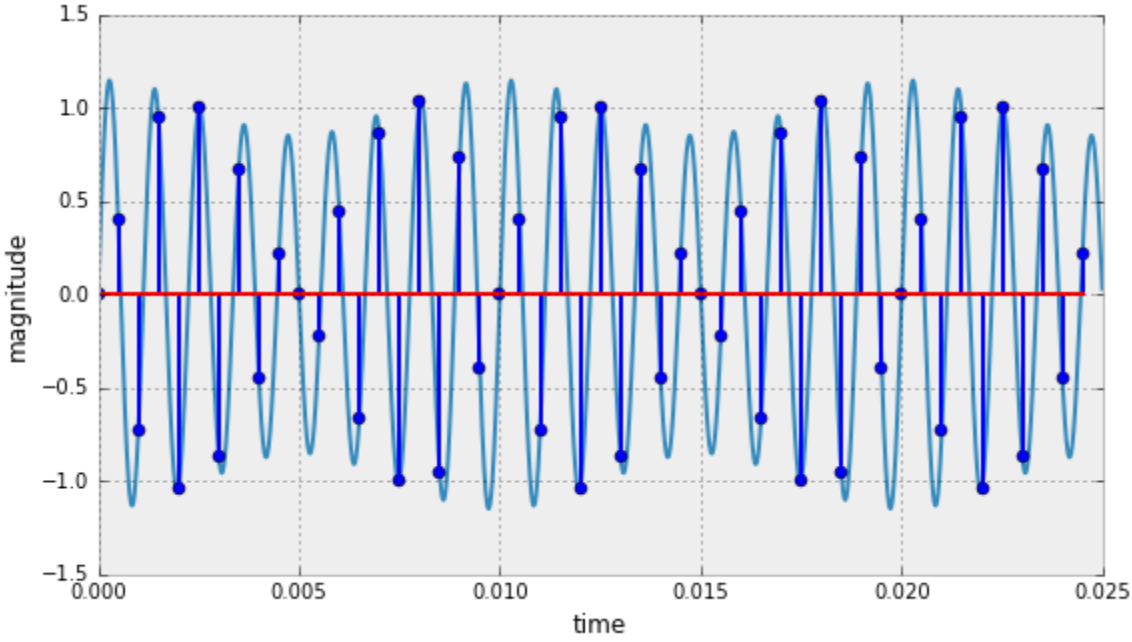
Section 2

In [6]:

```
%matplotlib inline
pl.figure(figsize=(9,5))
pl.ylabel("magnitude")
pl.xlabel("time")
t4 = np.linspace(0,0.025,4000,False)
x4 = np.sin(900*2*np.pi*t4)+ 0.15*np.sin(800*2*np.pi*t4)
pl.plot(t4,x4);
t5 = np.linspace(0,0.025,50,False)
x5 = np.sin(900*2*np.pi*t5)+ 0.15*np.sin(800*2*np.pi*t5)
pl.stem(t5,x5)
```

Out[6]:

```
<Container object of 3 artists>
```

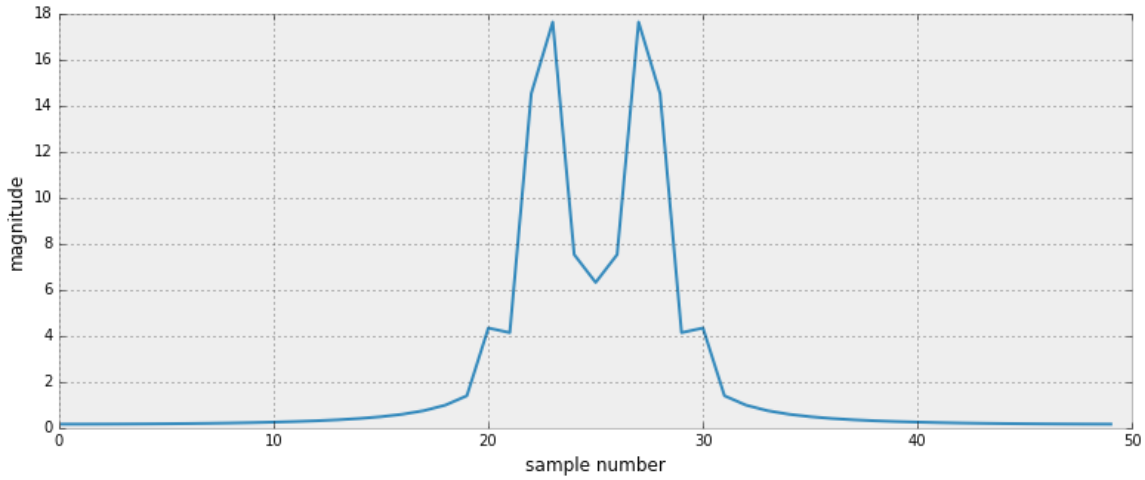


In [7]:

```
%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
X5 = np.fft.fft(x5)
pl.plot(np.abs(X5))
```

