EE123

Digital Signal Processing

Lecture 30

Project

- Default project:
  - SSTV tranciever does not necessarily requires an SDR -- Should work with just the radio. SDR as well is a plus.
  - SSTV project is individual! Unless you are proposing significant extensions.
  - SSTV is analog communications. Analog is not digital!!!

Lab 3 Part III - afsk, AX.25 and APRS

- The lab implements a packet based tranciever
- You will be able to send/receive packet to other classmates
- You will be able to send/receive APRS packets that users and stations with APRS equipped radios can decode.

AFSK1200 / Bell 202 modem

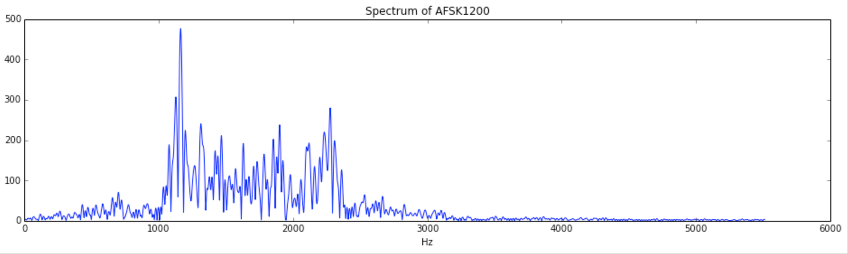
- Audio FSK
  - Encodes digital data at 1200b/s
  - Use audio frequencies 1200/2200Hz
  - Within the bandwidth of the audio input BP filter of your radios
  - Still(!) popular for ham packet networks

$$s(t) = cos\left(2\pi f_c t + 2\pi\Delta f \int_{\infty}^{t} m(\tau)d\tau\right)$$

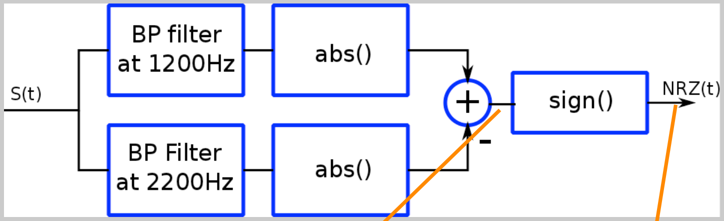
  - fc = 1700, △f = 500, m(t) = ± 1
  - Phase is not the same for each bit -- must use non coherent detection.

AFSK1200

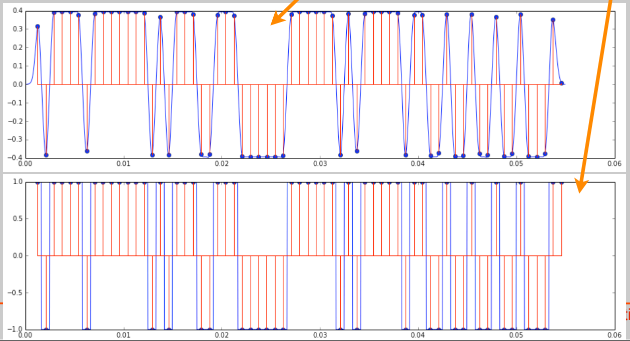
- Write a function to generate AFSK1200
  - Take care: sampling rate (44.1KHz) does not divide with bit-rate
  - Look at Spectrum



