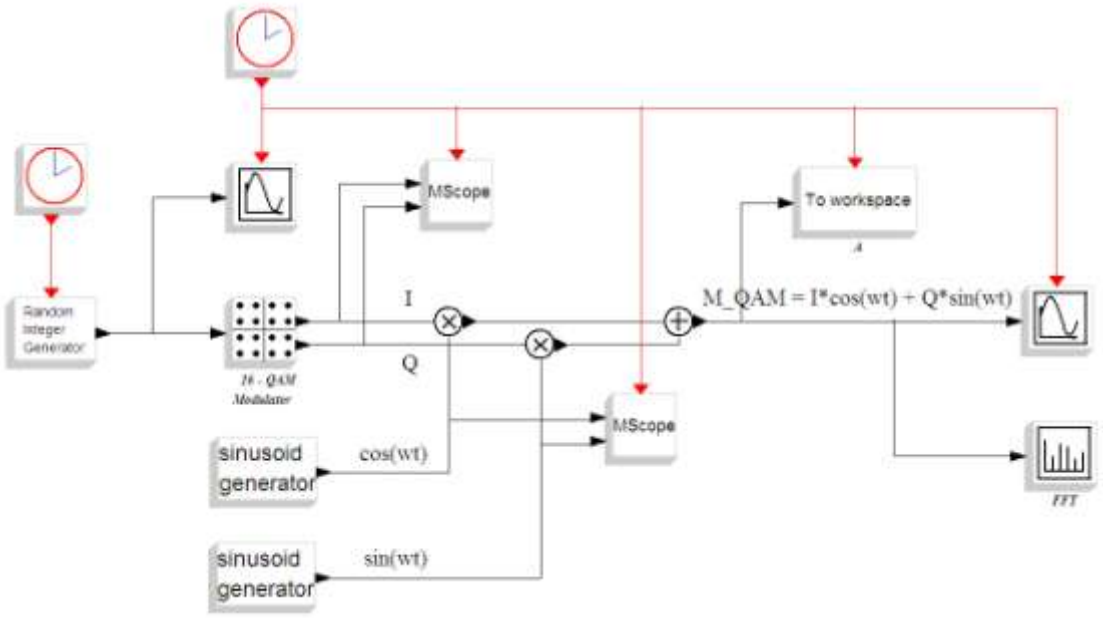
LEARN TELECOMMUNICATIONS BY SIMULATION

Jeremy Clark VE3PKC



ISBN 978-0-9880490-0-0

Clark Telecommunications
jclark@clarktelecommunications.com
https://www.clarktelecommunications.com/simulation.htm

Contents

Contents

1

Introduction

I have always found that learning something is helped by using more than one of your senses if at all possible. Linking several things together can be a powerful way to anchor something in your mind. One of the benefits of being an amateur radio operator is that when you study to get your license you invariably build or work with equipment. So diagrams and equations are not just abstract things, they relate to real things. Computer simulation adds a further dimension to learning. We are able to actually build a system in software and see the same waveforms that we would see on an oscilloscope or spectrum analyzer without having to build or buy the equipment.

In this text I will overview the most common Telecommunications signal types in use nowadays and build models to simulate their operation on a basic level. Wherever possible, I will compress the spectrum into the audio range so that we can not only see but hear the signal as well. To do this I will use freely available open source tools such as ScicosLab and the Modnum Toolbox. I will also include real world signal captures so that we can compare real hardware signals to our simulations.

When we analyze all the commonly used Telecommunications Signals in use today, we will discover that most of the newer ones use common digital modulation techniques such as PSK Phase Shift Keying, QAM Quadrature Amplitude Modulation and OFDM Orthogonal Frequency Division Multiplexing. A separate section is present for each one of these techniques as well as AM Amplitude Modulation, FM Frequency Modulation and PM Phase Modulation.

During my career as a Telecommunication engineer and Professor of Engineering Technology, I have found that keeping things as simple as possible pays off. I have been able to work on design projects and implement them in the field as well. I worked in many countries and learned valuable lessons. I always like doing measurements different ways on different equipment/systems and getting approximately the same answer. I like equipment that is small, rugged, portable and battery powered (many telecom sites only have DC power). Things never quite go as planned, so being flexible and organized is essential.

Building Telecom models in software also provides a bridge to DSP Digital Signal Processing. DSP is used in all branches of Telecommunications. Once a signal is described mathematically in a discrete fashion, then DSP algorithms can be applied. Young people these days love using computers, so tying math and computers together is a win-win process.

This text has been designed as an e-book. In order to see a diagram, just mouse over the figure and the hyperlink will open up the appropriate drawing. Links have been included to descriptive videos and code files. The video intro.mp4 explains how to use the text. Appendix A and video scicos.mp4 describes the installation and use of ScicosLab and the Modnum Toolbox. Read this section first!

2

Open Source Tools & Instrumentation. Scilab, ScicosLab.