Out[7]:

```
[<matplotlib.lines.Line2D at 0x7bf6a58>]
```



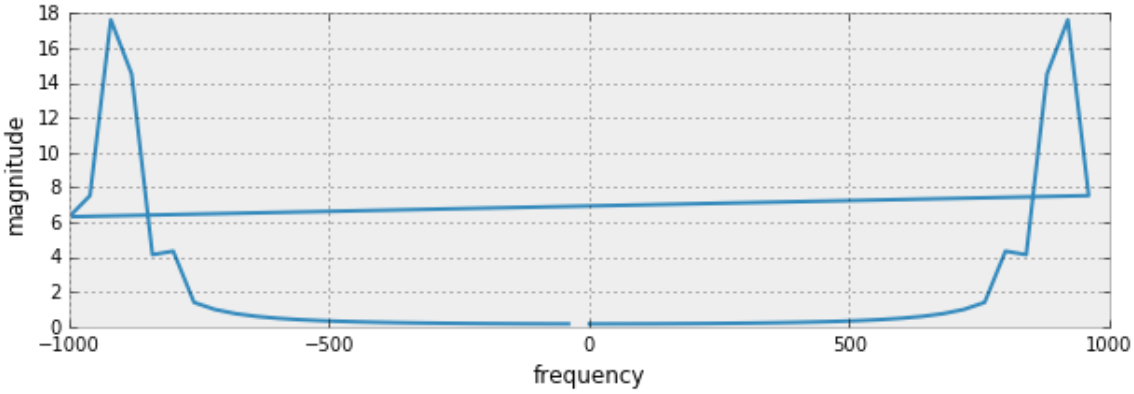
^I don't know

In [8]:

```
%matplotlib inline
pl.figure(figsize=(10,3))
pl.ylabel("magnitude")
pl.xlabel("frequency")
freq5 = np.fft.fftfreq(len(x5), t5[1] - t5[0])
pl.plot(freq5,np.abs(X5))
```

Out[8]:

```
[<matplotlib.lines.Line2D at 0x7cca4a8>]
```



^this doesn't help

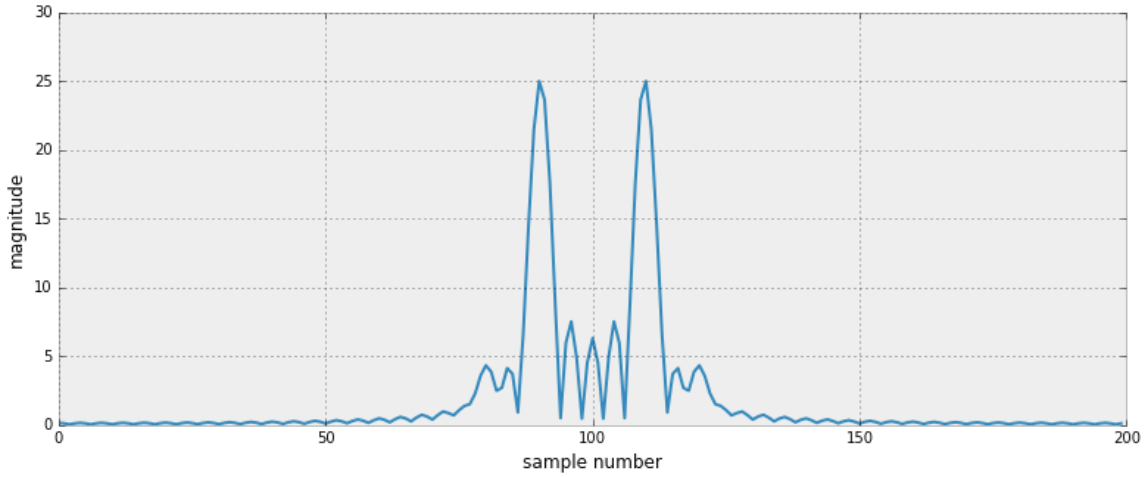
In [9]:

```
x6 = zpad(x5,(0,150))

%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
X6 = np.fft.fft(x6)
pl.plot(np.abs(X6))
```

Out[9]:

```
[<matplotlib.lines.Line2D at 0x8cfccf8>]
```



^Yes, it's clearer now.

