Wireless PHY: Modulation and Demodulation

Y. Richard Yang

09/11/2012
Outline

- Admin and recap
- Amplitude demodulation
- Digital modulation
Admin

- Assignment 1 posted
Recap: Modulation

- Objective
  - Frequency assignment

- Basic concepts
  - the information source (also called baseband)
  - carrier
  - modulated signal
Recap: Amplitude Modulation (AM)

- **Block diagram**
  
  \[ x(t) \xrightarrow{m} x \xrightarrow{+} x_{AM}(t) = A_c [1 + mx(t)] \cos \omega_c t \]

- **Time domain**

- **Frequency domain**

  \[ X(f) \xrightarrow{} X_{AM}(f) \]
Recap: Demod of AM

- Design option 1: multiply modulated signal by $e^{-jfc_t}$, and then LPF

- Design option 2: quadrature sampling
Example: Scanner

- Setting: a scanner scans 128KHz blocks of AM radio and saves each block to a file.
- For the example file
  - During scan, $f_c = 710K$
  - LPF = 128K (one each side)
Exercise: Scanner

- **Requirements**
  - Scan the block in a saved file to find radio stations and tune to each station (each AM station has 10 KHz)
  - Audio device requires 48K sample rate for playback
Remaining Hole: How to Design LPF

- Frequency domain view

[Diagram showing frequency response with -B and B on the frequency axis]
Design Option 1

This is essentially how image compression works.

Problem(s) of Design Option 1?
Design Option 2: Impulse Response Filters

- GNU software radio implements filtering using:
  - Finite Impulse Response (FIR) filters
  - Infinite Impulse Response (IIR) Filters
  - FIR filters are more commonly used

- FIR/IIR is essentially online, streaming algorithms

- They are used in networks/communications/vision/robotics...
FIR Filter

- An $N$-th order FIR filter $h$ is defined by an array of $N+1$ numbers:

$$h = [h_0, h_2, \ldots, h_N]$$

- They are often stored backward (flipped):

  $\begin{array}{c}
  h_N \\
  \vdots \\
  h_2 \quad h_1 \quad h_0
  \end{array}$

- Assume input data stream is $x_0, x_1, \ldots,$
FIR Filter

compute $y[n]$: $y_n = x_n h_0 + x_{n-1} h_1 + \ldots + x_{n-N} h_N$

$= \sum_{i=0}^{N} x_{n-i} h_i$
FIR Filter

\[ y[n+1] = h_0 x[n] + h_1 x[n-1] + h_2 x[n-2] + h_3 x[n-3] \]
FIR Filter

\[ y_n = x_n h_0 + x_{n-1} h_1 + \ldots + x_{n-N} h_N \]

is also called convolution between \( x \) (as a vector) and \( h \) (as a vector), denoted as

\[ y_n = x_n * h_n \]
Key Question Using h to Implement LPF

❖ Q:
  ❖ How to determine h?

❖ Approach:
  ❖ Understand the effects of $y=g*h$ in the frequency domain
$g \ast h$ in the Continuous Time Domain

- Remember that we consider $x$ as samples of time domain function $g(t)$ on $[0, 1]$ and (repeat in other intervals)
- We also consider $h$ as samples of time domain function $h(t)$ on $[0, 1]$ (and repeat in other intervals)

\[
y(t) = \int_{0}^{1} h(\tau)g(t - \tau) \, d\tau
\]

for $(i = 0; i < N; i++)$

\[
y[t] += h[i] \ast g[t-i];
\]
Visualizing $g \ast h$

$g(t)$

$h(t)$

0 $\rightarrow$ T $\rightarrow$ 0

$\rightarrow$ time
Visualizing $g \ast h$

$g(t)$

$h(0)$

$t$

$0 \rightarrow T$

$T \rightarrow 0$

$\ast$

$g(t)$
Fourier Series of $y=g*h$

$$y(t) = \int_{0}^{1} h(\tau)g(t-\tau)\,d\tau$$

$$Y[k] = \int_{0}^{1} y(t)e^{j2\pi kt}\,dt$$

$$= \int_{0}^{1} \left[ \int_{0}^{1} h(\tau)g(t-\tau)\,d\tau \right] e^{j2\pi kt}\,dt$$

