

In [24]: `#@Author: Mathuranathan Viswanathan: www.gaussianwaves.com`

```
from numpy import sinc, arange, pi, zeros, ones, hstack
import numpy as np

from scipy.signal import freqz, convolve
import matplotlib.pyplot as plt
# comment out the next line if you are using Python command line
%matplotlib inline
plt.style.use('ggplot')

fig, ax = plt.subplots(ncols=3, nrows=3) #creating plot axes

# Design a Low pass brick wall filter (sinc filter in time domain)
h = 0.4*sinc(0.4*arange(-25,25)) # cut-off at 0.4pi rad/second
W,H = freqz(h,1) # frequency response of the filter

# unit impulse through the band-limited Low pass channel
x = hstack((zeros((1,1)),ones((1,1)),zeros((1,1)))).flatten()
y = convolve(h,x); # output passed through the band-limited brickwall channel
ax[0,0].stem(x,use_line_collection=True);ax[0,0].set_title('Input')
ax[0,1].plot(W/pi,abs(H));ax[0,1].set_title('Channel freq. resp.')
ax[0,2].plot(y);ax[0,2].set_title('Output')

# Low data rate bits through the band-limited Low pass channel
# Low data rate implies sufficient distance between 1's and -1's
x = hstack((zeros((1,1)),ones((1,1)),zeros((1,20)),-1*ones((1,1)),zeros((1,20)),ones((1,1)),zeros((1,1)))).flatten()
y = convolve(h,x); # output passed through the band-limited brickwall channel
ax[1,0].stem(x,use_line_collection=True)
ax[1,1].plot(W/pi,abs(H))
ax[1,2].plot(y)

# Higher data rate bits through the channels
# Higher data rate implies bits are transmitted side by side immediately
x = hstack((zeros((1,1)),ones((1,1)),-1*ones((1,1)),ones((1,1)),zeros((1,1)))).flatten()
y = convolve(h,x); # output passed through the band-limited brickwall channel
ax[2,0].stem(x,use_line_collection=True);ax[2,0].set_xlabel('sample #')
ax[2,1].plot(W/pi,abs(H));ax[2,1].set_xlabel('Normalized Freq\n (x $\\pi$ rad/sample)')
ax[2,2].plot(y);ax[2,2].set_xlabel('sample #')
```

Out[24]: Text(0.5, 0, 'sample #')