Hamming window

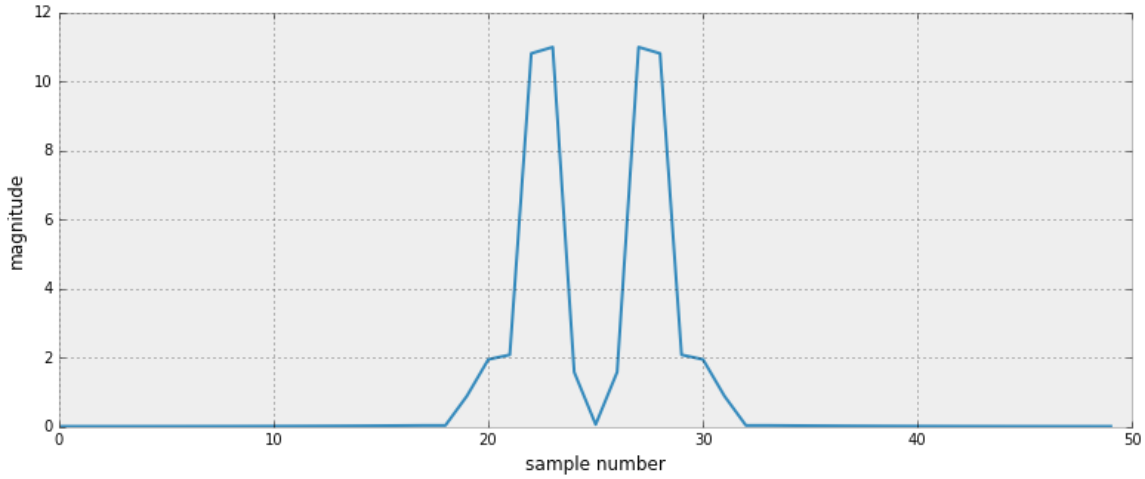
In [10]:

```
x7 = x5*np.hamming(50)

%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
X7 = np.fft.fft(x7)
pl.plot(np.abs(X7))
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0x8d1f550>]
```



Hamming with zeropad

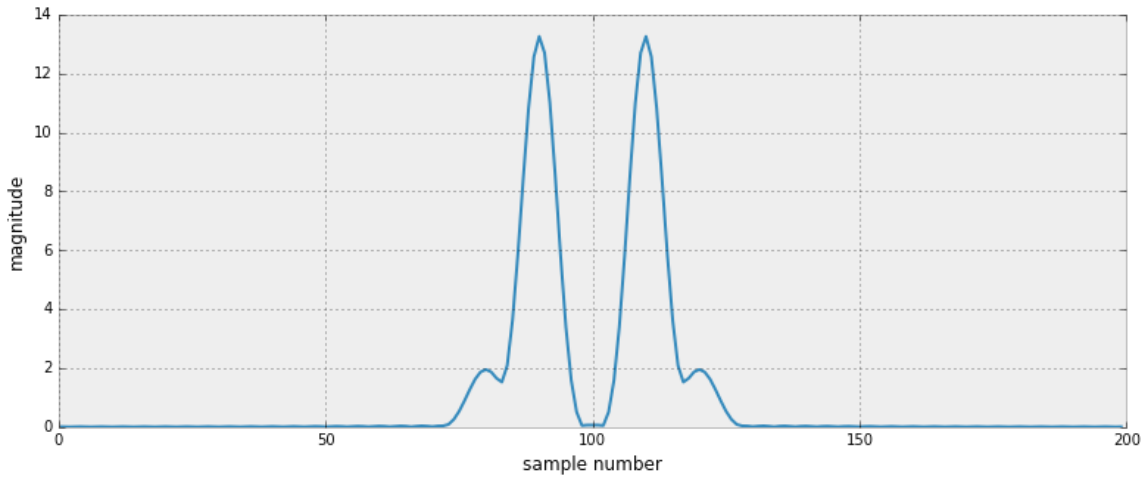
In [11]:

```
x8 = zpad(x7,(0,150))

%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
X8 = np.fft.fft(x8)
pl.plot(np.abs(X8))
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x8d86358>]
```



^Yes, it's clearer now. I don't know why.