In order to build the various models and scripts used in this text, I will use ScicosLab or Scicos with the Modnum Toolbox. This powerful software is Open Source and freely available for download:

ScicosLab Latest Version 4.4.1 as of Apr 2012:
www.scicoslab.org

Modnum Toolbox Version 4.2.2 as of Apr 2012:
www.scicos.org/ScicosModNum/modnum_web/web/eng/eng.htm

Appendix A and video scicos.mp4 contain detailed instructions how to download, install and use this software. Most examples in the text are built two ways. The first way is using a script program in ScicosLab and the second way is using the graphical blockset from the Modnum Toolbox within Scicos.

For real world signal captures, I will use either the RFSPACE SDR-IQ (0 - 30MHz) Software Defined Receiver or the Signal Hound SA44B Software Defined Spectrum Analyzer (0 - 4.4GHz).

RFSPACE SDR-IQ: www.rfspace.com/RFSPACE/SDR-IQ.html

Signal Hound: www.signalhound.com/SA44B.htm

3

Overview of Telecommunications Signals

Let us consider some of the most popular Telecommunications signals in use nowadays. **Fig 3.1** Lists the various systems. There are various ways to classify these signals:

- Personal, Commercial, Industrial, or Military
- Broadcast
- Baseband or Modulated
- Wired (Cable, Fiber Optics, Power Line Carrier) or Wireless
- Terrestrial or Satellite

Popular Telecommunications Signals	Description
Cellular Telephony	Personal, Wireless, PSK or CDMA
Internet	Personal, Wired, OFDM
WiFi	Personal, Wireless, OFDM
HDTV	Broadcast, Wireless, ASK
FM Radio	Broadcast, Wireless, FM
Satellite Radio	Broadcast, Wireless, PSK
AM Radio	Broadcast, Wireless, AM
Telephony Landline	Personal, Wired, Baseband Analog

Fig 3.1 Popular Telecommunications Signals

For example consider Cellular Telephony. Cellular Telephony uses Wireless to communicate between the subscriber cell phone and the cellular base station. Various modulation techniques are used. For instance GSM and CDMA are two widely used techniques. GSM uses a type of digital phase modulation or PSK. Another popular signal is analog FM radio. FM radio has been around since the 1950's. It allows for stereo transmission with excellent sound quality and very simple transmission equipment.

When we consider all the various signals, we see that they all contain some combination of the basic building blocks that we will cover in the next chapters:

- Baseband analog and digital
- AM amplitude modulation analog and digital
- FM Frequency modulation analog and digital
- PM Phase modulation analog and digital
- QAM Quadrature Amplitude Modulation, combination of digital AM & PM
- OFDM Orthogonal Frequency Division Multiplexing, many PSK/QAM signals
- CDMA Code Division Multiple Access

Digital Telecommunications systems are often analyzed according to the OSI or TCP/IP models, **Fig 3.2**. In this text, we will be considering the so called Physical 1 Layer. Upper layers are concerned more with software concepts. The Physical 1 Layer is concerned with the actual signal voltage that is placed on the cable or wireless link.

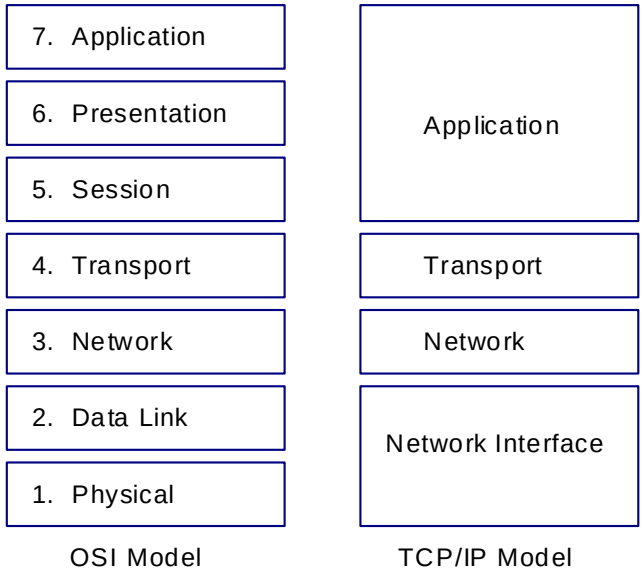


Fig 3.2 OSI & TCP/IP Models

The video over_sigs.mp4 discusses these concepts further.

4

Baseband Analog and Digital Signals

4.1 Baseband Analog - Single Tone

Baseband is a relative term and refers to frequencies that are close to DC and have not been modulated or shifted by a carrier. Human speech as monitored from a microphone contains frequencies from about 20 Hz to 7KHz. Human hearing extends to about 20KHz max. An analog video signal contains frequencies from approx. DC to several MHz. A baseband data signal contains frequencies from approx. DC to the data rate and beyond. The video bb.mp4 introduces these topics.

The simplest of all baseband signals is a single tone. **Fig 4.1** bb_tone.sce is a ScicosLab program to generate a single tone at 1000Hz and plot the waveform in time and plot its spectrum in the frequency domain. **Fig 4.2** bb_tone.cos does exactly the same thing but uses the Scicos graphical block structure. Both files are explained in the demo file bb_analog.mp4. **Fig 4.3** shows the time domain and spectral display of the single tone.

