SFWR ENG 4J03

Author: Kemal Ahmed

Instructor: Dr. Gowri Krishnasamy

Winter 2016

*Math objects made using* [*MathType*](http://www.dessci.com/en/products/mathtype/)*.*

Table of Contents

[Abbreviations 1](#_Toc447595592)

[Angle Modulation 1](#_Toc447595593)

[Frequency Modulation 1](#_Toc447595594)

[Power 1](#_Toc447595595)

[Phase Modulation 1](#_Toc447595596)

[Amplitude Modulation 1](#_Toc447595597)

[Generating SSB-SC 1](#_Toc447595598)

[Frequency Discrimination Method 1](#_Toc447595599)

[Phase Discrimination Method 1](#_Toc447595600)

[Current 1](#_Toc447595601)

[Information Theory 1](#_Toc447595602)

[Shannon-Fano 1](#_Toc447595603)

[Huffman Coding 1](#_Toc447595604)

[Message Sampling 1](#_Toc447595605)

[Pulse Code Modulation 1](#_Toc447595606)

[Regenerative repeater 1](#_Toc447595607)

[Pulse Width Modulation 1](#_Toc447595608)

[Pulse Position Modulation 1](#_Toc447595609)

[Quantization 1](#_Toc447595610)

[Uniform Quantization 1](#_Toc447595611)

[Quantization Error 1](#_Toc447595612)

[Signal : Noise 1](#_Toc447595613)

[Multiplexing 1](#_Toc447595614)

[Data Rate Management 1](#_Toc447595615)

[Error Control Coding 1](#_Toc447595616)

[Linear Block Coding 1](#_Toc447595617)

[Hamming 1](#_Toc447595618)

[Cyclic Code 1](#_Toc447595619)

[Long Division Method 1](#_Toc447595620)

[Convolution Code 1](#_Toc447595621)

[Properties of Maximal Length Sequences 1](#_Toc447595622)

[I. Balance Property 1](#_Toc447595623)

[II Run Property 1](#_Toc447595624)

[III Auto Correlation Property 1](#_Toc447595625)

[Shift Keying 1](#_Toc447595626)

[FHSS 1](#_Toc447595627)

# Abbreviations

(O/P): Output

Tx: transmission

# Angle Modulation

**Signal to Noise Ratio (SNR)**: signal power / Pnoise


## Frequency Modulation

**Modulation**: the process of varying one or more properties of a periodic waveform, called the carrier signal, with a modulating signal that typically contains information to be transmitted

**Angular Frequency** [ω]: 2πf

**Message Angular Frequency** [ωm]: 2πfm

**Carrier Angular Frequency** [ωc]: 2πfc

**Angle Modulation**: frequency or phase modulation

**Frequency Modulation (FM)**:

* better for audio signals, needing kHz (300Hz - 3kHz)
* multiple sidebands: amplitude higher
* non-linear

**Demodulation**:

**Inductance** [L]:

**Capacitance** [C]:

**Message Frequency** [fm]:

**Carrier Frequency** [fc]: 

**Instantaneous Frequency** [fi]: fc + kf m(t)

**Angle** [θ]: unmodulated carrier

θi(t) = 2πfc t + ϕ

**Oscillator**: produces a signal that converts the digital message into analog signal

* requires very precise frequency and phase to match the carrier

**Difference Signal**: oscillator frequency – input signal

**Balanced Modulator**: frequency translations

**Bandwidth (BW)**: = upper sideband – lower sideband = 2fm

**Carson’s Rule**:

**Transmission Bandwidth** [BT]: BT = 2Δf + 2fm = 2Δf (1 + 1/β)

**Peak Frequency Deviation** [Δfc]: Δfc = kf ∙ Am = (βAc)2/R

**Narrow Band Frequency Modulation (NBFM)**:

**Wide Band Frequency Modulation (WBFM)**:

**Frequency Modulation index** [β]: (rad) max frequency deviation / fm

= Δf/fm × 100%

= phase deviation [Δϕ]

