1 FFT Example

A simple set of examples of Aliasing and applying the FFT

In [1]: %matplotlib inline
   import matplotlib.pyplot as plt
   import numpy as np

2 Aliasing

Create a simple sinusoidal function with frequency of 10 Hz with a large sampling frequency ($f_s = 1$ kHz).

In [2]: fs = 1000.  # Sampling frequency (in Hz)
   Ts = 1/fs    # delta t (spacing between each time point)
   Range = 1.  # Total range of data (in seconds)

   f0 = 10.0
   t0 = np.arange(0., Range, Ts)  # Point every 1 ms
   y0 = np.cos(2*np.pi*f0*t0)+1

   plt.plot(t0,y0)
   #plt.plot(t0,y0, 'o')
   plt.xlabel('Time (s)'
   plt.ylabel('Amplitude')
   plt.show()
Plot the exact same function, but now sampled at \( f_s = 10.5 \text{ Hz} \), illustrating aliasing when \( f_0 > 2f_s \).

In [3]: # First plot the original function
   plt.plot(t0, y0)

   fs = 10.5  # Reduce sampling frequency a lot
   Ts = 1/fs

   t1 = np.arange(0., Range, Ts)
   y1 = np.cos(2*np.pi*f0*t1)+1
   plt.plot(t1, y1, 'ro')
   plt.plot(t1, y1, 'r')
   plt.xlabel('Time')
   plt.ylabel('Amplitude')
   plt.show()
3 FFT Example

Now sample at a reasonable rate (above the Nyquist criteria) with $f_s = 50$ Hz over a fixed range of $\Delta t = 1$s. Note, the sampling actually looks quite scrappy.

In [4]: # Again, plot the original function
   plt.plot(t0,y0)

   $fs = 50$. # Sample this at 50 Hz
   $Ts = 1/fs$

   $t1 = np.arange(0., \text{Range}, Ts)$
   $y1 = np.cos(2*\text{pi}*f0*t1)+1$
   plt.plot(t1,y1,'ro')
   plt.xlabel('Time')
   plt.ylabel('Amplitude')
   plt.show()
Plot the FFT transform of this data series. Even though the sampling looked pretty poor above, the algorithm finds the right frequency. Since the input function is even, only the real components are non-zero. Note the frequencies are positive and negative, and run from zero to the Nyquist limit.

In [5]: # Discrete Fourier transform
    # length of the signal
    f1 = np.fft.fft(y1)/n # fft computing and normalization
Plot amplitude by taking the absolute value of the fft output, and adding together the positive and negative frequencies. Note the point at frequency of 0 only occurs once, as does the point at the Nyquist frequency.

```python
In [6]: nlim = n//2-1
```

Plot amplitude by taking the absolute value of the fft output, and adding together the positive and negative frequencies. Note the point at frequency of 0 only occurs once, as does the point at the Nyquist frequency.