Note that a single tone has a single line spectrum at the tone frequency. Conversely, a line in the spectrum of a signal indicates that the signal contains a tone component at that frequency.

Fig 4.1 Baseband Analog Single Tone bb_tone.sce

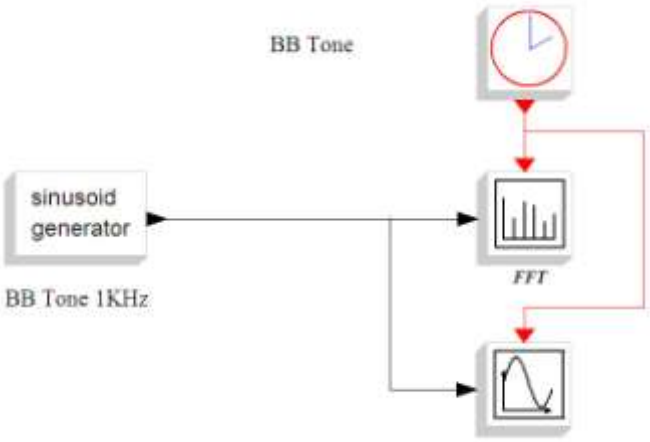


Fig 4.2 Baseband Analog Single Tone bb_tone.cos

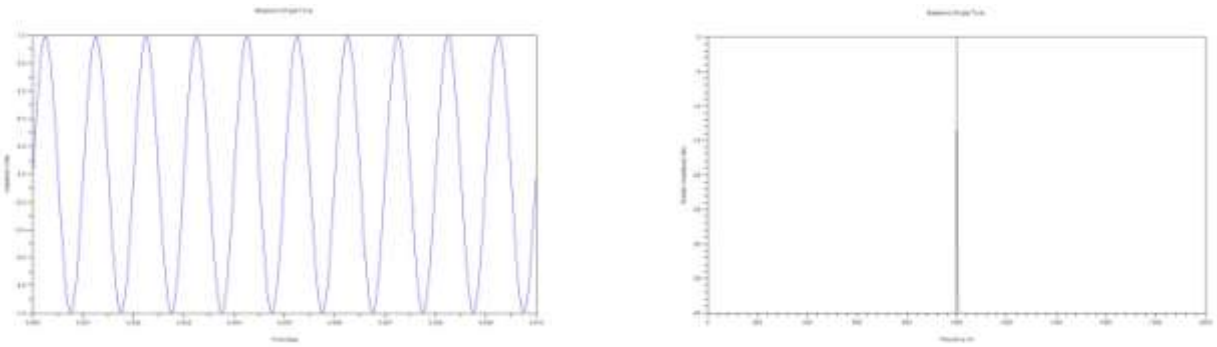


Fig 4.3 Baseband Analog Single Tone bb_tone.sce Scope & Spectral Display

4.2 Analog Baseband - Multiple Tones

In most cases the baseband will consist of a more complicated signal consisting of many tones. Speech for instance consists of a multiple tones and music consists of hundreds of tones. To get a feeling for this, **Fig 4.4** bb_tones.sce & **Fig 4.5** bb_tones.cos show the time and frequency domain displays of a baseband signal consisting of 3 tones with different frequencies and amplitudes:

-Tone 1 = 1KHz @ 3 Volt
-Tone 2 = 2KHz @ 2 Volt
-Tone 3 = 3KHz @ 1 Volt

Fig 4.4 Baseband Analog Tones bb_tones.sce

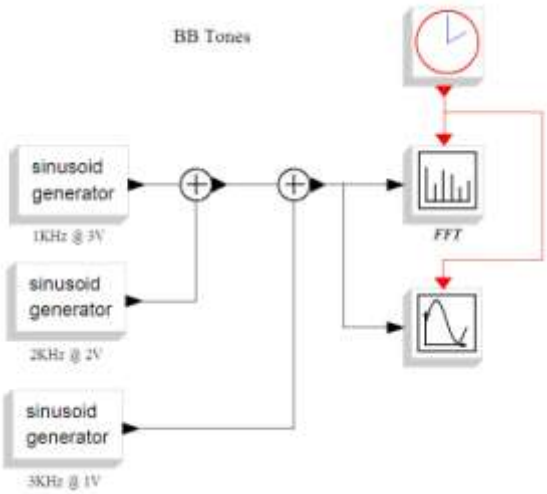


Fig 4.5 Baseband Analog Tones bb_tones.cos

In general, the sum of many sinusoids of different amplitudes and phases but the same frequency will give a sinusoid at that frequency with a different amplitude and phase given by the vector addition of each component. However, the addition of sinusoids of different frequencies, amplitudes and phases will give rise to a non-sinusoidal periodic signal. This is further explained in bb_analog.mp4.

