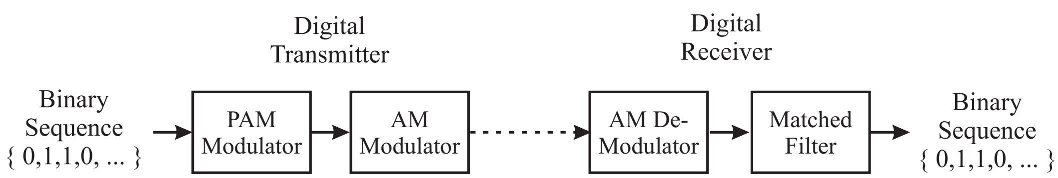
Homework: Pulse Shaping for Digital Coms

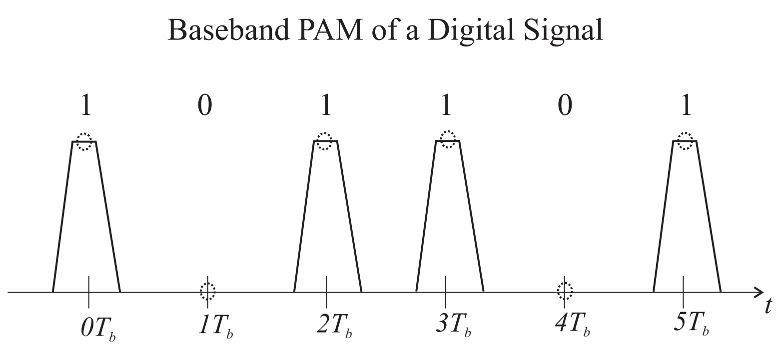
1. **Pulse Shaping:** *Terrestrial wireless example taken from an old undergraduate project*

   Your company has been contracted to design a wireless modem for use by laptop computers operating in office buildings or college classrooms. These wireless modems will be placed in PCMCIA cards that plug into a computer and allow it to communicate with fixed transmitter-receiver access points placed at strategic areas throughout a building. The end result is a *wireless local area network* (WLAN) that allows mobile laptop terminals to access the internet and other information services anywhere within the building. You must provide a scheme for modulating and demodulating a high-data rate signal with one goal in mind: get the most bits-per-second through to the other side.

   Below is a block diagram of the system that you are to design. Refer to this figure for the following technical specifications.



   The transmitter uses pulse-amplitude modulation (PAM) to represent a *binary signal* of 0's and 1's. This procedure is illustrated below using trapezoidal pulses. You have the freedom to chose the bit rate, $R_b = \frac{1}{T_b}$, and the shape of the pulse, $p(t)$, used in the baseband modulator. There is only one restriction on your pulse – digital PAM chip sets are incapable of generating pulses with more than $4T_b$ of time support.



   Although the binary signal you will modulate is *random*, the resulting signal spectrum will resemble the pulse spectrum $P(f)$. In fact, the *power spectral density* of this baseband signal is given mathematically as

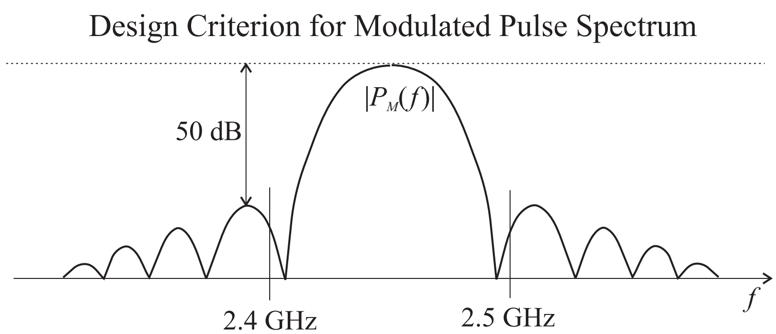
$$G_Y(f) = |P(f)|^2 + \frac{1}{4}|P(0)|^2\delta(f)$$

The baseband signal at the transmitter is then *amplitude modulated* to a carrier frequency. The resulting modulated signal must fit within the *Industrial-Scientific-Medical* (ISM) band, an unlicensed wireless frequency band in the range of 2.4-2.5 GHz.

The Federal Communications Commission (FCC) regulates wireless transmissions and requires that transmitters confine most of their radiated energy within the specified band. The specification for your device requires that the power spectrum outside your frequency band be 50 dB *less* than your peak value in band. For a modulated time-domain pulse, $p_M(t)$, this criterion may be written mathematically as

$$20\log_{10}\frac{|P_M(f)|}{|P_M(f_{\text{peak}})|} < 50 \text{ dB} \quad \text{for all out-of-band } f$$

This is illustrated below:



Find the best possible pulse shape for this system and attach plots of the pulse spectrum that demonstrate the validity of your pulse design. See below for help in taking Fourier Transforms in Matlab.

Notes on Using the FFT Command in Matlab

You will need to calculate and plot the spectrum of pulses in Matlab for this project. Thus, you will need to know how to use the `fft` command. Here is a generic piece of Matlab code you can use:

```
t1 = -10;       % time of first sample
t2 =  10;       % time of last sample
N  = 512;       % number of time-domain samples

t = linspace(t1,t2,N);         % create a time-domain axis
f = (-N/2:N/2-1)/(t2-t1);      % create a frequency-domain axis

x=ustep(.5-abs(t));  % create a time-domain sinc function
X=fft(x);            % take the Fast Fourier Transform

subplot(2,1,1); % make a figure with two plots
plot(t,x);      % plot the time domain signal
xlabel('time (s)'); title('Example of an FFT');

XX = abs(fftshift(X))*(t2-t1)/N;    % shifts/scales FFT output for plotting
subplot(2,1,2);                     % switch to second plot
plot(f,XX,'b-');                    % use semilogy for logarithmic plot
```

```
xlabel('Frequency (Hz)');
```

A few helpful notes:

- The number of samples N in the FFT should be a power of 2. ($N = 2^n$)

- When taking the FFT of a time-domain function of finite-support, be sure to include a large portion of 0-valued samples on either side of the function. This is called *zero-padding* and increases the frequency domain resolution.

- Note that the output of the FFT must be *shifted* using the `fftshift` command. The `abs` function is used to plot just the *magnitude* of the FFT. The output must also be multiplied by $(t2 - t1)/N$ to get the units right. These operations are only used for *plotting*.

- It is best to choose $t_1 = -t_2$ to keep a symmetric time axis.

- Most properties that work on Fourier transforms still work on FFT's (i.e. time shift, convolution, etc.)