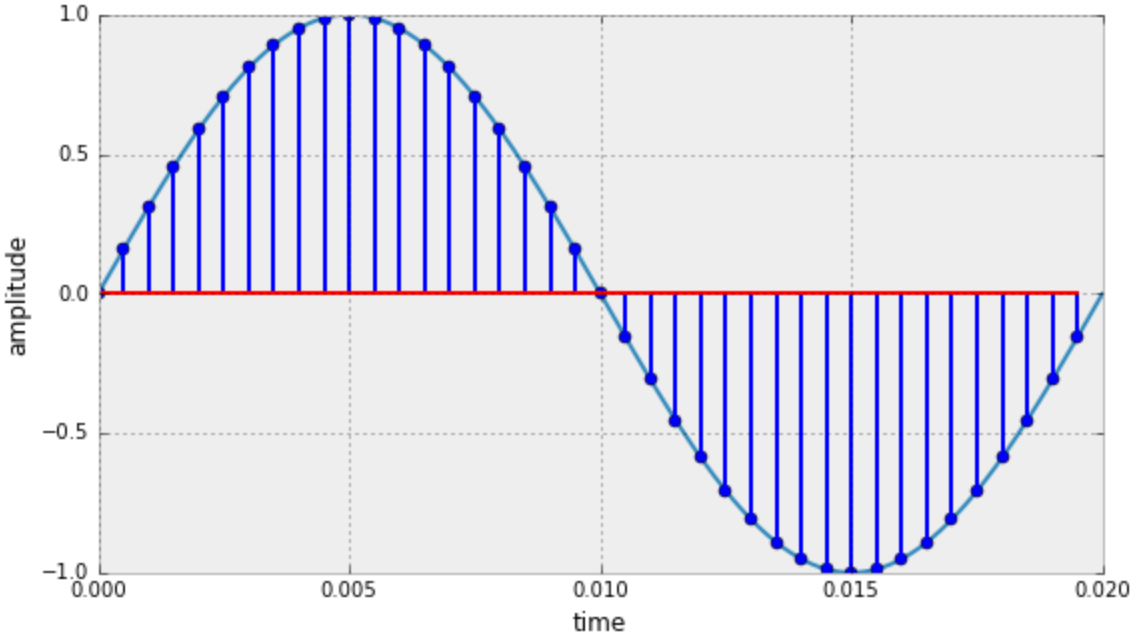
Section 3

In [12]:

```
%matplotlib inline
pl.figure(figsize=(9,5))
pl.ylabel("amplitude")
pl.xlabel("time")
t10 = np.linspace(0,0.02,4000,False)
x10 = np.sin(50*2*np.pi*t10)
pl.plot(t10,x10);
t11 = np.linspace(0,0.02,40,False)
x11 = np.sin(50*2*np.pi*t11)
pl.stem(t11,x11)
```

Out[12]:

<Container object of 3 artists>

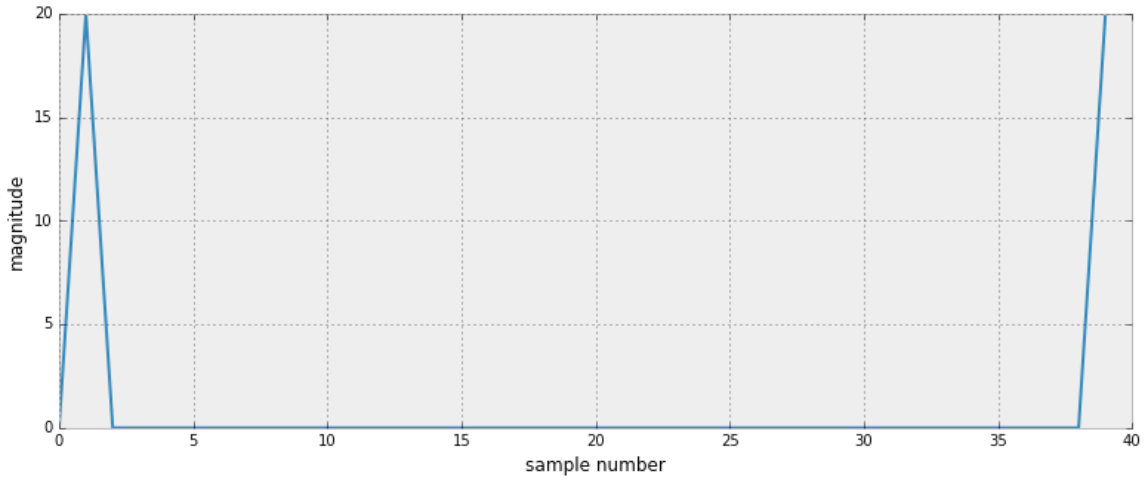


In [13]:

```
%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
X11 = np.fft.fft(x11)
pl.plot(np.abs(X11))
```

Out[13]:

```
[<matplotlib.lines.Line2D at 0x8f37908>]
```

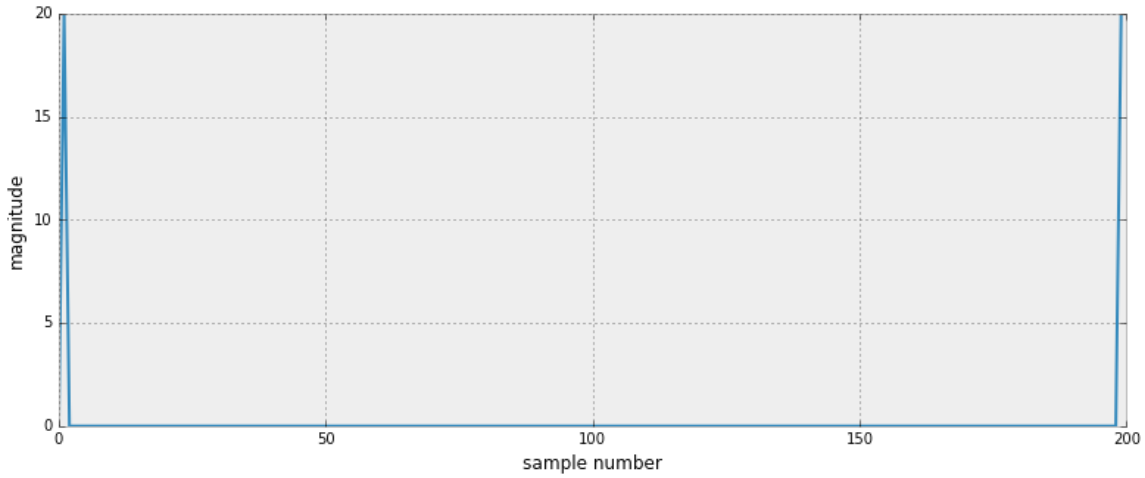


In [14]:

```
u = X11[0:20]
u1 = zpad(u,(0,160))
X12 = np.append(u1,X11[20:40])
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.abs(X12))
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x8f9f780>]
```