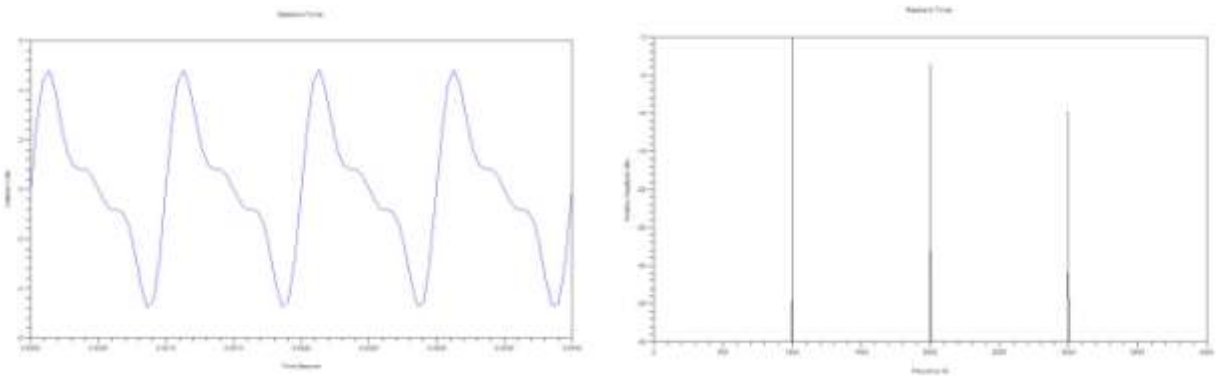


Fig 4.6 Baseband Analog Tones bb_tones.sce Scope & Spectral Display

From **Fig 4.6** we see that the addition of the 1KHz, 2KHz and 3KHz tones gives rise to a non-sinusoidal periodic wave of 1KHz.

Consider the spectrum of the baseband tones in **Fig 4.6.** Note that there is a discrete line for each component frequency. We can convert the ratio of the voltage of each component and see if the relative spectrum amplitude indicates the same thing:

Component	**Relative Amplitude**	**dB=20log(V/Vref)**	**Fig 4.6 Spectrum**
1KHz	3/3 = 1 reference	0 dBc	0 dBc
2KHz	2/3	-3.5dBc	-3.5dBc
3KHz	1/3	-9.5dBc	-9.5dBc

4.3 Baseband Analog - WAV

Now consider what happens when we have a real speech signal consisting of energy at many frequencies. **Fig 4.7** bb_wav.sce & **Fig 4.8** bb_wav.cos examine the speech file 'demo.wav' to examine its waveform and spectrum.

In order to process the signal, we need to know the sample rate and duration. By opening 'demo.wav' in Windows Sound Recorder, the sample rate is 22050Hz and time 36.75 seconds. In **Fig 4.7**, the file is read in using the 'wavread' routine. The file is then limited to 4 seconds = 22050 x 4 = 88,200 samples. An equivalent time vector is created to match this data. The waveform is then plotted and spectrum calculated.

In **Fig 4.8**, a similar process is used. A script file wav_read.sce is used to read in the sound data and associate it with a structure variable **V**. **V** is then read into the block file. **V** contains both the sample data and time data.

Note the nature of a speech sound signal shown in **Fig 4.9.** It is bursty in nature when the vocal chords are active, with relatively long periods of zero activity between bursts. The spectrum indicates that for this speech sample, the majority of signal energy lies between DC and about 3500Hz, with a peak energy around 300Hz.

```
//This file simulates a WAV baseband
//A file 'demo.wav' is read into the workspace
//File is 8.74 seconds long
//Fs = sampling frequency 22050Hz
//J.Clark April2nd-2012
//---------------------------------------------------------

//Read in 'demo.wav' & truncate to 4 secs
//Truncate to 4secs
Vm=wavread('demo.wav');
Vm=Vm(1: 88201);
Fs=22050;
Ts=1/Fs;
t=0:Ts:4;
N=size(t,'*');
clf(0);
clf(1);
//---------------------------------------------------------

//Graph the tone waveform
//scf(0) open graphic window #0
scf(0);
plot(t,Vm)
xtitle("Baseband WAV", "Time Seconds", "Relative Amplitude")
//---------------------------------------------------------

//Calculate the spectrum with fft
//scf(1) open graphic window #1
scf(1);
y=fft(Vm);
sample_rate=1/Ts;
f=sample_rate*(0:(N/2))/N;
n=size(f,'*');
z=max(abs(y(1:n)));
plot2d(f,20*log10(abs(y(1:n))/z))
xtitle("Baseband WAV", "Frequency Hz", "Relative Amplitude dB")
//---------------------------------------------------------
```

Fig 4.7 Baseband Analog WAV
bb_wav.sce

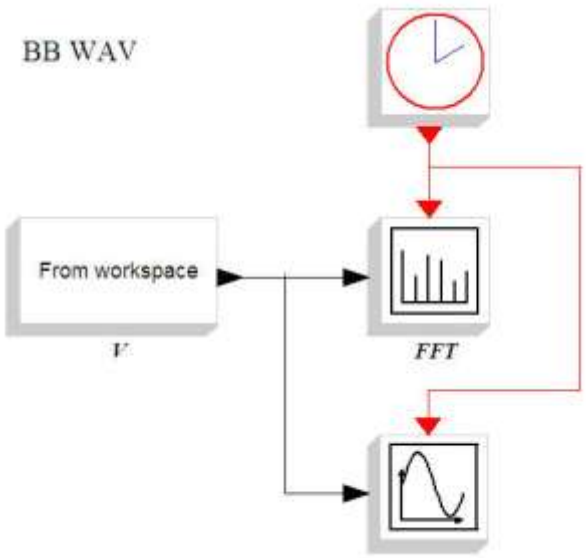


Fig 4.8 Baseband Analog WAV
bb_wav.cos

Note: Copy demo.wav from the /chapter_bb/wav to the desktop for ScicosLab to work