**Frequency Sensitivity** [kf]: (Hz/V) sensitivity of modulator

## Power

**Power**:

**Resistance** [R]: 1Ω by default

**Carrier Signal Power** [Pc]: Ac2/(2R)

**Power Spectral Density** [SM(f)]:

**Message Power** [Pm]: PUSB + PLSB

[PUSB]: m2/4

[PLSB]: m2/4

**Total Power** [Ptotal]: 

**Peak Power** [Pp]: 

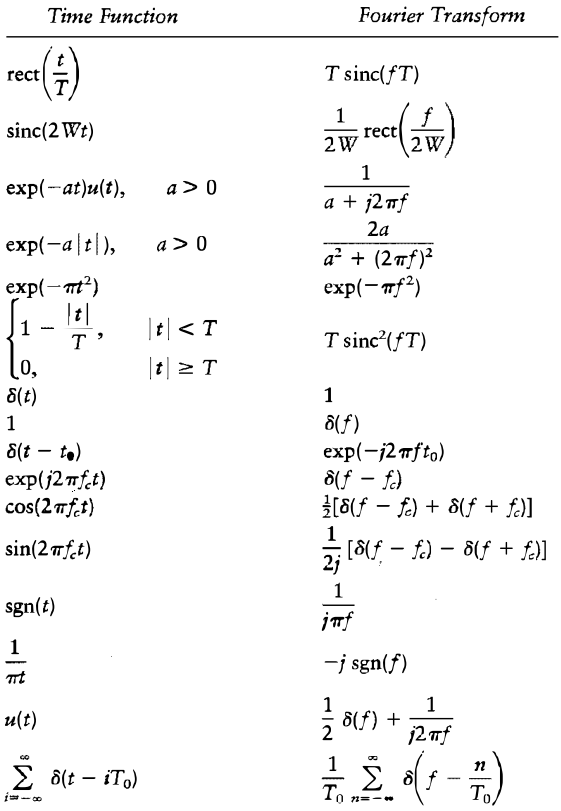
**Harmonics**: when waves build up…

**Audible frequency range**:

**Audio modulating frequency range**:



Fourier





## Phase Modulation

**Phase modulation (PM)**:

**Phase Sensitivity** [kp]:

θi(t) = 2πfct + kp m(t)

S(t) = Ac cos (2πfct + kp m(t))

**Phase Deviation** [Δϕ]:

# Amplitude Modulation

**Amplitude Modulation (AM)**:

* modulated signal contains two side bands and an unmodulated carrier signal
* better for video signals (over the air), needing MHz, perhaps 5.5
* linear

Don’t look at phase shifts for AM, so you can convert between sin ↔ cos

**Maximum Amplitude** [Am]: Am= Acm/2 ?

**Message Signal** [m(t)]: m(t) = Am cos(ωmt)

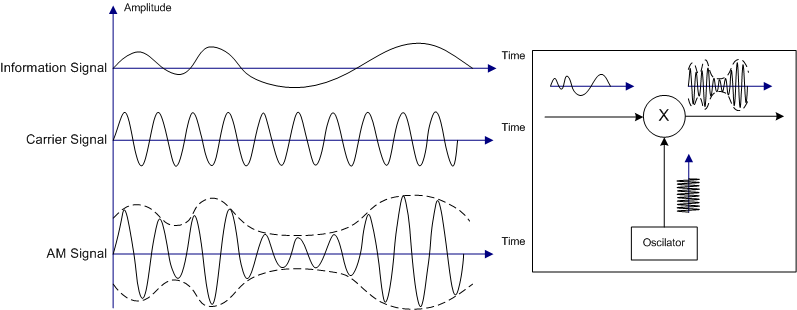
a.k.a. modulating signal, unmodulated signal, data signal, information signal