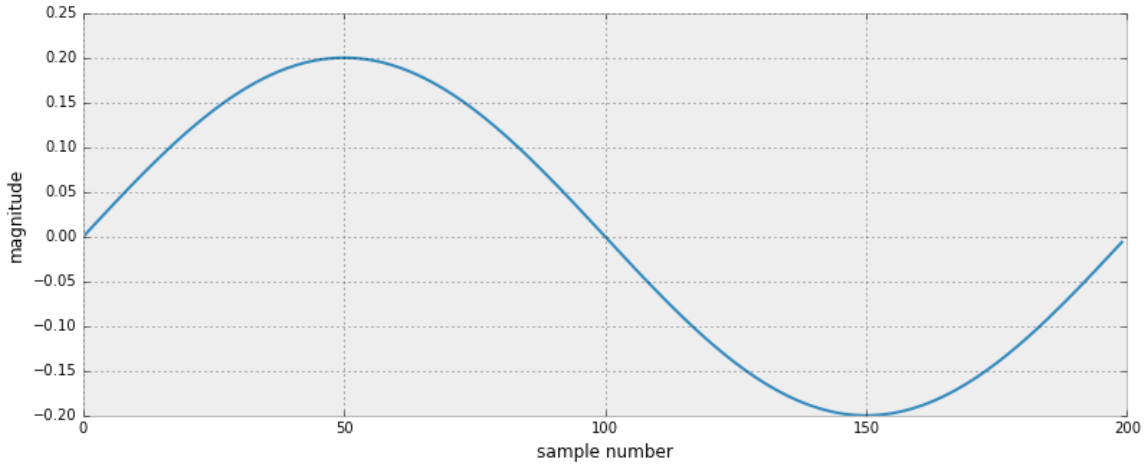


In [15]:

```
x12 = np.fft.ifft(X12)
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.real(x12))
```

Out[15]:

```
[<matplotlib.lines.Line2D at 0x8fbb978>]
```



^The original signal

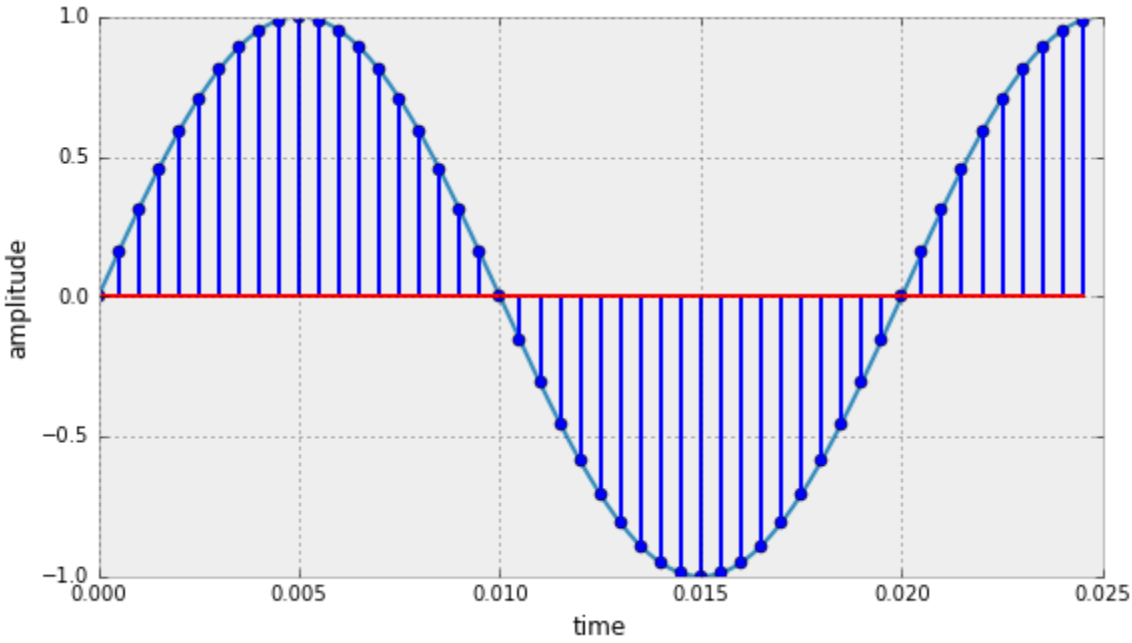
Section 4

In [16]:

```
%matplotlib inline
pl.figure(figsize=(9,5))
pl.ylabel("amplitude")
pl.xlabel("time")
t15 = np.linspace(0,0.025,5000,False)
x15 = np.sin(50*2*np.pi*t15)
pl.plot(t15,x15);
t16 = np.linspace(0,0.025,50,False)
x16 = np.sin(50*2*np.pi*t16)
pl.stem(t16,x16)
```