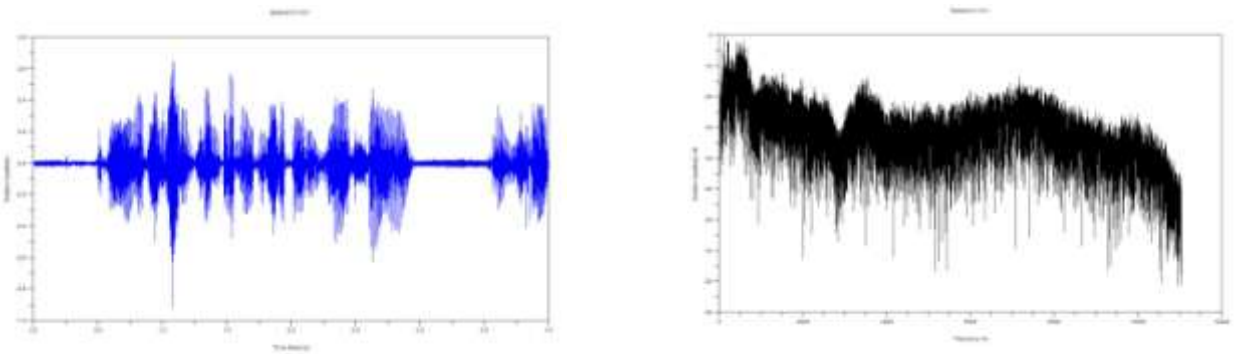


Fig 4.9 Baseband Analog WAV bb_wav.sce Scope & Spectral Display

4.4 Baseband Analog Signal Captures

Now let's look at some real signals to see how they compare to the simulations. The first signal is an audio tone of 1KHz. The second signal is a microphone signal. The Video bb_an_sigcap.mp4 describes the signal captures.

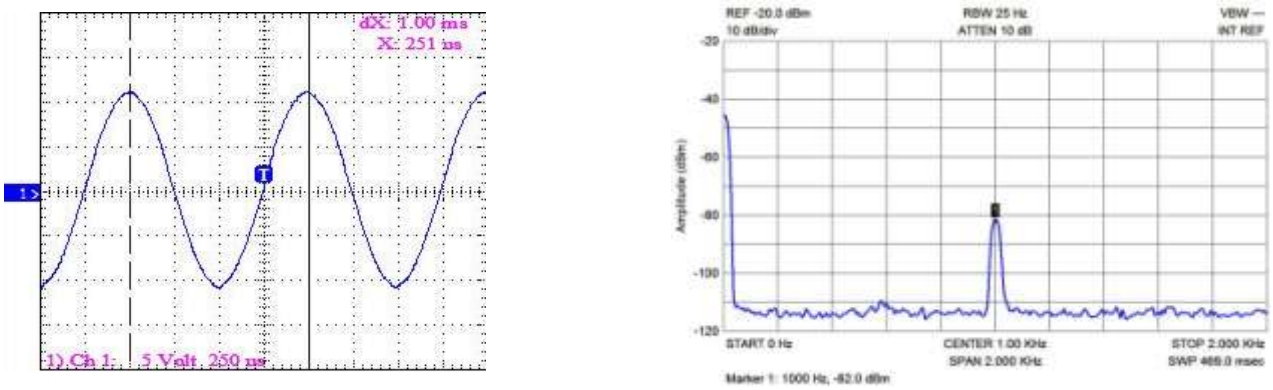


Fig 4.10 Baseband Analog Signal Capture Tone Scope & Spectral Display

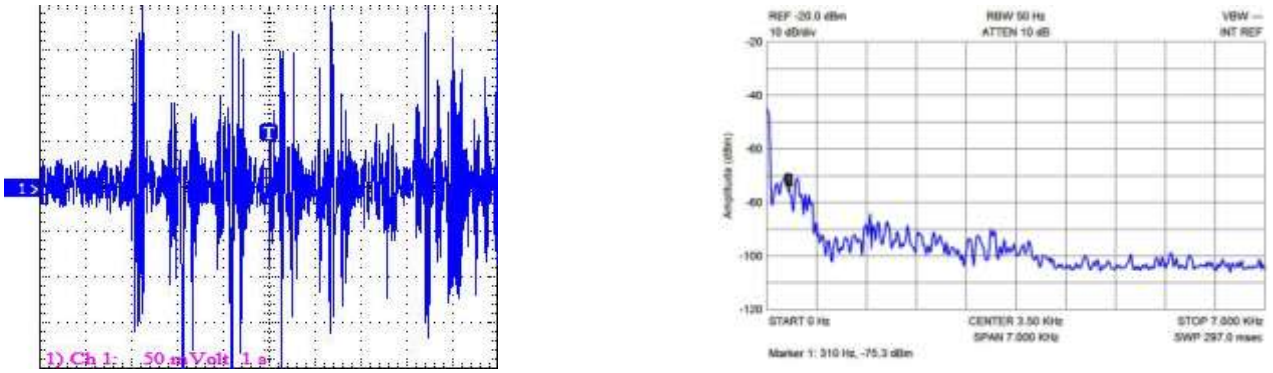


Fig 4.11 Baseband Analog Signal Capture WAV Scope & Spectral Display

4.5 Baseband Digital - NRZ Random Data

So far we have considered a baseband consisting of analog signals. Let's look at a digital data signal. **Fig 4.12** & **Fig 4.13** generate the digital baseband for a NRZ random data signal. The video bb_digital.mp4 describes this.