**Carrier Signal** [c(t)]: original carrier signal

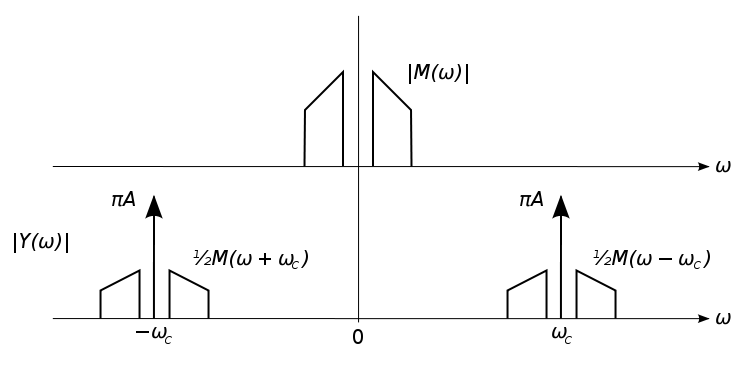
**Modulated Signal** [s(t)]: a.k.a. **Output** (O/P) or **sampled signal**

s(t) = m(t) × c(t)

[AAM]: Ac + m(t)

[](https://upload.wikimedia.org/wikipedia/commons/8/8d/Illustration_of_Amplitude_Modulation.png)

**DSB-AM**: a.k.a. conventional AM

[](https://upload.wikimedia.org/wikipedia/commons/a/ae/AM_spectrum.svg)



**Amplitude Modulation Index** [m]: 

**AM BW**: max – min = (fc + fm) – (fc – fm) = 2∙fm

**Band Pass Filter (BPF)**:

**Vestigeal Sideband Transmission (VSB)**: one side band along with just a trace of the other side band (a.k.a. **vestige**). This trace is useful for ensuring that important information is not cut off when reading

**Minimum SB (MSB)**:

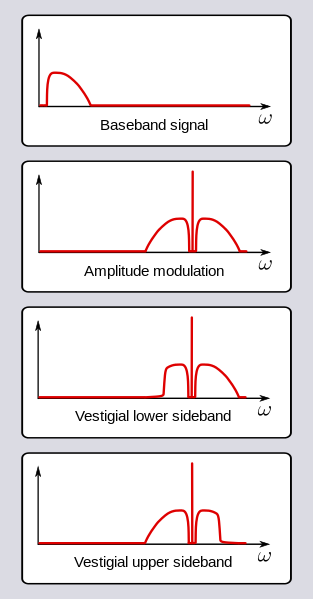
Filters, such as BPF, tend to remove a little bit of the message. You can avoid this by extending the length of the transmission and including a trace of the opposite SB.

**Message Bandwidth** [W]: maximum message frequency, i.e. message

**SideBand (SB)**:

**Suppressed Carrier (SC)**: don't transmit carrier signal with the message signal, so less power, but complicated filter because you transmit the signals on the sidebands  
  
**Double SB-SC (DSB-SC)**: USB & LSB

**Single SB-SC (SSB-SC)**: transmit only one sideband frequency, usually a DSB through a BPF  
  
**Lower SB (LSB)**: (fc – fm)  
  
**Upper SB (USB)**: (fc + fm)

[](https://upload.wikimedia.org/wikipedia/commons/thumb/0/0a/VSB_bandform.svg/313px-VSB_bandform.svg.png)

**Envelope Detection**: a demodulation method that converts AM to m(t), using c(t)

* Cannot be used with SC because envelope is no longer representing m(t)

## Generating SSB-SC

1. Frequency discrimination method
2. Phase discrimination method

### Frequency Discrimination Method



### Phase Discrimination Method

π/4 phase shift → product modulator → S1(x) → Sum → SSB-SC S(k)

**Phase** [ϕ]:

# Current

[IC]:

[IT]:

# Information Theory

**binits** [v]: binary bits

**Information Bits** [Ik]: log2(1/Pk)

**Prefix Property**: no sequences can be obtained from each other by adding more binary digits to the shorter sequence

**Source Information Rate** [R]: R = r × H

## Shannon-Fano

**Shannon-Fano code**: an encoding method that finds efficiency of code, listed with probabilities in decreasing order

**Ensemble**: source of the messages

1. Split into 2 groups as similar in size as possible without first rearranging. Sometimes it may be more efficient to put a smaller group on top because it is more probable and will require less bits.
2. Allocate 1s to one group and 0s to the other. Either put 0s on all the top groups or 1s in all the top groups
3. Split your groups into smaller groups
4. Continue partitioning until you only have groups of size one.

Since the bottom division is always larger, looking at the first bits will most likely be the same. Thus, flipping the order of the bits causes more variance, i.e. unique, increasing the probability that the identifying bits will come sooner while checking messages from left to right.