Out[16]:

<Container object of 3 artists>

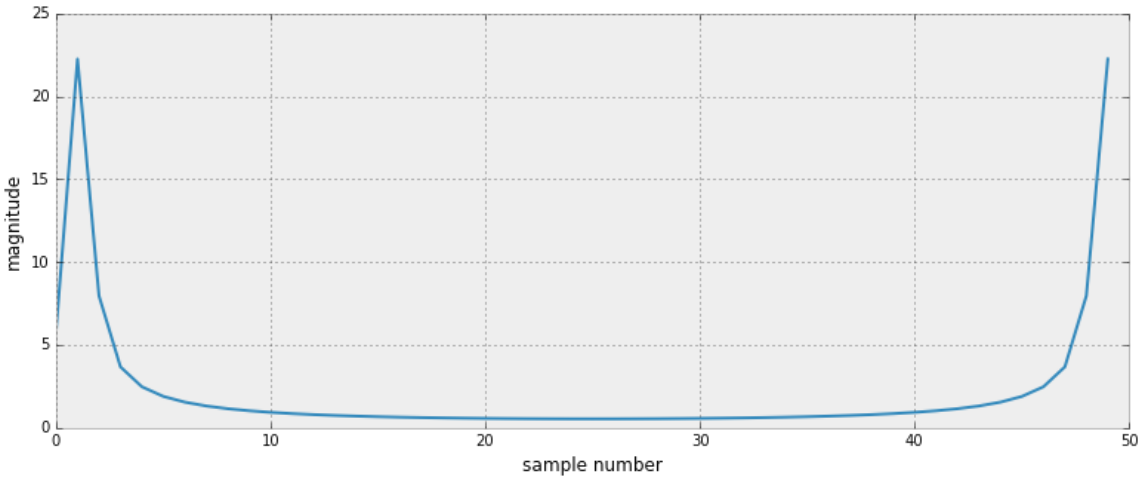


In [17]:

```
%matplotlib inline
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
X16 = np.fft.fft(x16)
pl.plot(np.abs(X16))
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x90f4710>]
```

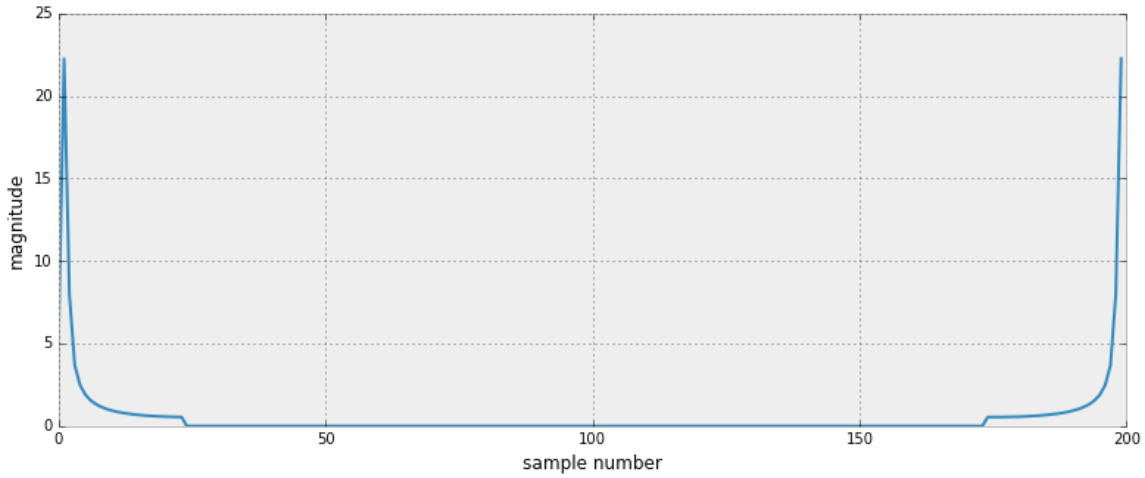


In [18]:

```
u4 = X16[0:24]
u5 = zpad(u4,(0,150))
X17 = np.append(u5,X16[24:50])
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.abs(X17))
```