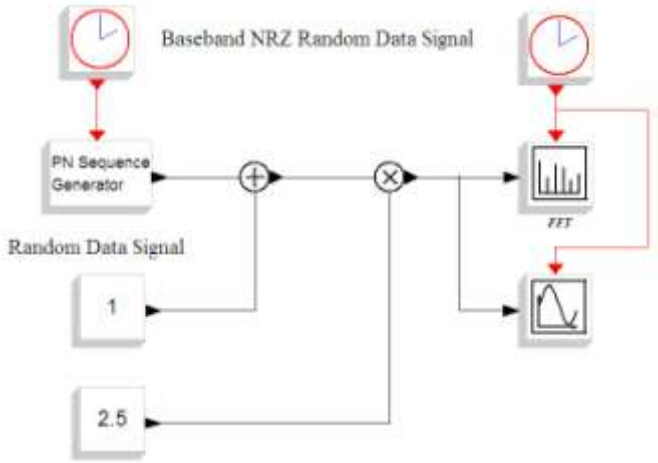


Fig 4.12 Baseband Digital NRZ Data bb_rnd_data.sce

Fig 4.13 Baseband Digital NRZ Data bb_rnd_data.cos

In **Fig 4.12**, random data is generated by the 'rand' function. This generates a random number between 0 & 1. This is declared to be 5 volts if >=0.5, or 0 volts if < 0.5. In **Fig 4.13**, the random data is generated using a PN Sequence Generator. This generates a random data pattern that repeats after 2^N-1 bits. This generates a slightly different random pattern with longer sequences of 0's and 1's.

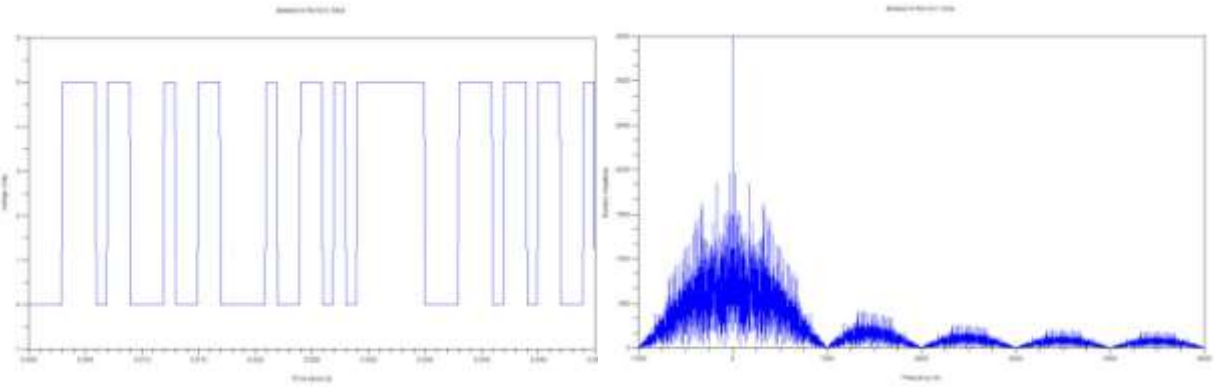


Fig 4.14 Baseband Digital NRZ Data bb_rnd_data.sce Scope & Spectral Display

Note the defining characteristics for NRZ random data. First of all the spectrum contains no discrete lines except at 0Hz. For an NRZ signal, there is a DC component of approx 2.5volts since the signal is on average half way between 0 volts and 5 volts.

The spectrum has a so called (sinx/x) shape **Fig 4.15**, with nulls at exactly the data rate and multiples thereof. The data rate is 1Kbps, so the nulls are at 1KHz, 2KHz etc. Note that the data rate null at 1KHz = 1/(pulse width for a bit) = 1/1msec = 1KHz. In summary:

-DC Component for NRZ data
-1st null at data rate and at integer multiples thereof
-Smooth shape, no spectral lines
-Nulls decreasing amplitude

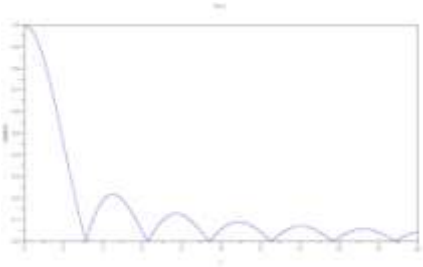


Fig 4.15 Shape = sinx/x

4.6 Baseband Digital - Manchester Random Data

NRZ is the basic data type generated by computing devices. However, it has several disadvantages for transmission on wire pairs etc. Manchester coded data is very popular for cable transmission in Ethernet. In Manchester coding, a '1' is encoded with a +V to -V transition and a '0' is encoded by a -V to +V transition. This ensures that for every data bit, there is equal positive and negative voltage. This means the overall DC voltage is 0 volts. The effective bit time is halved, so the bandwidth or required spectrum is doubled.

