

MIT OpenCourseWare  
<http://ocw.mit.edu>

## 16.36 Communication Systems Engineering

Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
Department of Aeronautics and Astronautics

16.36: Comm. Sys. Engineering  
Problem Set No. 5

**Problem 1:** Text problem 7.5

**Problem 2:** Text problem 7.14, parts 1, 2, and 3

**Problem 3:** Text problem 7.16 (for simplicity let  $g(t) = \text{Sinc}(t)$ )

**Problem 4:** Text problem 7.18\*

\*Determine and sketch the impulse response of the matched filter for all 3 cases. Determine the impulse response of the output of the matched filter for case c only (you may do the others for fun, if you like...)

### Problem 5: Matlab Exercise

In this exercise, you will create a PAM matched-filter demodulator function to work with the PAM bandpass modulator that you created in the previous problem set.

- A) Your function should look like the following:
- a. Inputs to your function should be the bandpass modulated waveform generated by your modulator function, the carrier frequency, the sampling frequency, the distance between amplitude levels, and the number of modulation levels  $M$ . You may find the following modifications to your modulator useful:
    - i. Output the time vector for the modulated waveform corresponding to the sampling rate. This can be used as an input to your demodulator, allowing you to not recalculate it.
    - ii. Output the pulse shaping function  $g(t)$ . This can then be used as your matched filter,  $g(T-t)$ .
  - b. The notes and text give a schematic for demodulation.
    - i. First multiply your input waveform by  $2\cos(2\pi f_c t)$ . Remember that Matlab only works with discrete waveforms and you will need to generate a cosine that corresponds to the length in time of your modulated waveform, and sampled at the correct rate. This is why it is useful to have the time vector generated by the modulator as an input.
    - ii. Now pass the signal through the matched filter. This is commonly implemented as a convolution between the input waveform and the matched filter. In Matlab, use the function `conv` to perform the convolution. The output of your convolution must be scaled by  $R_s * f_c / f_s$ , where  $R_s$  is the symbol rate, which we previously set to 1. This scaling is necessary because Matlab is performing a convolution on a vector representing discrete points in time.
  - c. The output of the matched filter will be sampled at  $t=kT$  to decide on the  $k$ th transmitted symbol. Time  $T$  corresponds to  $f_s / (f_c * R_s)$ .
  - d. Design a simple detector based on the distance between amplitude levels. Make the decision region boundaries equidistant between amplitude levels.
  - e. Decode the decision using the same encoding scheme you used for the modulator. Create a bit string output.
- B) We will use the same inputs that we used for the modulator
- a. Carrier frequency of 1 Hz.
  - b. Sampling frequency of 100 Hz.
  - c. Distance between the different signaling levels is 2.
  - d. Two different  $M$ 's with respective input bit strings
    - i.  $M = 4$ , input: '00011011'
    - ii.  $M = 8$ , input: '000001010011100101110111'
- C) Please produce as output the plot of the output of your matched filter (convolution) for  $M=4$  and  $M=8$ . Also print your decision points and output bit string. Make sure to comment all of your code.