$$= \int_{0}^{1} \left[ \int_{0}^{1} h(\tau)g(t-\tau)e^{j2\pi kt}\,d\tau \right] dt$$
Fubini’s Theorem

\[ \int_A \left( \int_B f(x, y) \, dy \right) \, dx = \int_B \left( \int_A f(x, y) \, dx \right) \, dy = \int_{A \times B} f(x, y) \, d(x, y), \]

- In English, you can integrate
  - first along \( y \) and then along \( x \)
  - first along \( x \) and then along \( y \)
  - at \( (x, y) \) grid

They give the same result

See \url{http://en.wikipedia.org/wiki/Fubini's_theorem}
Fourier Series of \( y=g\ast h \)

\[
y(t) = \int_0^1 h(\tau)g(t-\tau)\,d\tau
\]

\[
Y[k] = \int_0^1 \left[ \int_0^1 h(\tau)g(t-\tau)e^{j2\pi kt}\,d\tau \right]dt
\]

\[
= \int_0^1 \left[ \int_0^1 h(\tau)g(t-\tau)e^{j2\pi kt} \,dt \right]d\tau
\]

\[
= \int_0^1 h(\tau) \left[ \int_0^1 g(t-\tau)e^{j2\pi kt} \,dt \right]d\tau
\]

\[
= \int_0^1 h(\tau)e^{j2\pi k\tau} \left[ \int_0^1 g(t-\tau)e^{j2\pi k(t-\tau)} \,dt \right]d\tau
\]

\[
= \int_0^1 h(\tau)e^{j2\pi k\tau} G[k] \,d\tau
\]

\[
= G[k]H[k]
\]
Summary of Progress So Far

\[ y = g * h \Rightarrow Y[k] = G[k] H[k] \]

In the case of Fourier Transform,
\[ y = g * h \Rightarrow Y[f] = G[f] H[f] \]

is called the Convolution Theorem, an important theorem.
Applying Convolution Theorem to Design LPF

- Choose $h()$ so that
  - $H()$ is close to a rectangle shape
  - $h()$ has a low order (why?)
The h() is often related with the sinc(t) = sin(t)/t function.

\[ \int_{-\infty}^{\infty} \frac{\sin(\pi t)}{\pi t} e^{-j2\pi ft} = rect(f) \]
FIR Design in Practice

- **Compute h**
  - MATLAB or other design software
  - GNU Software radio: optfir (optimal filter design)
  - GNU Software radio: firdes (using a method called windowing method)

- **Implement filter with given h**
  - freq_xlating_fir_filter_ccf or
  - fir_filter_ccf
LPF Design Example

- Design a LPF to pass signal at 1 KHz and block at 2 KHz
#create the channel filter
# coefficients
chan_taps = optfir.low_pass(
    1.0, #Filter gain
    48000, #Sample Rate
    1500, #one sided mod BW (passband edge)
    1800, #one sided channel BW (stopband edge)
    0.1, #Passband ripple
    60) #Stopband Attenuation in dB
print "Channel filter taps: ", len(chan_taps)

#creates the channel filter with the coef found
chan = gr.freq_xlating_fir_filter_ccf(
    1, # Decimation rate
    chan_taps, #coefficients
    0, #Offset frequency - could be used to shift
    48e3) #incoming sample rate
Outline

- Recap
- Amplitude demodulation
  - frequency shifting
  - low pass filter
- Digital modulation
Modulation

- Modulation of digital signals also known as **Shift Keying**
- Amplitude Shift Keying (ASK):
  - vary carrier amp. according to data
- Frequency Shift Keying (FSK)
  - vary carrier freq. according to bit value
- Phase Shift Keying (PSK)
  - vary carrier freq. according to data
Phase Shift Keying: BPSK