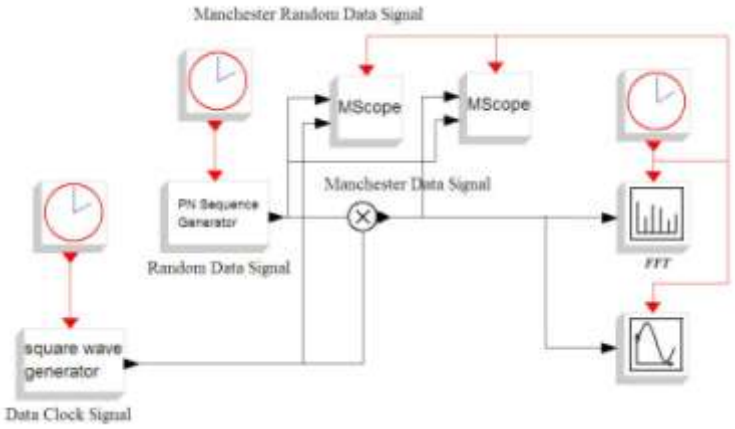


Fig 4.16 Baseband Digital Manchester Data bb_rnd_data_man.sce

Fig 4.17 Baseband Digital Man.Data bb_rnd_data_man.cos

In **Fig 4.17** bb_rnd_data_man.sce both NRZ and Manchester data are generated and compared. We can see the mid bit transitions in Manchester and notice that there is no DC component in the spectrum, but the data null is at twice the data rate now, because of the half pulse width.

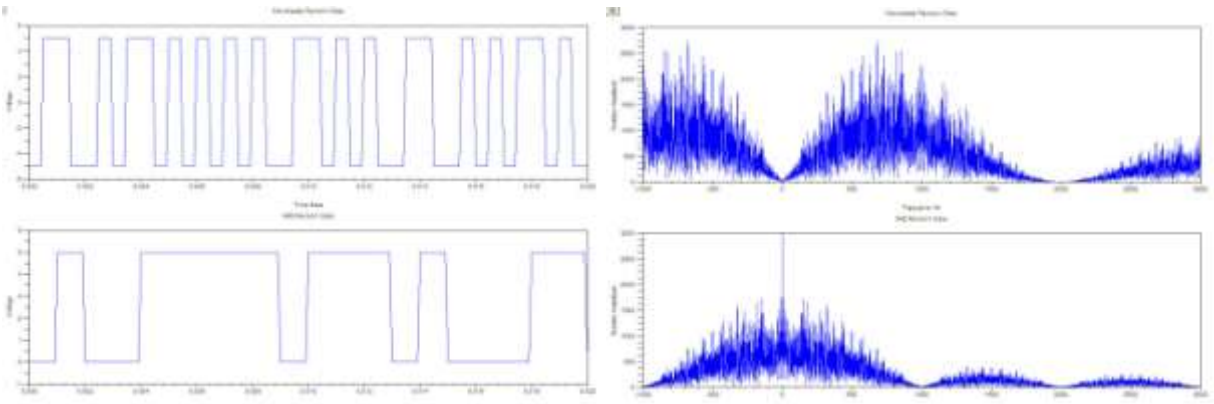


Fig 4.17 Baseband Digital Manchester & NRZ Data bb_rnd_data_man.sce
Scope & Spectral Displays

4.7 Baseband Digital - Signal Captures

Fig 4.18 Baseband Digital Sigcap.
NRZ & Man Data bb_rnd_data_nrz_man.bs2

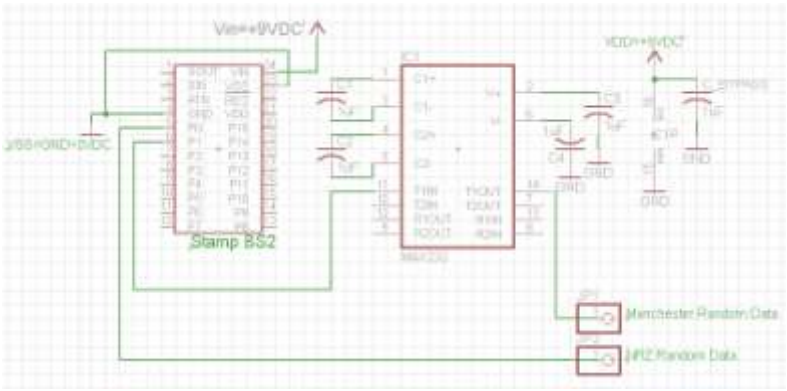


Fig 4.19 Baseband Digital Sigcap.
NRZ & Man Data bb_rnd_data_man.sch

Fig 4.18 shows the BS2 program code used to generate NRZ & Manchester random data on a Stamp Microcontroller. **Fig 4.19** shows the schematic diagram of the BS2 Microcontoller and the MAX232 TTL/RS232 converter used to generate the bipolar voltages used for the Manchester coding. The videos bb_dig_sigcap.mp4 & bb_dig_sigcap_code.mp4 describe the process.