## Huffman Coding

**Huffman Coding**: an encoding method

[N~]: average number bits per message



**Code efficiency** [η]:



Reverse when transmitted.

**Entropy** [H]: average information content per source symbol

H = Itotal/L

= sum(Pklog2(1/Pk), k=1..m)



Properties:

1. Pk = 0 / 1 🡪 H = 0
2. When m symbols equally likely, i.e. symbols have equal probability:
   1. H = log2m
   2. Hmax = log2m

When entropy < capacity of a channel, error-free communication is possible

# Message Sampling

**Sampled signal**:

**S**ignal modulation

* Analog Data:
  + **Pulse Amplitude Modulation (PAM)**: message encoded in amplitude
    - **Single Polarity**: DC bias ensures all positive
    - **Double Polarity**: pulses both negative / positive
      * **Natural Sampling**: s(t) retains shape of x(t)
      * **Flat Top Sampling**: left Riemann sum with bars
      * **Ideal Sampling**: left Riemann sum, but with points
  + **Pulse Width Modulation (PWM)**: a.k.a. **Pulse Distance Modulation (PDM)**
    - messages encoded in pulse width
  + **Pulse Position Modulation (PPM)**: message encoded in distance between pulses
    - higher → more distance
* Digital Data:
  + **Pulse Code Modulation (PCM)**: message encoded in distance between pulses
  + **Delta Modulation (DM)**: messages encoded pulse width
    - can be shifted with amplitude
    - Staircase approximation
  + **Adaptive DM (ADM)**: non-fixed step-size
  + **Line Coding**:
    - Types:
      * **L**: line follows 1 and 0 directly
      * **M**: line changes every 1
      * **S**: line changes every 0
    - **Non-Return to Zero (NRZ)**: line follows 1 and 0 directly
      * Higher energy
    - **Return to Zero (RZ)**: pulse every 1 that returns to 0 before next bit
    - **Biphase**:
      * clock is double data rate
      * 1 is high for first clock
      * 0 is high for second clock
    - **Differential Manchester**:
      * clock is double data rate
      * 1 is transition before
      * 0 is transition after
      * transition means continuing the previous value
    - **Bipolar**:
* Message signal
  + contains information to be transmitted
  + Analog signals, i.e. continuous with respect to time and amplitude
* Carrier Signal
  + Analog signal (analog modulation)
  + Pulse train (pulse modulation)

## Pulse Code Modulation

Message Source → Band limiting filter → Sampler → quantization → encoder → PCM signal → Regenerative repeater → Regenerative repeater → … → Regenerative repeater → PCM signal output

PCM → regenerative repeater → decoder → LPF →

Analogous:

* PAM → AM
* PWM → FM
* PPM → PM

### Regenerative repeater

**Regenerative repeater**: amplitudes to make sure that the signal stays strong after travelling over large distances

brb tho

### Pulse Width Modulation

**Sampling Process**:

* Think Riemann Sum or amplifying instantaneous samples
* Change the amplitude of the carrier, such that the frequency, amplitude is constant, but width of the pulse is varied in accordance to the message signal

### Pulse Position Modulation

* Amplitude, width constant
* Frequency varied

# Quantization

**Quantization**: truncates, rounds sample amplitudes, reducing precision, to lowering bits necessary for encoding

**Quantization level** [L]: number of possible values within range, after rounding

**Quantization error** [q]:

## Uniform Quantization

**Quantizer**: rounds the sampled amplitudes to whatever amount of decimals that you want

Round to nearest 2 decimal bits (0.00, 0.01, 0.10, 0.11):

0, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.5, 0.5, 0.5, 0.75, 0.75

**Staircase waveform**: vertical cliffs and plateaus climbing upwards with x

* First step is a half-step
* Used in quantization

**Number of zones** [L]:

**Step Size** [δ]: (xmax – xmin)/L

**Quantization Error** [ε]: error from rounding

**Mid-tread**: origin is in middle of plateau, where error is sawtooth wave

– δ/2 < x(nTs) ≤ δ/2

xq(nTs) = 0

δ/2 ≤ x(nTs) ≤ 3 δ/2

xq (nTs) = δ

**Mid-rise**: origin is in middle of cliff, where error is negative sawtooth wave

0 < x(nTs) < δ

xq (nTs) = δ/2

## Quantization Error

**Quantizer Output** [xq(nts)]:

**Sampler Output** [x(nts)]:

**Noise Voltage** [Vnoise]:

**Noise Power** [Pnoise]: v2noise/R mean square value of noise voltage

E[ε2] = 

1. Error: 
2. Step size: 
3. δ/2 ≥ εmax ≥ δ/2. Note: uniform distribution
4. δ = 2xmax / q
5. **Noise Power** = v2noise/R ⇒ Mean square value =

↑SNR = signal power↑ / Pnoise↓

**Bit transmission rate** [r]: a.k.a. **signaling rate**

**Regenerative Repeater**: taking digital signals and regenerating the values to maintain a level of quality

## Signal : Noise

**Normalized signal power** [S]:

**Normalized noise power** [N]:

S/N ⇒ 

δ = 2xmax / q = 2xmax / 2V





**Nyquist sampling criterion**: you’ll get aliasing if fm > **Nyquist rate**

**Nyquist rate**: 2∙fs

**Nyquist interval**:

**Nyquist frequency**: ½ fs of discrete processing system

If you have multiple messages, choose the highest one to be the rate to be doubled

# Multiplexing

When you have too many inputs going over 1 wire, you need to figure out a way to identify which input goes to which output

Types:

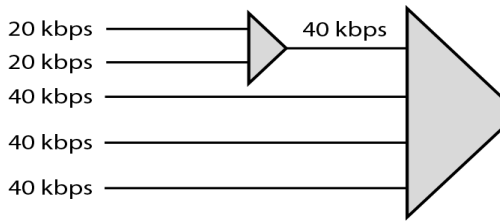
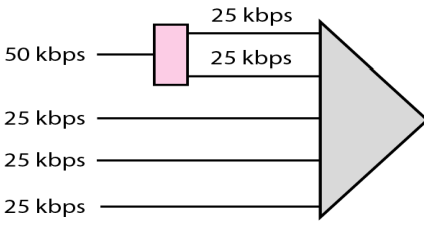
* **Frequency-Division Multiplexing (FDM)**: analog
* **Wavelength-Division Multiplexing (WDM)**: analog
* **Time-Division Multiplexing (TDM)**: digital
  + **Synchronous TDM**:
  + **Statistical TDM**:

**Interleaving**: the process of cycling through multiple input channels onto one using frames/packets

**Synchronization**: the process of isolating individual output channels from an *interleaved* output channel

## Data Rate Management

Strategies:

* **Multilevel**:
  + when the data rate of the input links are multiples of each other
  + 
* **Multishot**: higher bit rate channels are allocated more slots per frame, and the output frame rate is a multiple of each input link
  + ∈ GCD between the data rates
  + 
* **Pulse Stuffing**: slowest speed link will be brought up to the speed of the other links by bit insertion
  + no GCD
* Know when input channels have no data because then it is just wasting bandwidth

# Error Control Coding

**Error Control Coding**: coding in a way that checks correctness of Tx

Mechanisms:

* [**Linear Block Coding**](#_Linear_Block_Coding)
* **Repetition Coding**:
* **Convolution Coding**:

## Linear Block Coding

**Message Bits** [k]:

**Codeword Length** [n]:

**Linear Block Code** [(n, k)]: an error-correcting code

a.k.a. **systematic code**

**Control bits** [m]: a.k.a. **parity bits** or **check bits**

(n – k)



pij = { 1, bi depends on mj

0, else }

**Codeword** [ci]: 

**Systematic form**: {*n – k* check bits, *k* information bits}

# Hamming

**Hamming weight** [w(c)]: number of nonzero elements in a code vector

t ≤ ½ (dmin – 1)