- **BPSK (Binary Phase Shift Keying):**
  - bit value 1: cosine wave \( \cos(2\pi f_c t) \)
  - bit value 0: inverted cosine wave \( \cos(2\pi f_c t + \pi) \)
  - very simple PSK

- **Properties**
  - robust, used e.g. in satellite systems

![Diagram of BPSK](image)

- One bit time \( T \)
- Q vs. I axis
Phase Shift Keying: QPSK

- QPSK (Quadrature Phase Shift Keying):
  - 2 bits coded at a time
  - we call the two bits as one symbol
  - symbol determines shift of cosine wave
  - often also transmission of relative, not absolute phase shift: DQPSK - Differential QPSK
Quadrature Amplitude Modulation

- Quadrature Amplitude Modulation (QAM): combines amplitude and phase modulation
- It is possible to code $n$ bits using one symbol
  - $2^n$ discrete levels

Example: 16-QAM (4 bits = 1 symbol)
- Symbols 0011 and 0001 have the same phase $\phi$, but different amplitude $a$. 0000 and 1000 have same amplitude but different phase.
Generic Representation of Digital Keying (Modulation)

- Sender sends symbols one-by-one
- $M$ signaling functions $g_1(t)$, $g_2(t)$, ..., $g_M(t)$, each has a duration of symbol time $T$
- Each value of a symbol has a signaling function
Exercise: $g_i()$ for BPSK

1:
- $g_1(t) = \cos(2\pi f_c t) \ t \text{ in } [0, T]$

0:
- $g_0(t) = -\cos(2\pi f_c t) \ t \text{ in } [0, T]$

Are the two signaling functions independent?

- Hint: think of the samples forming a vector, if it helps, in linear algebra
- Ans: No. $g_1(t) = -g_0(t)$

\[
\begin{array}{c|c}
\text{t} & g_0(t) \\
\hline
-1 & g_1(t) \\
1 & \cos(2\pi f_c t) [0, T]
\end{array}
\]
Exercise: Signaling Functions $g_i()$ for QPSK

- **11:**
  - $\cos(2\pi f_c t + \pi/4) \ t \in [0, T]$

- **10:**
  - $\cos(2\pi f_c t + 3\pi/4) \ t \in [0, T]$

- **00:**
  - $\cos(2\pi f_c t - 3\pi/4) \ t \in [0, T]$

- **01:**
  - $\cos(2\pi f_c t - \pi/4) \ t \in [0, T]$

Are the four signaling functions independent?
- Ans: No. They are all linear combinations of $\sin(2\pi f_c t)$ and $\cos(2\pi f_c t)$. 
QPSK Signaling Functions as Sum of $\cos(2\pi f_c t)$, $\sin(2\pi f_c t)$

- **11**: $\cos(\pi/4 + 2\pi f_c t)$  $t$ in $[0, T]$
  \[ \rightarrow \cos(\pi/4) \cos(2\pi f_c t) + \sin(\pi/4) \sin(2\pi f_c t) \]

- **10**: $\cos(3\pi/4 + 2\pi f_c t)$  $t$ in $[0, T]$
  \[ \rightarrow \cos(3\pi/4) \cos(2\pi f_c t) + \sin(3\pi/4) \sin(2\pi f_c t) \]

- **00**: $\cos(-3\pi/4 + 2\pi f_c t)$  $t$ in $[0, T]$
  \[ \rightarrow \cos(3\pi/4) \cos(2\pi f_c t) + \sin(3\pi/4) \sin(2\pi f_c t) \]

- **01**: $\cos(-\pi/4 + 2\pi f_c t)$  $t$ in $[0, T]$
  \[ \rightarrow \cos(\pi/4) \cos(2\pi f_c t) + \sin(\pi/4) \sin(2\pi f_c t) \]

We call $\sin(2\pi f_c t)$ and $\cos(2\pi f_c t)$ the bases.
Outline

- Recap
- Amplitude demodulation
  - frequency shifting
  - low pass filter
- Digital modulation
  - modulation
  - demodulation
Key Question: How does the Receiver Detect Which $g_i()$ is Sent?