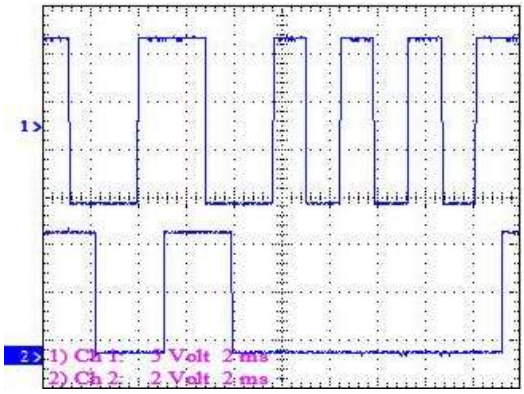


Fig 4.20 Baseband Digital Sigcap. NRZ & Man Random Data Scope Display

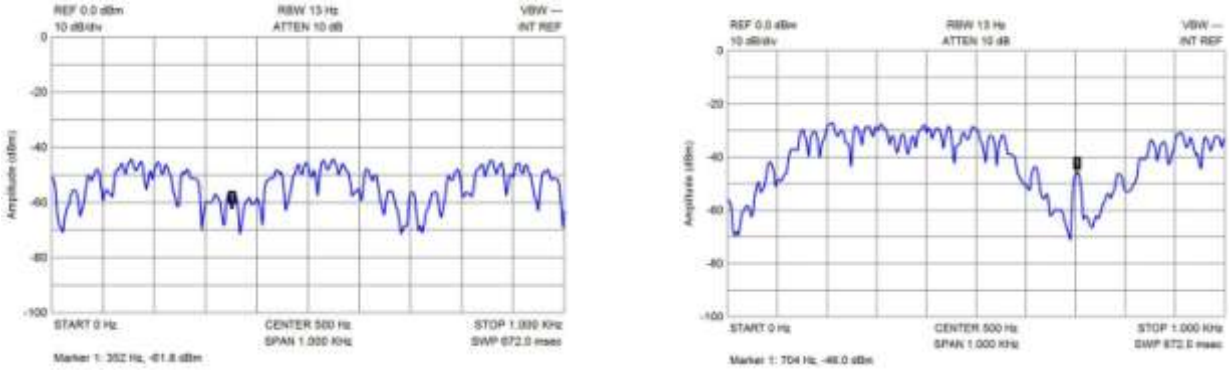


Fig 4.21 Baseband Digital Sigcap. NRZ & Man Random Data Spectral Displays

In **Fig 4.20** The NRZ data is shown on the bottom trace and the Manchester data is shown on the upper trace. For an NRZ “1" at 5 volts, the Manchester data goes from +12 volts to - 12 volts. For an NRZ “0" at 0 volts, the Manchester data goes from -12 volts to + 12 volts. In **Fig 4.21** The NRZ spectrum has a data null at approx. 352Hz and the Manchester spectrum null is at 704Hz or twice this amount. Note that these nulls represent the bit rate and twice the bit rate.

-Bit Time = 2.8msec (**Fig 4.20**)
-Data Rate = 1/Bit Time = 1/2.8msec = 357bps

Note that there is no DC component on the NRZ data. The spectrum analyzer cannot support DC voltage on the input, so there is a DC blocking capacitor inserted for instrument protection.

4.8 Baseband Analog & Digital - Exercises

1.
The spectrum of an unknown signal contains a spectral line at 32.768KHz? What can we say about this signal?

2.
An unknown signal has a totally noise like spectrum. What can we say about this signal?

3.
Construct a signal consisting of the following components:

V1(t)=4sin(ω1t)	F1=1KHz
V2(t)=2sin(ω2t)	F2=8KHz
V3(t)=5sin(ω3t)	F3=15KHz
V(t)=V1(t)+V2(t)+V3(t)	ωn=2*pi*Fn

Plot the time and frequency displays. How would you classify this signal?

4.
Using Windows Sound Recorder or a similar program, record approx 4 seconds of your voice at a sampling rate of 44.1KHz. Modify the ScicosLab files to plot your voice in the time and frequency domain (Adjust either .sce/.cos files to accommodate different sampling rate of 44.1KHz vs 22.05KHz). Where is your voice spectrum most powerful? For your voice display, what proportion of the 4 seconds is the voltage approx = 0 volts?

5.
A random data stream of NRZ data is produced by a microcontroller. The ‘1' state is represented by 3.3 VDC and the ‘0' state by 0 VDC. What is the average voltage of the waveform or the DC value?

6.
An NRZ data stream has a bit rate of 1.544Mbps. Describe the waveform in the frequency domain.

7.
Why does Manchester coded data have twice the bandwidth of NRZ?

8.
What particular advantage does Manchester coding have over NRZ?