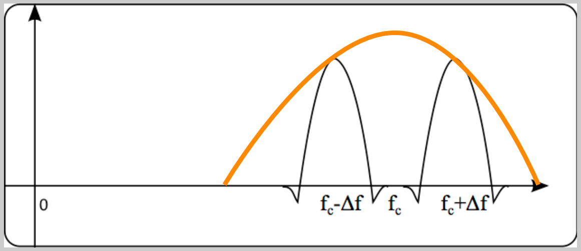
Non-Coherent Demodulator



- Complex BP filters around frequencies



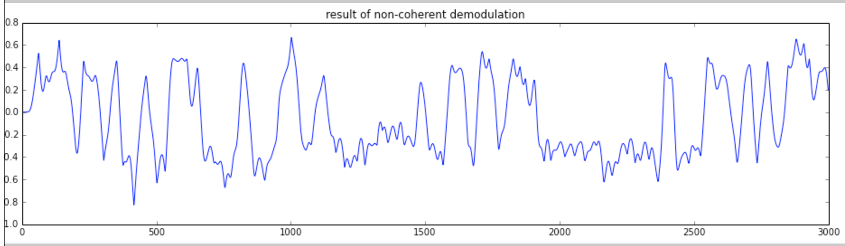
FM Demodulator



- Complex bandpass filter
- Compute Phase derivative to get frequency
- Low-pass filter again with a BW of 1200hz corresponding to bit rate

Bit Error Rate

- When adding noise, things are not so nice

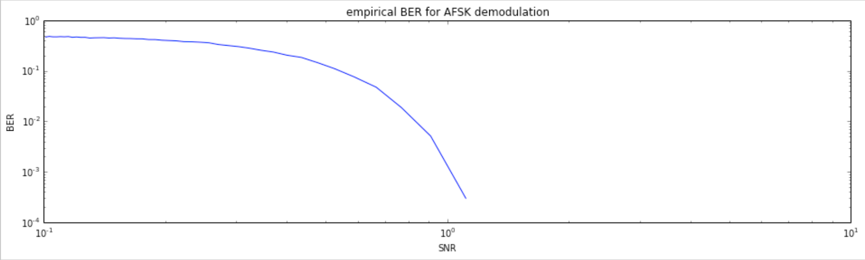


- Compute % or bits incorrectly decoded with respect to total bit sent.

```
• 'BER of non-coherent:', 0.0021 in this case
```

Bit Error Rate Curves

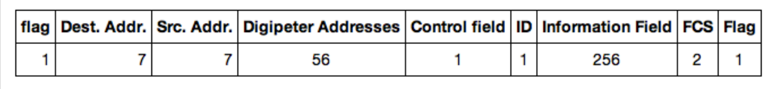
- Compute BER vs SNR



- Compare between parameters and methods.

AX.25

- Link Layer packet based protocol
- Used by ham radio, based on X.25



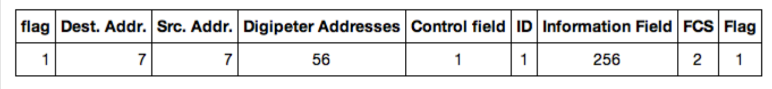
flag	Dest. Addr.	Src. Addr.	Digipeter Addresses	Control field	ID	Information Field	FCS	Flag
1	7	7	56	1	1	256	2	1

  - NRZI: 0 is encoded in change, 1 is no change 11011000 is converted to 11000101
  - Bit stuffing: include a ‘0’ every 5 ‘1’s to guarantee signal change -- help syncronization
  - Flag: 01111110 at beginning an end. The only sequence with 6 ‘1’s.
  - FCS field for checksum error detection

Automatic Positioning and Reporting System

- Ham packet system for real-time tactical digital communication
- Based on AX.25
- Many commercial products implementing APRS
- National frequency 144.39MHz (ch-117)
- ISS packet: 145.825 (ch-50)

APRS Packet



flag	Dest. Addr.	Src. Addr.	Digipeter Addresses	Control field	ID	Information Field	FCS	Flag
1	7	7	56	1	1	256	2	1

- Dest address: APDSP (software version)
- Source address: Your call sign
- Digipeter addresses - Wide2-2/ Wide1-1
- Control field (UI X.25 packet) : \x03
- ID: \xF0