Out[18]:

```
[<matplotlib.lines.Line2D at 0x93090f0>]
```

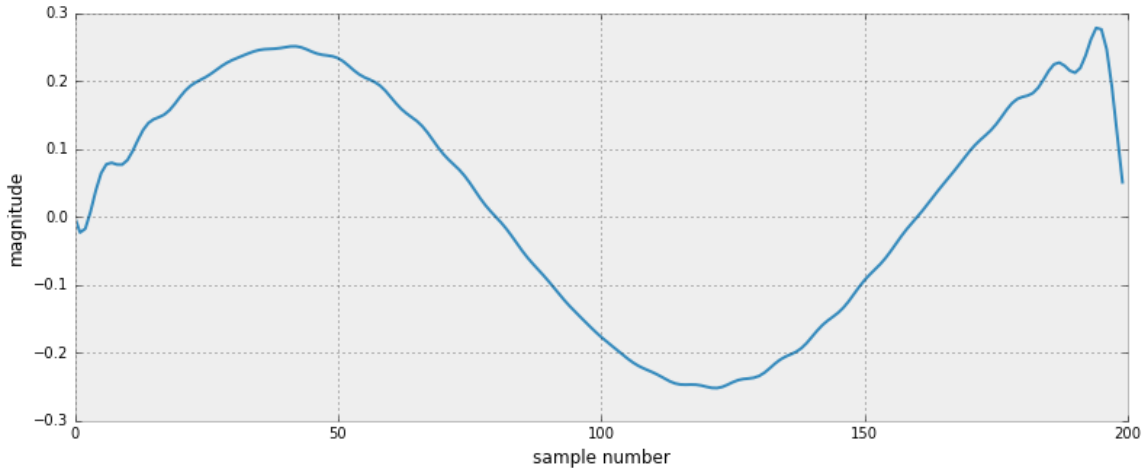


In [19]:

```
x17 = np.fft.ifft(X17)
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.real(x17))
```

Out[19]:

```
[<matplotlib.lines.Line2D at 0x9320da0>]
```

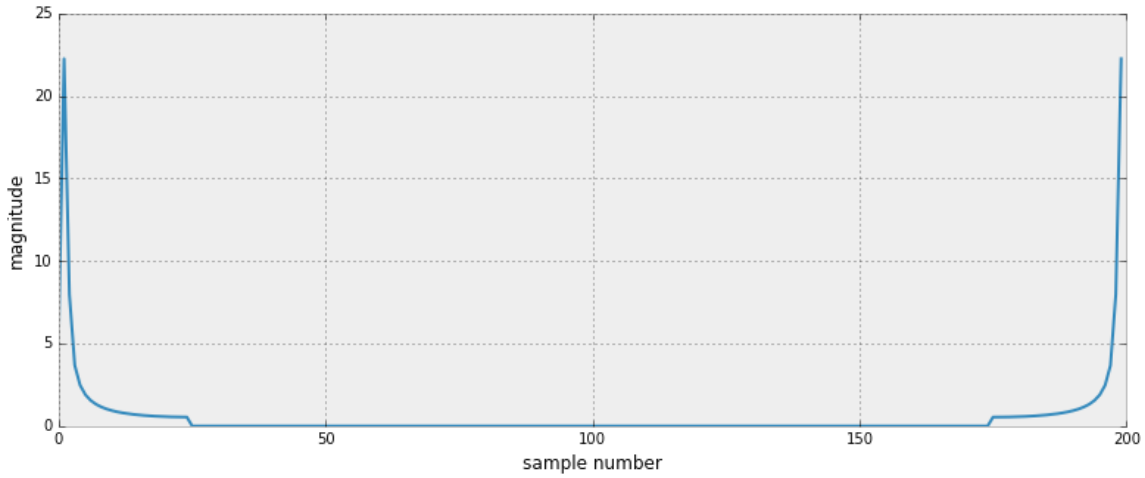


In [20]:

```
u6 = X16[0:25]
u7 = zpad(u6,(0,150))
X18 = np.append(u7,X16[25:50])
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.abs(X18))
```

Out[20]:

```
[<matplotlib.lines.Line2D at 0x938a5c0>]
```

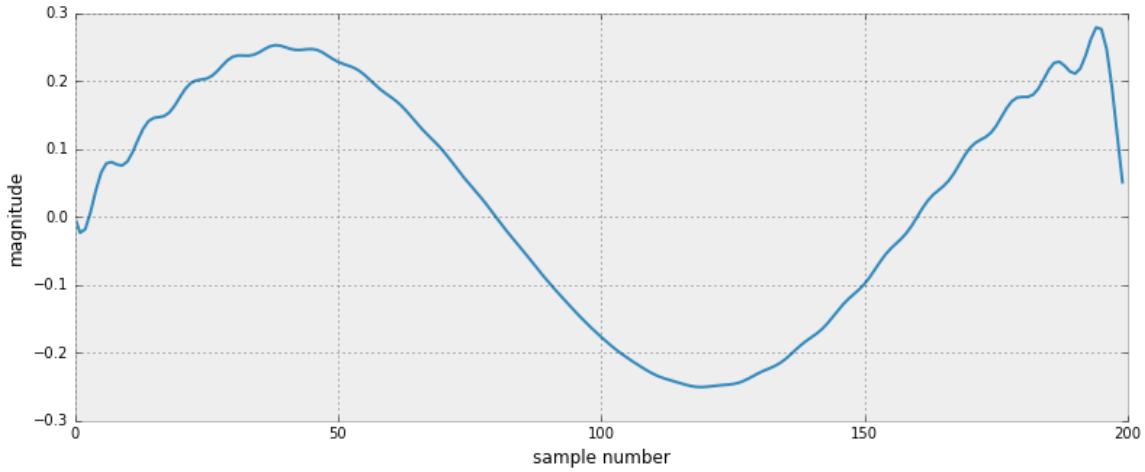


In [21]:

```
x18 = np.fft.ifft(X18)
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.real(x18))
```