- Assume synchronized (i.e., the receiver knows the symbol boundary).
Starting Point

- Considered a simple setting: sender uses a single signaling function $g()$, and can have two actions
  - send $g()$ or
  - nothing (send $0$)

- How does receiver use the received sequence $x(t)$ in $[0, T]$ to detect if sends $g()$ or nothing?
Design Option 1

- Sample at a few time points (features) to check
- Issue
  - Not use all data points, and less robust to noise
Design Option 2

- Streaming algorithm, using all data points in [0, T]
  - As each sample $x_i$ comes in, multiply it by a factor $h_{T-i-1}$ and accumulate to a sum $y$

![Diagram showing streaming algorithm]

- At time $T$, makes a decision based on the accumulated sum at time $T$: $y[T]$
Example Streaming (Convolution/Correlation):

- Assume incoming $x$ is a rectangular pulse (in baseband) and $h$ is also a rectangular pulse.

- A gif animation:
  - redline $g()$: the sliding filter $h(t)$
  - blue line $f()$: the input $x()$

Determining the Best $h$

$$y = (g + w) * h = g * h + w * h = g_o + n$$

where $w$ is noise, $g_o(t) = g * h$

$$n = w * h$$

Design objective: maximize peak pulse signal-to-noise ratio

$$\eta = \frac{|g_o(T)|^2}{E[n^2(T)]} = \frac{\text{instantaneous signal power}}{\text{average noise power}}$$
Determining the Best $h$

Assume Gaussian noise, one can derive

$$E[n^2(T)] = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 \, df$$

Using Fourier Transform and Convolution Theorem:

$$g_o(T) = \int_{-\infty}^{\infty} G_0(f) e^{j2\pi fT} \, df = \int_{-\infty}^{\infty} G(f) H(f) e^{j2\pi fT} \, df$$

$$\eta = \left| \int_{-\infty}^{\infty} G(f) H(f) e^{j2\pi fT} \, df \right|^2$$

$$\eta = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 \, df$$
Determining the Best $h$

Apply Schwartz inequality

$$\eta = \frac{\left| \int_{-\infty}^{\infty} G(f)H(f)e^{j2\pi fT} \, df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 \, df}$$

$$\int_{-\infty}^{\infty} x(f)y(f) \, df \leq \int_{-\infty}^{\infty} |y(f)|^2 \, df \quad \text{equal iff} \quad x(f) = ky^*(f)$$

By considering

$$x(f) = H(f)$$
$$y(f) = G(f)e^{j2\pi Tf}$$

$$H_{opt}(f) = k[G(f)e^{j2\pi fT}]^*$$
$$= kG^*(f)e^{-j2\pi fT}$$
Determining the Best $h$

$$H_{opt}(f) = kG * (f)e^{-j2\pi f T}$$

$$h_{opt}(t) = \int_{f=-\infty}^{\infty} H_{opt}(f)e^{j2\pi ft} = \int_{f=-\infty}^{\infty} kG * (f)e^{-j2\pi f T} e^{j2\pi ft}$$

$$= \int_{f=-\infty}^{\infty} kG(-f)e^{-j2\pi f T} e^{j2\pi ft}$$

$$= \int_{f=-\infty}^{\infty} kG(-f)e^{-j2\pi f (T-t)}$$

$$h_{opt}(t) = kg(T - t)$$

$$= \int_{f=-\infty}^{\infty} kG(f)e^{j2\pi f (T-t)}$$

$$\eta = \frac{\left| \int_{-\infty}^{\infty} G(f)H(f)e^{j2\pi f T} df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df}$$
Determining Best $h$ to Use

\[ h_{opt}(t) = kg(T - t) \]
Matched Filter Decision

\[ h_{opt}(t) = kg(T - t) \]

is called Matched filter.

Example

\[ h_{opt}(t) = kg(T - t) \]

\[ g_o(t) = g(t) \ast h(t) \]
Backup Slides
Modulation