APRS Information Field

- 256 Bytes
- Messages:
  - :ALL------:Everyone will capture this 64 byte message tex
  - :KK6MRI---:This message will only show on Miki's APRS enabled Yaezu VX-8dr radio screen
  - :EMAIL----:mlustig@eecs.berkeley.edu I sent you an email Miki through an OpenAPRS node!
- Position:

! or = symbols	Lattitude 8 chars	/	Longitude 9 chars	icon 1 char	Comment max 43 chars
=	3752.50N	/	12215.43W	K	Shows a school symbol on Cory Hall position
=	3752.45N	/	12215.98W	[	Shows a person walking on Oxford and Hearst
=	2759.16N	/	08655.30E	[	I'm on the top of the world! (Mt. Everest)

  - =3752.50N/12215.43WKShows a school symbol on Cory Hall position

- Status   (starts with a '>')
  - >I like radios

Generate APRS packet

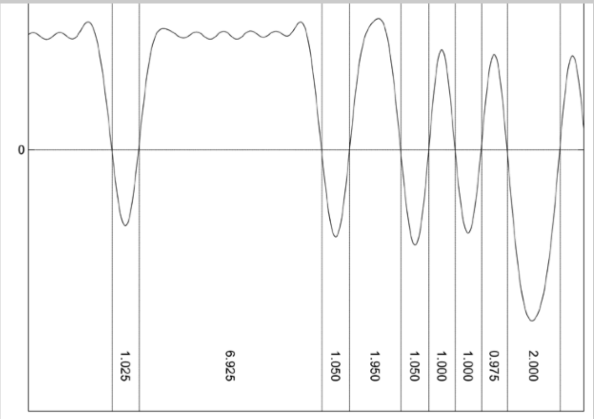
```
import ax25
callsign = "KK6MRI"
Digi =b'WIDE1-1,WIDE2-1'
dest = "APDSP"

# Uncomment to Send Email
info = ":EMAIL    :mlustig@eecs.berkeley.edu What a great lab!"

# uncomment to report position
info = "=3752.50N/12215.43WKThis is Cory Hall!"

# uncomment to send a status message
# info = ">I like radios"


packet = ax25.UI(
                        destination=dest,
                        source=callsign,
                        info=info,
                        digipeaters=Digi.split(b','),
        )
print(packet.unparse())
```

APRS packet

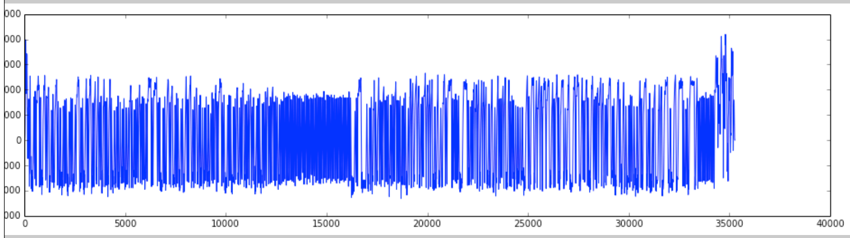
```
bitarray('0111111001000001000001010001000101100101000
001010000001000000110011010010110100100110110010111
0010010010101001001000001100111010101001001000100
01010100010100011000000010010001100111010101001001
0001000101010001001001100000001011000110110000000000
0011111001111001100110011101100101011000100110001111
01001010110000000110001110010111101001000110001001 1
0001001100100011001010110001110100001011001100110 0
1110101011010010001010100000101101001011011001110 00
0001001001011011001110000000100110000101111011001 00
1110100111100000001000001001010000110001101100011 01
101000010010100110010001000111111 0')
```

Decode APRS packets

- **From: Sivan Toledo, 4X6IZ**
- **Look at zero-crossing**



Packet from ISS



```
•   Dest: CQ    0 | Source: RS0ISS | Digis:   | >ARISS - International Space Station |
```

Implement Stream Processing

- Data comes in
- Process in chunks
- Make sure overlaps are taken care of

- Write an application:
  - Decode in real time
  - Interactive text messaging