**Hamming distance** [d(c1, c2)]: the number of bits in which codewords, c1 and c2, differ

**Minimum Distance** [dmin]: minimum hamming distance between two codewords

Possible error detected: dmin – 1

Possible errors corrected: (dmin – 1)/2

**Hamming(7,4)**: a type of Hamming with 3 check parity bits for every 4 data bits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | P1 | P2 | D1 | P3 | D2 | D3 | D4 |
| P1 |  |  |  |  |  |  |  |
| P2 |  |  |  |  |  |  |  |
| P3 |  |  |  |  |  |  |  |

**Hamming Matrix**:



# Cyclic Code

**Cyclic Code**: a block code, where the circular shifts of each codeword gives another word that belongs to the code

**Codeword Polynomial**: *linear code* whose valid code words are divisible by a generator polynomial

a.k.a. **polynomial code**

**Generator Polynomial**: 

**Syndrome calculator**:

## Long Division Method

**Long division method**: xor your X’s instead of subtracting at each step

**Cyclic Redundancy Check (CRC)**:

**BCH**: don’t worry about the acronym-it’s the names of the authors of the paper

**Reed Solomon (RS)**:

# Convolution Code

**Convolution Code**:

**Viterbi decoding algorithm**:

* two paths: top and bottom
  + calculated separately
* Value:
  + If there’s a path: 1
  + No path: 0
* Each path from left to right is a different number
* g(D) = Σexists × 2path
* c(1)(D) = g(1)(D) m(D)
  + if two of same power, cancel to 0
* Answer: c1 mod c2

**Code Trellis**:

**Spreading Sequence**: spreading information bits over multiple bits, a.k.a. **chips**

**Chip**: when information bits are spread over multiple

**Jamming**: intentional or unintentional

**Feedback Shift Register**:

**Pseudo Noise (PN) Sequence Code**: periodic binary sequence with a noise-like waveform, generated by means of a linear feedback shift register

* Period of a PN Sequence with m flip-flops cannot exceed 2m – 1

**Maximal Length Sequence**: PN sequence if p = 2m – 1

**Taps**: registers whose values are referenced or *tapped*

**Feedback Taps**: taps whose values are XOR’d together to make the next bit

**Shift Register Length** [m]:

## Properties of Maximal Length Sequences

### I. Balance Property

In each period of a maximal length sequence, the number of 1’s is always one more than number of 0’s

### II Run Property

**Run**: subsequence of identical symbols

Number of runs = 

N = 2m – 1

number of runs = p/2

1. One run of length m number of 1’s
2. One run of length m–1 number of 0’s
3. One half of the runs are of length 1
4. One fourth are of length 1

### III Auto Correlation Property

The (ACP) of a max length sequence is periodic of binary values

Period, PN Sequence Length: N = 2m – 1

**Bit Duration** [Tb]:

**Chip Duration** [Tc]: 1/chip rate

**Symbol Duration** [Ts]:

Tb = NTc





**Bandwidth efficiency**:



# Shift Keying

**Shift Keying (SK)**:

**Amplitude SK (ASK)**:

**Frequency SK (FSK)**: modulates by changing the frequency of carrier

**Binary FSK (BFSK)**: hopping between 2 frequencies



Energy per bit





**M-ary FSK (MFSK)**: hopping between M frequencies

[k]: PN segment length per hop

[K]: bits/ symbol

[M]: tones, M = 2K

[Rb]: bit rate

[Rc]: chip rate

[Rh]: hop rate

[Rs]: symbol rate

Total frequency hops: 2k

**Phase SK (PSK)**: conveys data, i.e. modulates, by changing the phase of the carrier

**Binary PSK (BPSK)**: a.k.a. **2PSK** or 2 phase shift keying

**Quadrature PSK (QPSK)**: 4 phase shifting

**M-ary PSK (MPSK)**: M = 2­­b symbols





**Differential PSK (DPSK)**: message flips the phase

Error:



# FHSS

**Frequency Hopping Spread Spectrum (FHSS)**: method of transmitting radio signals by rapidly switching a carrier among many frequency channels, using PN codes

* avoids jamming
* commonly uses MFSK

**Slow FHSS**: multiple symbols per hop

* has Tc ≥ Ts
* Rc = Rs = Rb/K ≥ Rh

**Fast FHSS**: multiple hops per symbol; has Tc < Ts

* Each hop represented by chip
* Rc = Rs/K = Rh