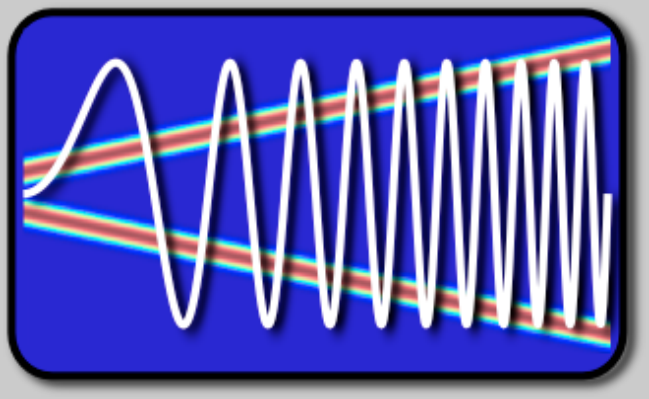
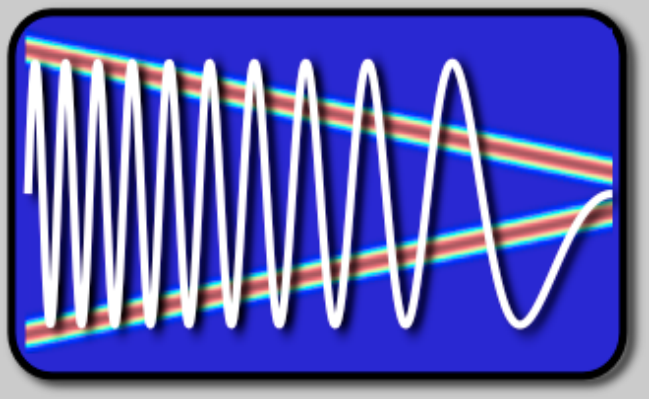


EE123



Digital Signal Processing

Lecture 32
Lab 5
AFSK AX.25 and APRS

M. Lustig, EECS UC Berkeley

Announcements

- Lab 5 Part II is out, Due 04/21
- Lab 5 part III will be out Sunday, Due Monday 04/25
- Midterm grades will be posted this weekend.
- Projects:
  - Make 2-3 slides on progress for every meeting with us
  - Slides should show progress and preliminary results, experimentations, and todo's for next meeting

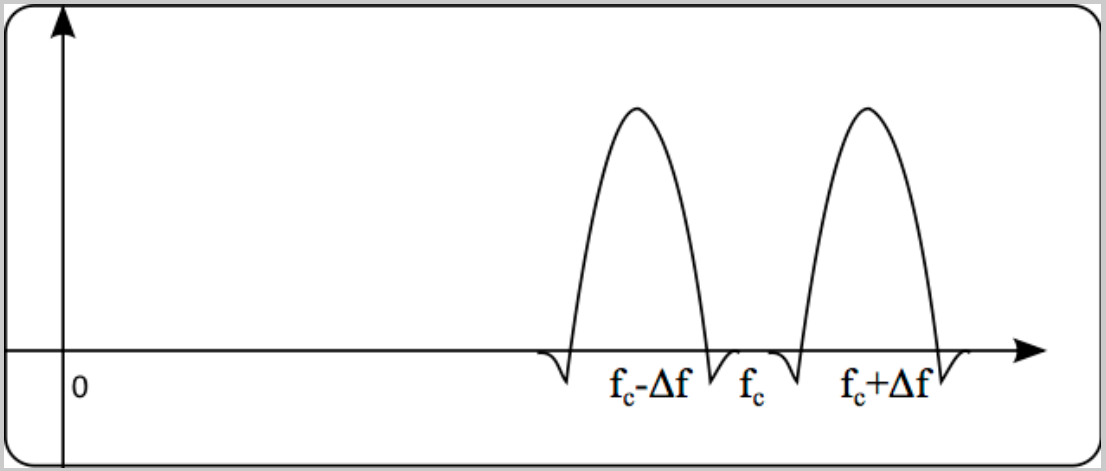
AFSK1200 / Bell 202 modem

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

AFSK 1200



- For spectrum to be narrow, need continuous phase
- NRZ signal, m(t) = 1, or -1 for a duration of a bit, then

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

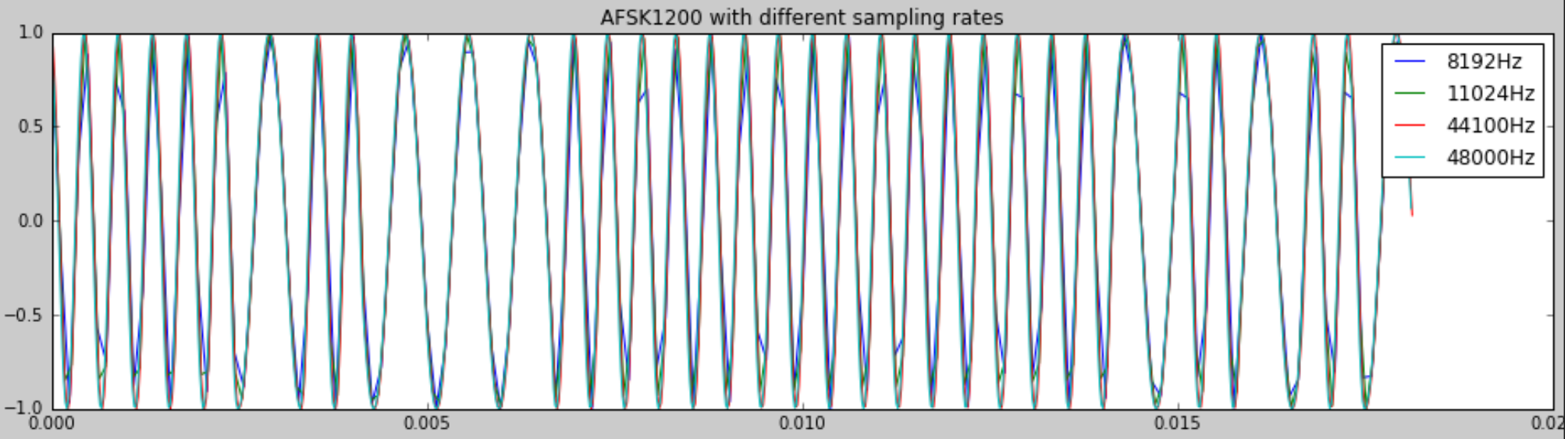
- Frequency: $2\pi f_c + 2\pi\Delta f m(t)$

Write a modulator

- sig = afsk1200(bits,fs)
- If fs does not divide with 1200, then each bit has a fractional sample
  - Can cause drift
- Solution: Generate signal at higher rate that divides with 1200, and downsample

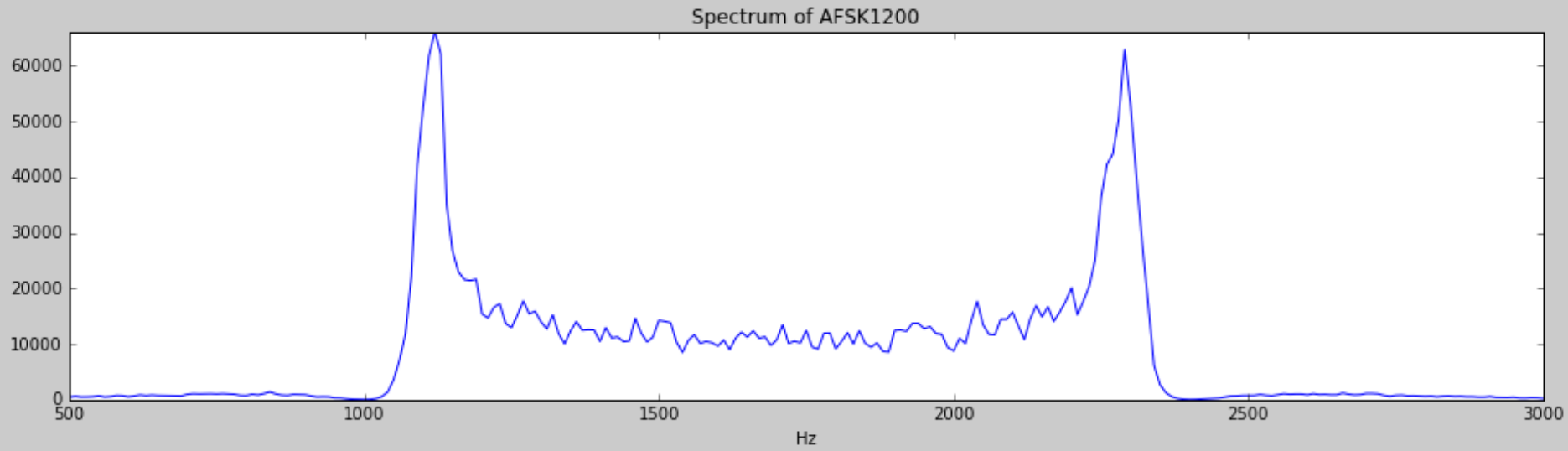
Modulator

- We will give you a sequence to compare to our implementation



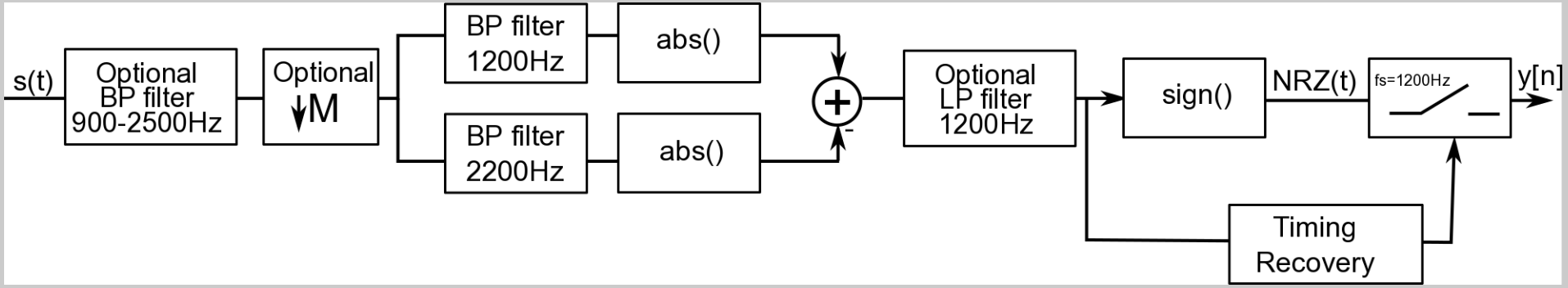
Spectrum of AFSK1200

- Generate random 4096 bits, and compute average power spectrum

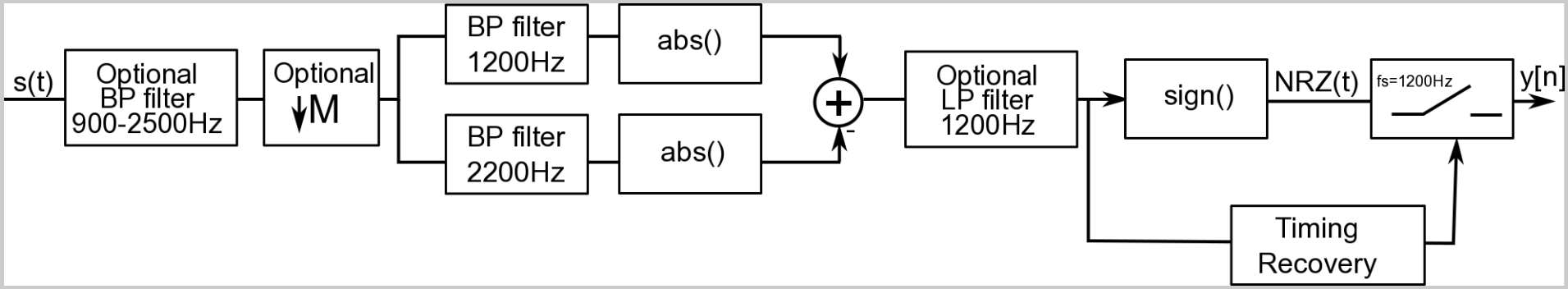


AFSK Demodulation

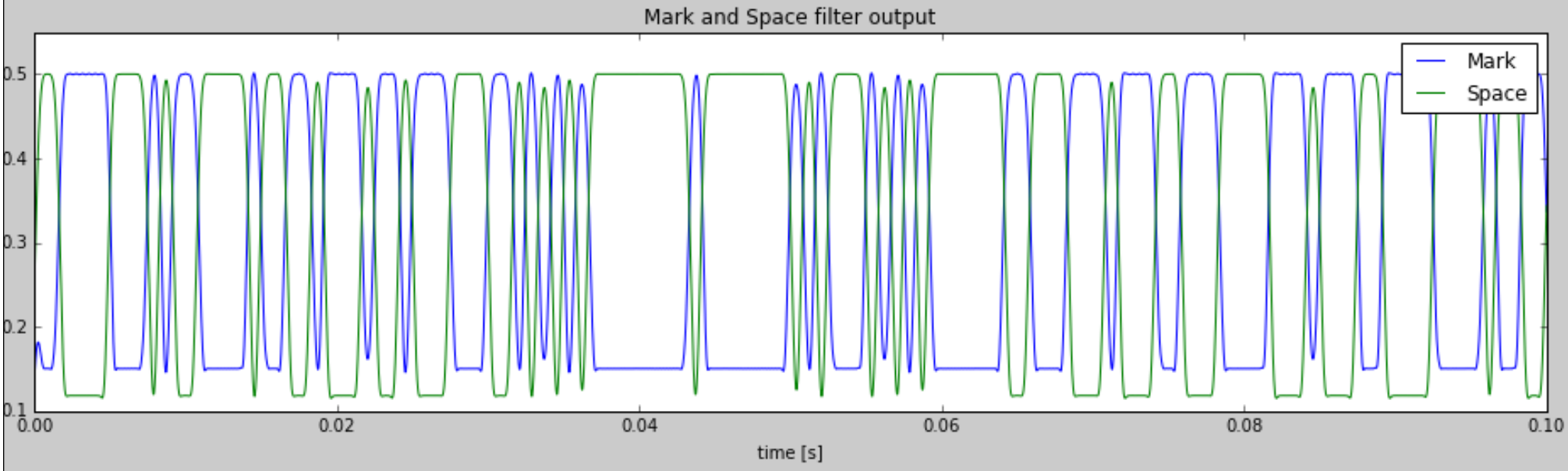
- While afsk is also digital FM -- we will use a non-coherent demodulator
- Based on article by Sivan Toledo (4x6IZ) and DireWolf aprs package.



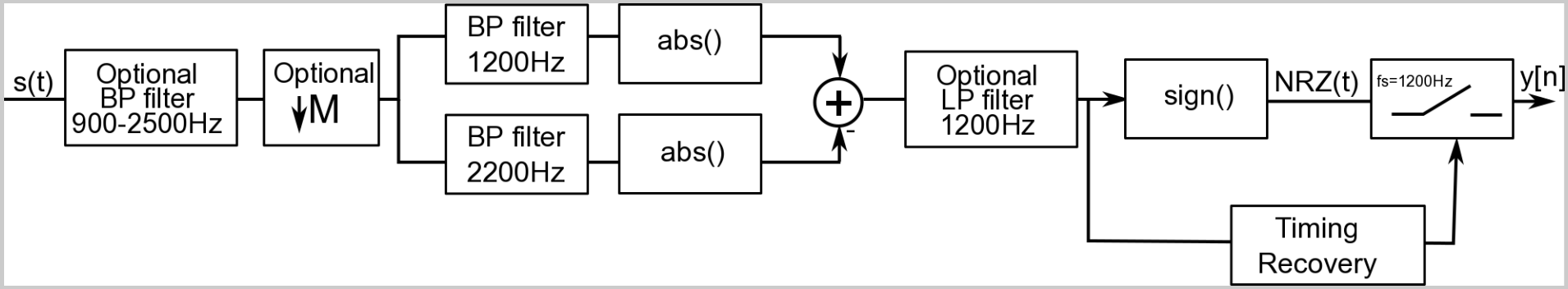
Matched filter detection

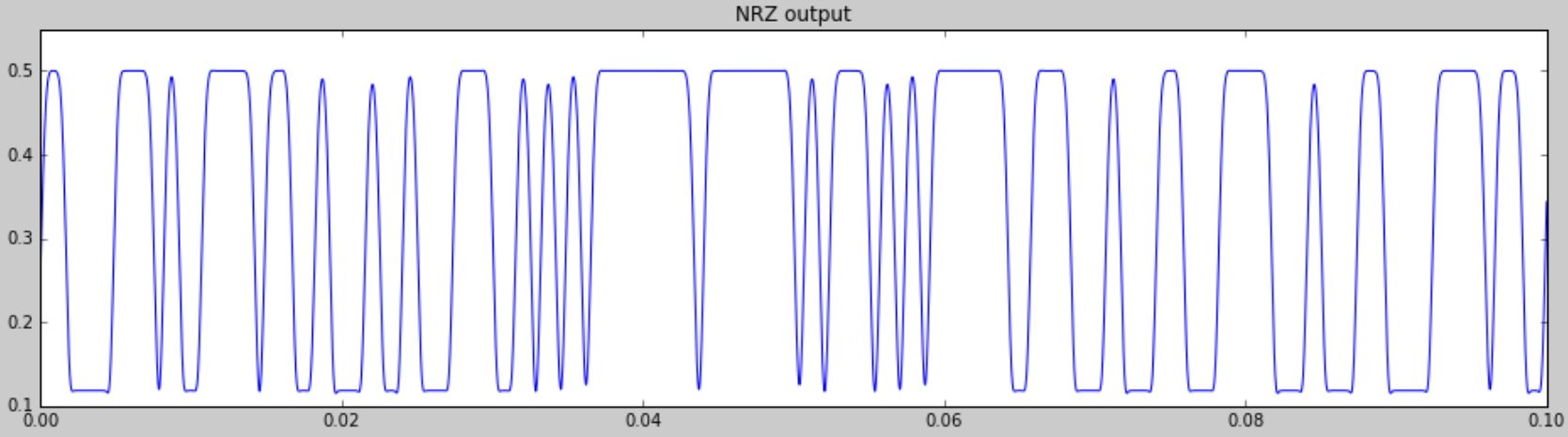


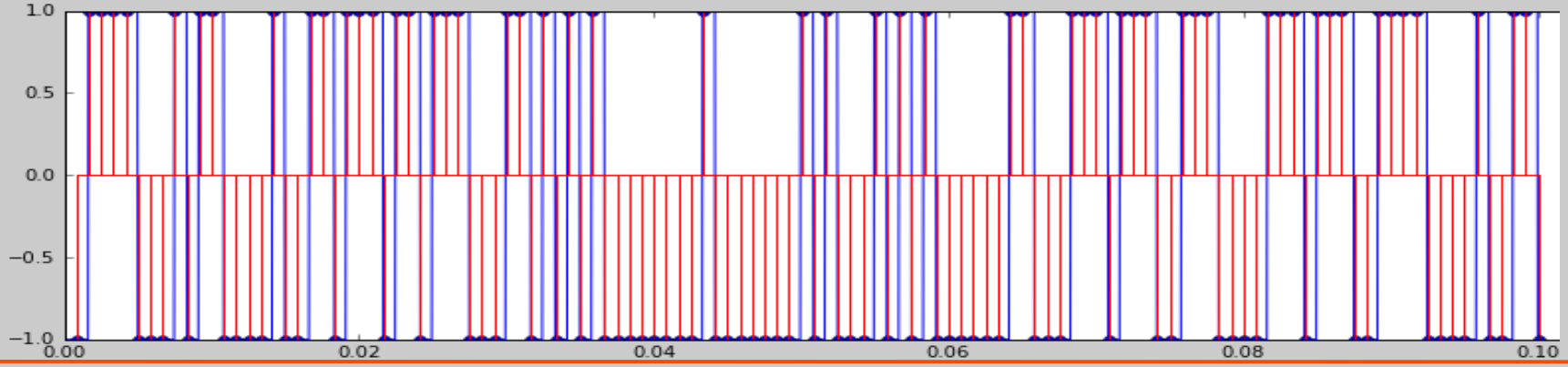
- Design narrow low pass filters around 1200Hz and 2200Hz, compute envelope



Recover NRZ

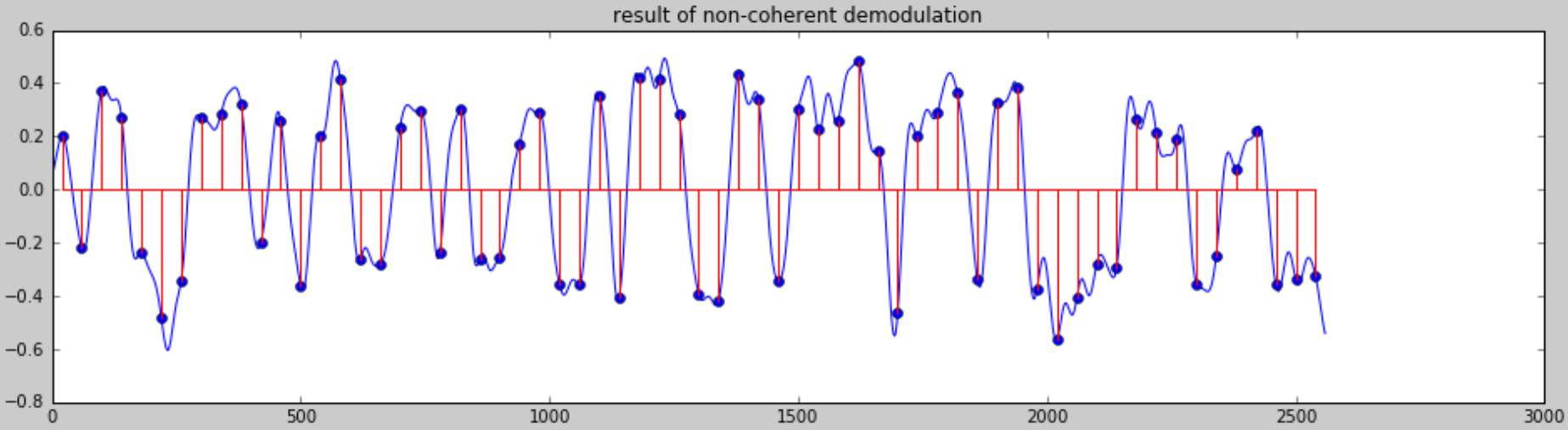

s(t)
Optional
BP filter
900-2500Hz
Optional
↓M
BP filter
1200Hz
BP filter
2200Hz
abs()
abs()
+
-
Optional
LP filter
1200Hz
sign()
NRZ(t)
fs=1200Hz
y[n]
Timing
Recovery


NRZ output
0.5
0.4
0.3
0.2
0.1
0.00
0.02
0.04
0.06
0.08
0.10
time [s]


1.0
0.5
0.0
−0.5
−1.0
0.00
0.02
0.04
0.06
0.08
0.10

Compute Bit Error Rate

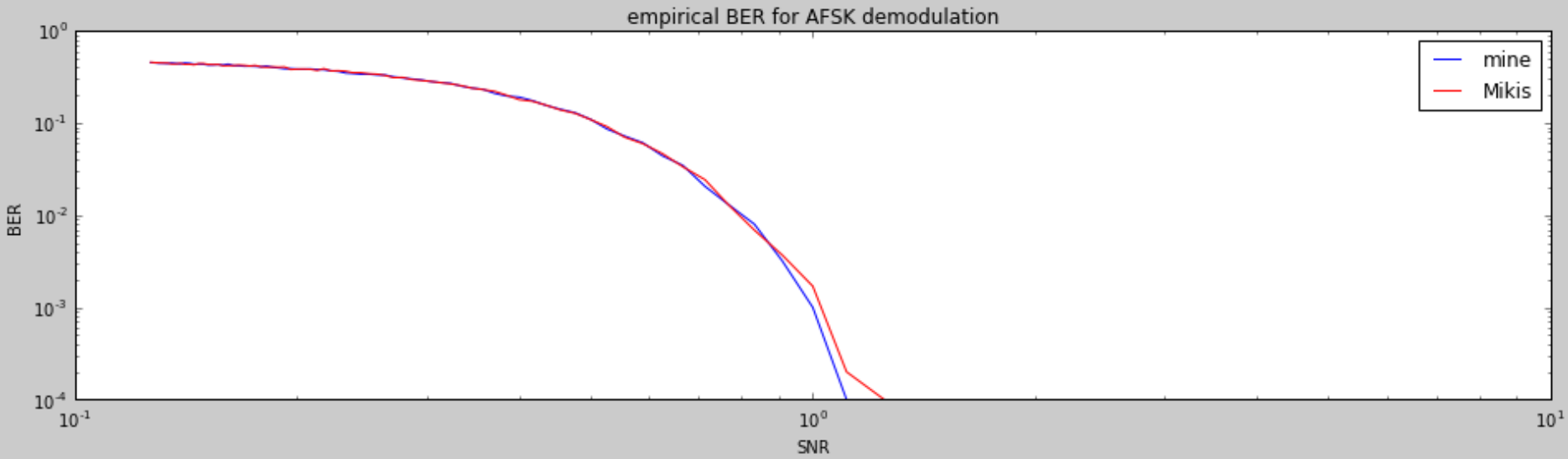
- BER = #altered bits/ Total bits
- Simulate noisy data with gaussian noise
- Run for 10000 long bitstream




```
('BER of non-coherent:', 0.0012)
```

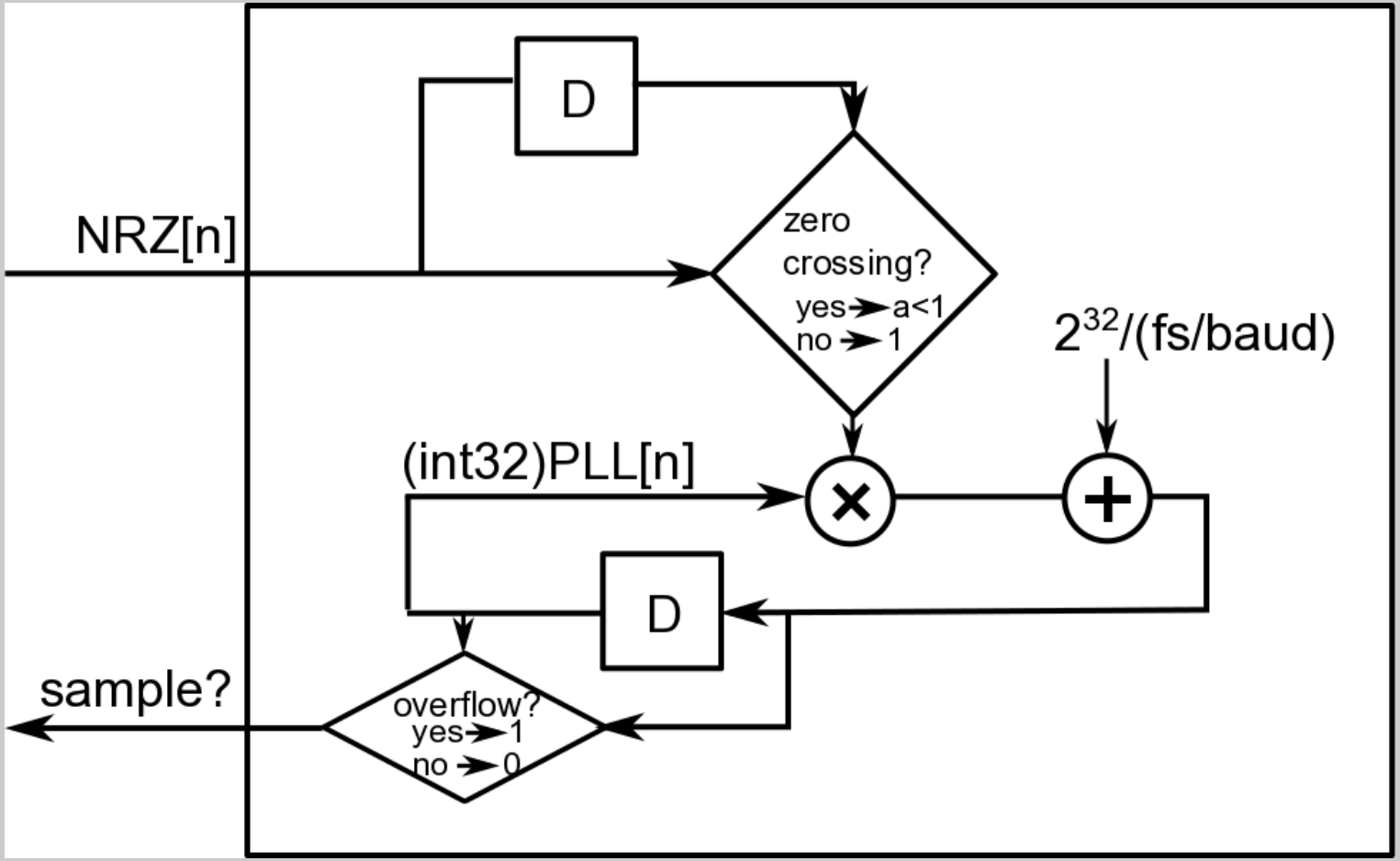
Compute empirical BER curves

- Simulate, and compare to Miki's
- Should be similar performance!

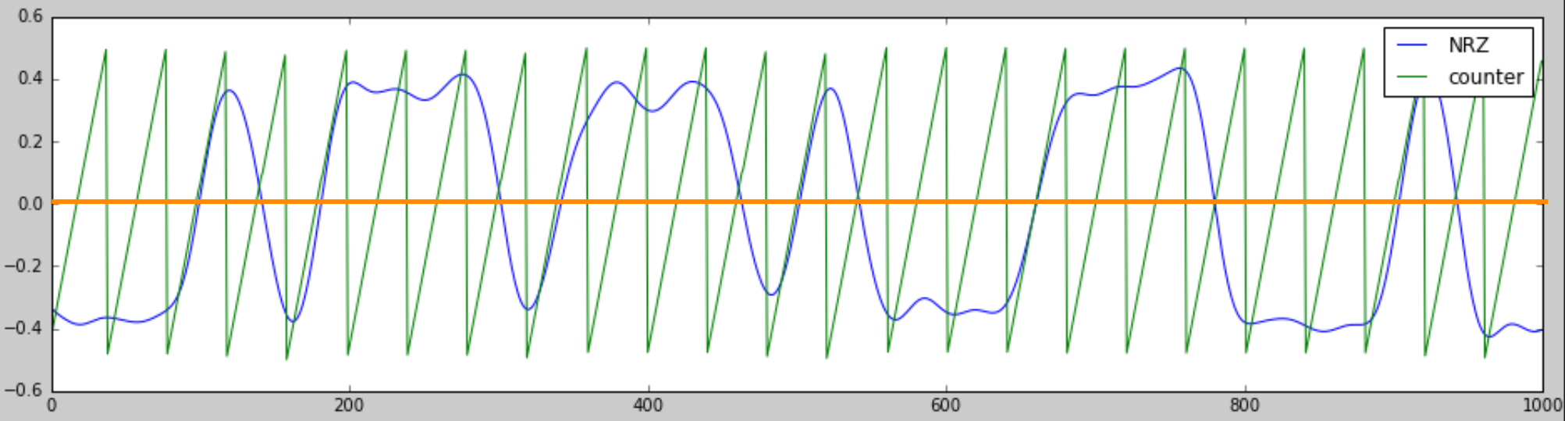


Timing Recovery

- Where to sample???????
- Implement a simple phase-locked loop (PLL)



PLL



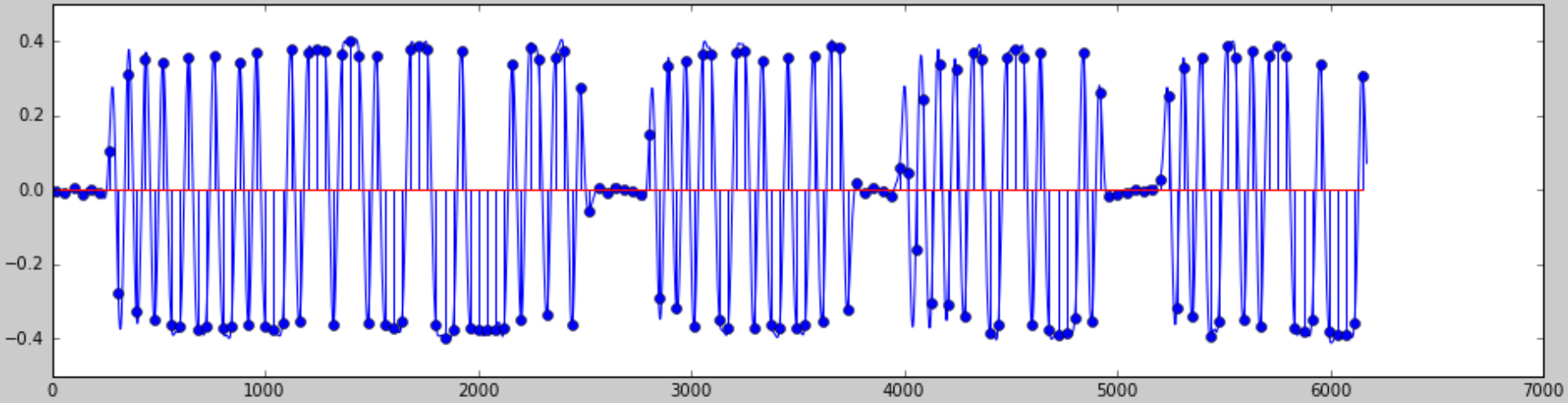
$$c[n] = c[n-1](1 - (\text{sign}(m[n])! = \text{sign}(m[n-1])) * 0.25) + \Delta c$$

- minimizes |C[n] - m[n]|
  - for zero-crossings of m[n]
- C[n] “nudges” is zero-crossing not aligned
- First order non-linear difference eqn.

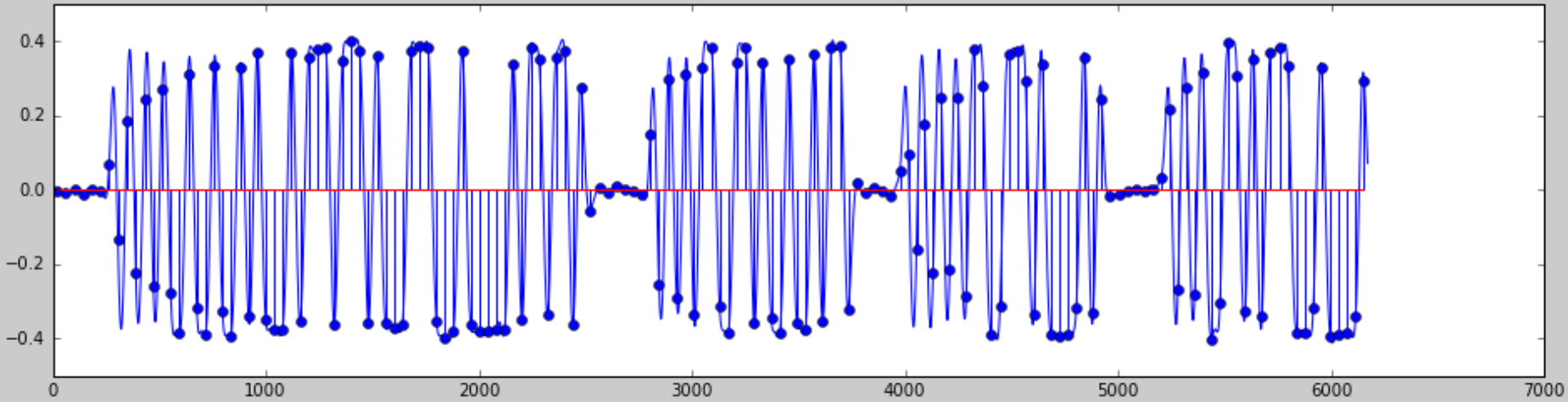
$$c[n] = 0.75c[n-1] + \Delta c \qquad H(z) = \frac{\Delta c}{1 - 0.75z^{-1}}$$

PLL Lock

$a = 0.75$

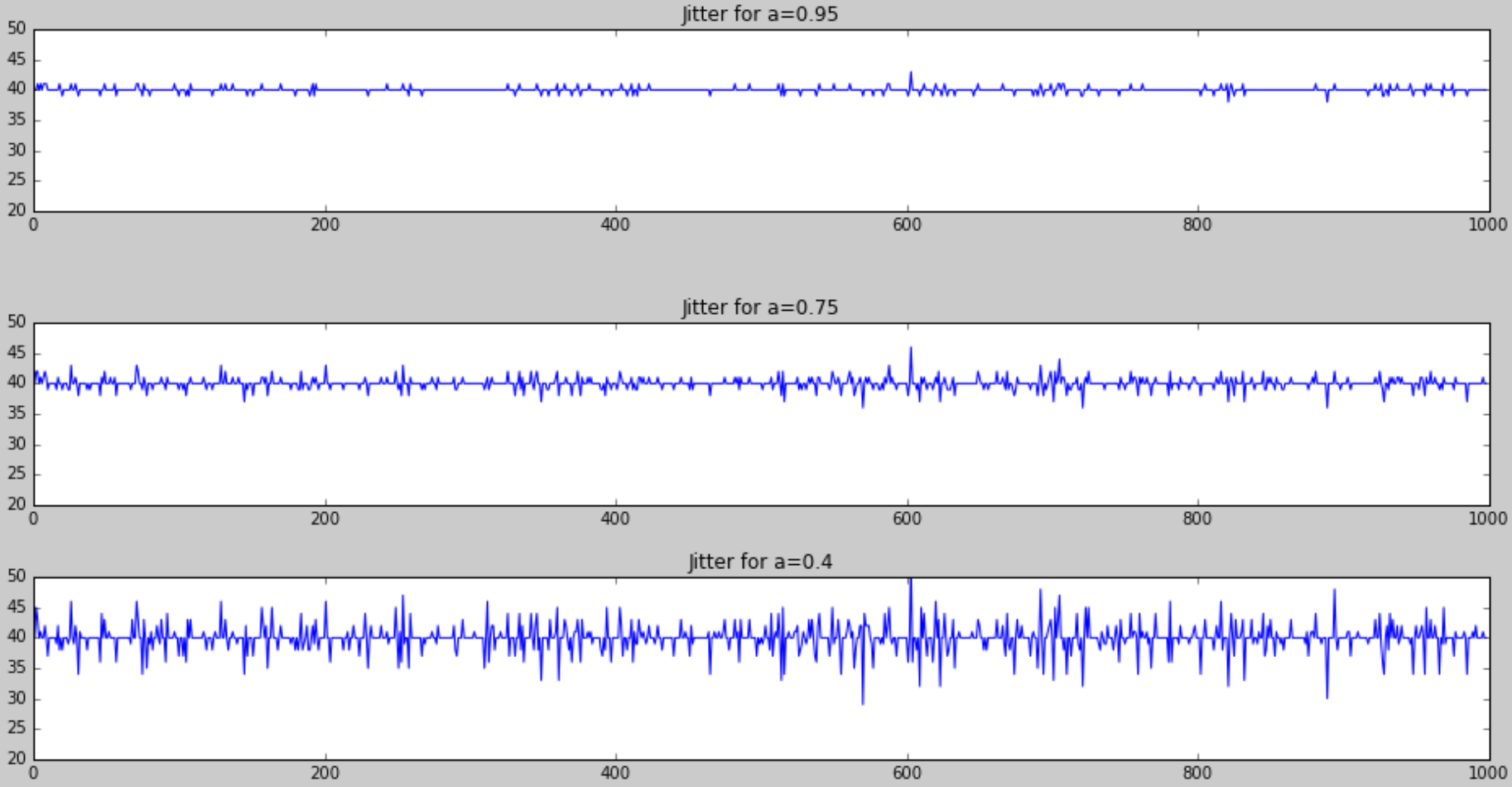


$a = 0.9$



PLL Jitter

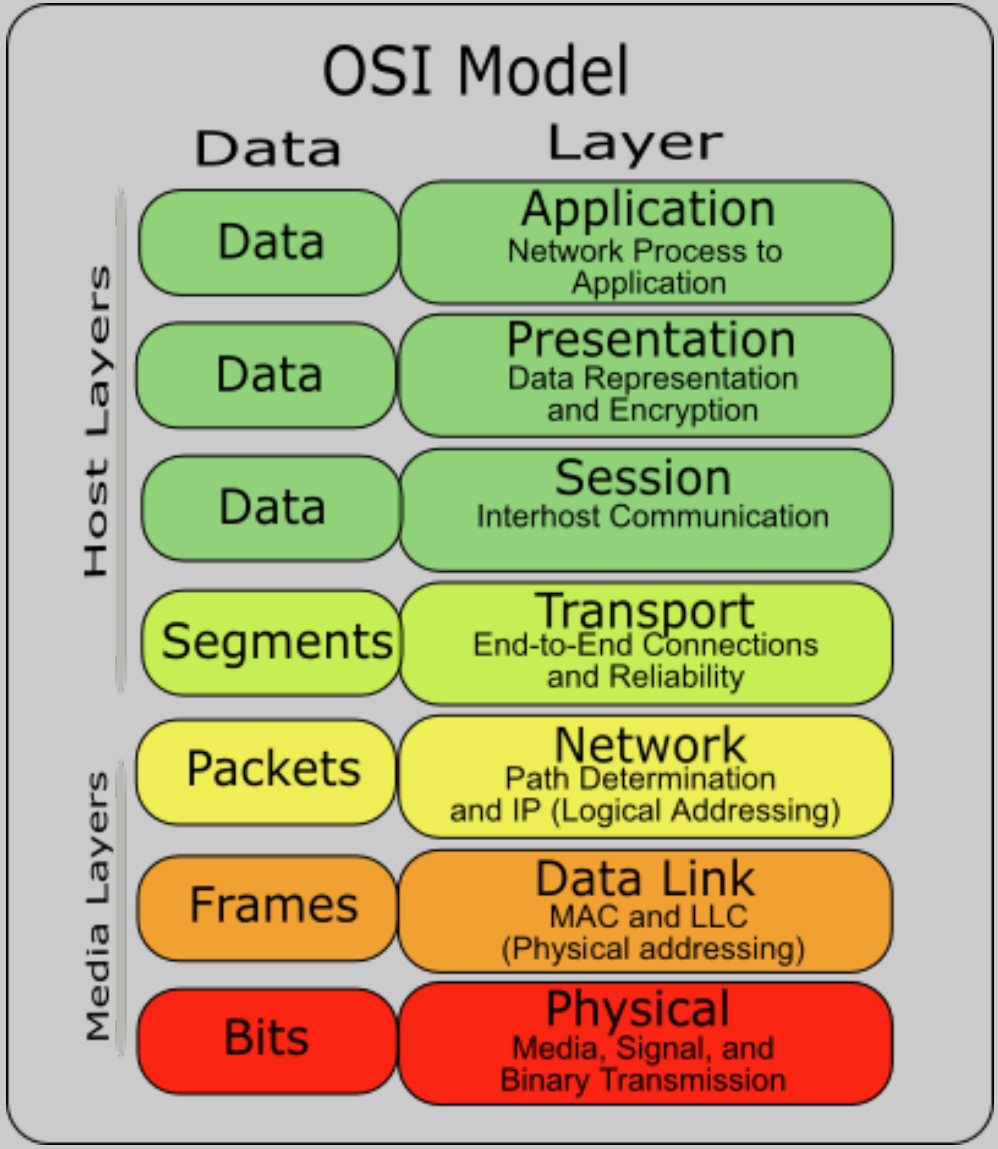
- Estimate timing on noisy data



Lab 5 Part C - 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.

Network Layers



https://commons.wikimedia.org/wiki/File:Osi-model-jb.png

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 synchronization
  - Flag: 01111110 at beginning and 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)
- Internet aggregation and services
  - –Email, SMS, geo-location

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 - Wide1-1,Wide2-1
- 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!
  - :SMSGTE---:@5551231234 I'm sending this number and SMS message
  - :CQSRVR---:CQ EE123 Starting an EE123 chat group
- 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"

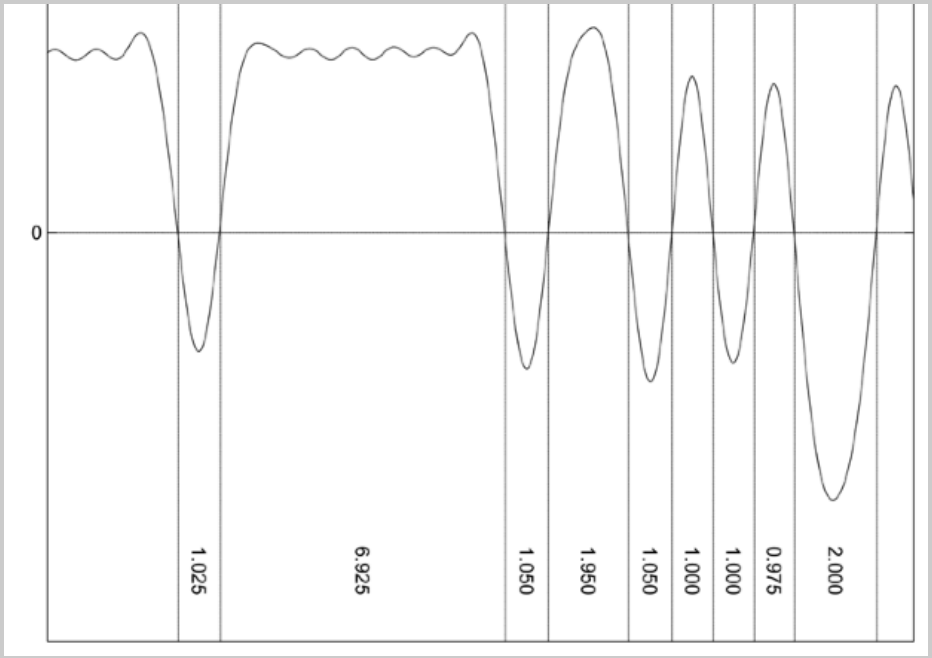
- packet = ax25.UI(
- destination=dest,
- source=callsign,
- info=info,
- digipeaters=Digi.split(b','),
-)
- print(packet.unparse())

APRS packet

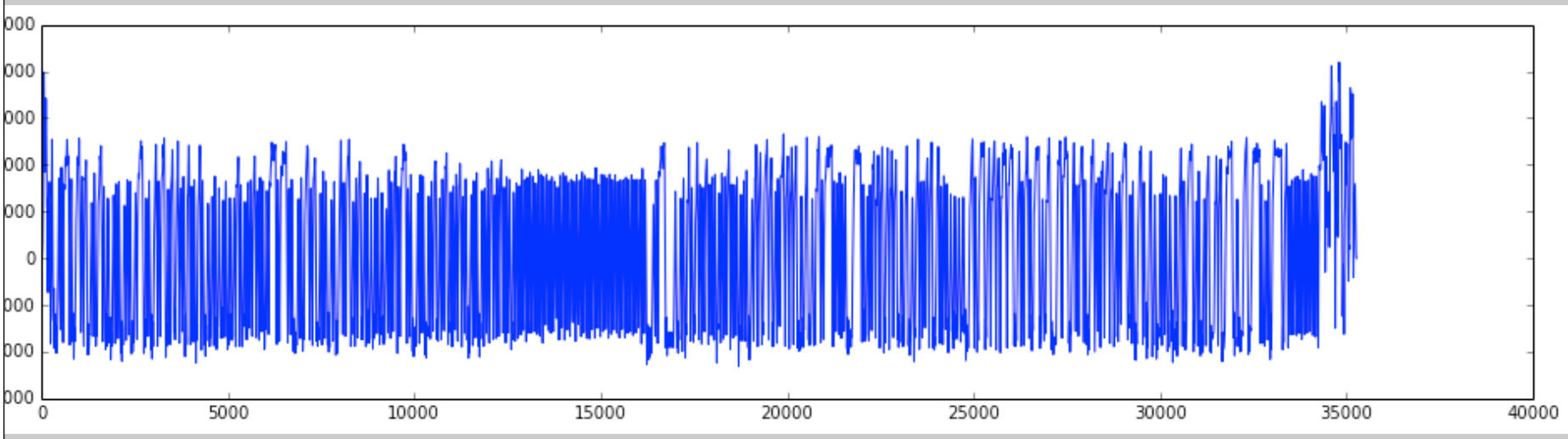
- bitarray('01111110010000010000010100010001011001010000
  00101000000100000011001101001011010010011011001011
  00100100101010010010000011001110101010010010001000
  01010100010100011000000010010001100111010101001001
  00010001010100010010011000000010110001101100000000
  00111110011111001100110011101100101011000100110001111
  01001010110000001100011100101111010010001100010011
  00010011001000110010101100011101000010110011001100
  11101010110100100010101000010110100101101100111000
  00010010010110110011100000010011000010111101100100
  11101001111000000100000100101000011000110110001101
  10100001001010011001000100011111110')

Decode APRS packets

- **Often flag sent a few times**
- **PLL synchronizes on zero-crossings, sends data to decoder**
- **Decoder implements a state machine**
  - **look for 3 consecutive flags**
  - **Start collecting bits**
  - **end packet with a flag**
- **Check FCS field for errors**



Packet from ISS



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

Implement Stream Processing

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

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

GUI

```
00:00:05:15 : KG6NUB]CQ,RS0ISS*,qAS,KG6HSQ-2::KK6MRI :Wow, impressive!
00:00:05:48 : KK6MRI]APDSP,RS0ISS*,qAS,KG6HSQ-2::KG6NUB :Using a measure tape antenna{10
00:00:06:03 : KK6MRI]APDSP,RS0ISS*,qAS,KG6HSQ-2::KG6NUB :csn we reschedule for tomorro 2:40pm?{09
00:00:06:44 : KG6NUB]CQ,RS0ISS*,qAS,KG6HSQ-2::KK6MRI :Seems to be working great !
00:00:06:53 : KG6NUB]CQ,RS0ISS*,qAS,KG6HSQ-2::KK6MRI :ack07
00:00:06:55 : KK6MRI]APDSP,RS0ISS*,qAS,KG6HSQ-2::KG6NUB :this is my ee123 python app{07
00:00:07:23 : KG6NUB]CQ,RS0ISS*,qAR,N6VUD-2::KK6MRI :Yes, almost too easy! :-)
00:00:07:36 : KG6NUB]CQ,RS0ISS*,qAR,N6VUD-2::KK6MRI :ack05
00:00:08:10 : RS0ISS]CQ,qAR,N6VUD-2:]ARISS - International Space Station
00:00:08:37 : KG6NUB]CQ,RS0ISS*,qAR,N6VUD-2::KK6MRI :Thanks for QSO from CM87, 73!
00:00:08:43 : KK6MRI]APDSP,RS0ISS*,qAR,KJ6VCP-2::KG6NUB :qso!{03
00:00:08:52 : KG6NUB]CQ,RS0ISS*,qAR,N6VUD-2::KK6MRI :ack02
00:00:08:55 : KK6MRI]APDSP,RS0ISS*,qAR,N6VUD-2::KG6NUB :hi sawson!
00:00:08:59 : KK6MRI]APDSP,RS0ISS*,qAR,N6VUD-2:=3752.50N/12215.43WKEE123 Rocks!
00:00:09:10 : KG6NUB]CQ,RS0ISS*,qAR,N6VUD-2::KK6MRI :Hello from CM87, qsl?
00:00:10:10 : RS0ISS]CQ,qAR,N6VUD-2:]ARISS - International Space Station
```

APRS123

```
2015-04-19 22:11>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Hello from CM87, qs
l?
2015-04-19 22:11>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Hello from CM87, qs
l?
2015-04-19 22:12>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack02
2015-04-19 22:12>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack02
2015-04-19 22:12>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Thanks for  QSO fro
m CM87, 73!
2015-04-19 22:12>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Thanks for  QSO fro
m CM87, 73!
2015-04-19 22:12>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Thanks for  QSO fro
m CM87, 73!
2015-04-19 22:12>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack04
2015-04-19 22:12>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack04
2015-04-19 22:12>  APP:CQ    0   FRM:RS0ISS   MSG:>ARISS - International Space S
tation
2015-04-19 22:13>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack05
2015-04-19 22:13>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack05
2015-04-19 22:13>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Yes, almost too eas
y! :-)
2015-04-19 22:13>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Yes, almost too eas
y! :-)
2015-04-19 22:14>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack07
2015-04-19 22:14>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :ack07
2015-04-19 22:14>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Seems to be working
 great !
2015-04-19 22:14>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Seems to be working
 great !
2015-04-19 22:14>  APP:CQ    0   FRM:KG6NUB   MSG::KK6MRI   :Using my Arrow here
.
```

```
hi sawson!
hi sawson!
 qso!qso!
my first through iss
Very cool, thanks for the contact
this is my ee123 python appthis is my ee123 python app
It is. csn we reschedule for tomorro 2:40pm?csn we reschedule for tomorro 2:40pm
?
Using a measure tape antennaUsing a measure tape antenna
Using a measure tape antenna
```

MSG:

Your Callsign: KK6MRI
Digi Path: ARISS
Dest: APDSP
To Callsign: KG6NUB
USB In:

- AirPlay
- Built-in Microph
- Built-in Output
- C-Media USB Headphone Set
- Soundflower (2ch)
- Soundflower (16ch)
- ProcasterAudioredirector

USB Out:

- AirPlay
- Built-in Microph
- Built-in Output
- C-Media USB Headphone Set
- Soundflower (2ch)
- Soundflower (16ch)
- ProcasterAudioredirector

Speaker:

- AirPlay
- Built-in Microph
- Built-in Output
- C-Media USB Headphone Set
- Soundflower (2ch)
- Soundflower (16ch)
- ProcasterAudioredirector

PTT Serial Port: /dev/tty.SLAB_USBtoU

Beacon

Lat/Lon 3752.50N 12215.43W

Symb/Comnt K EE123 Rocks!

Quit

M. Lustig, EECS UC Berkeley

Tips for Debugging

- Check audio device volume on computer and radio
- Turn Squelch off
- Create an audio look without the radio