Out[21]:

```
[<matplotlib.lines.Line2D at 0x93ea240>]
```

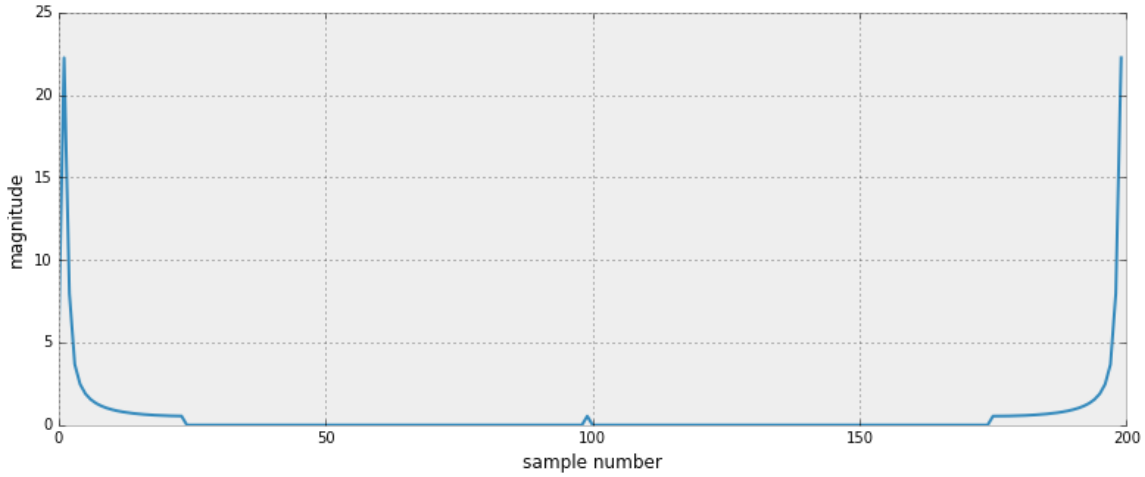


In [22]:

```
u10 = X16[0:24]
u11 = zpad(u10,(0,75))
u12 = np.append(u11,X16[25])
u13 = zpad(u12,(0,75))
X19 = np.append(u13,X16[25:50])
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.abs(X19))
```

Out[22]:

```
[<matplotlib.lines.Line2D at 0x944dcc0>]
```

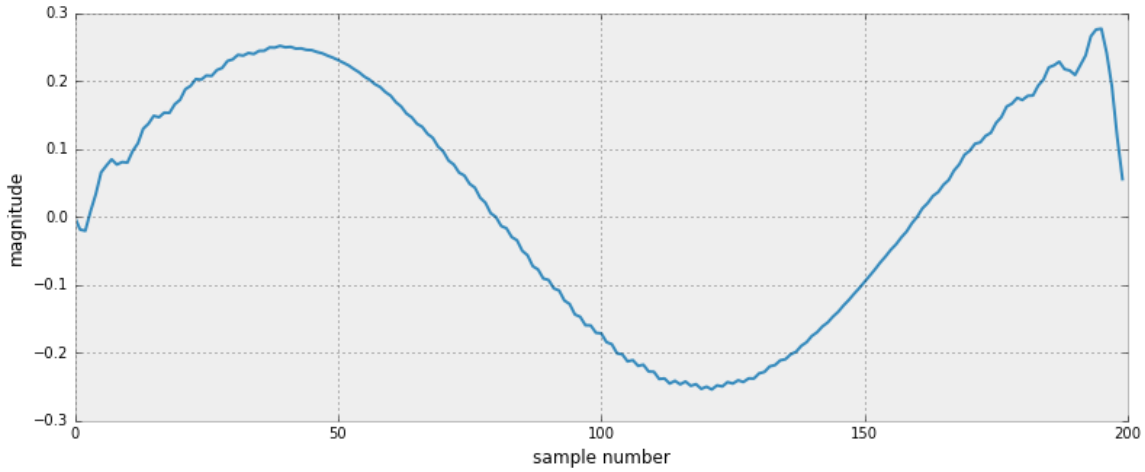


In [23]:

```
x19 = np.fft.ifft(X19)
pl.figure(figsize=(13,5))
pl.ylabel("magnitude")
pl.xlabel("sample number")
pl.plot(np.real(x19))
```

Out[23]:

```
[<matplotlib.lines.Line2D at 0x9741748>]
```



In [ ]: