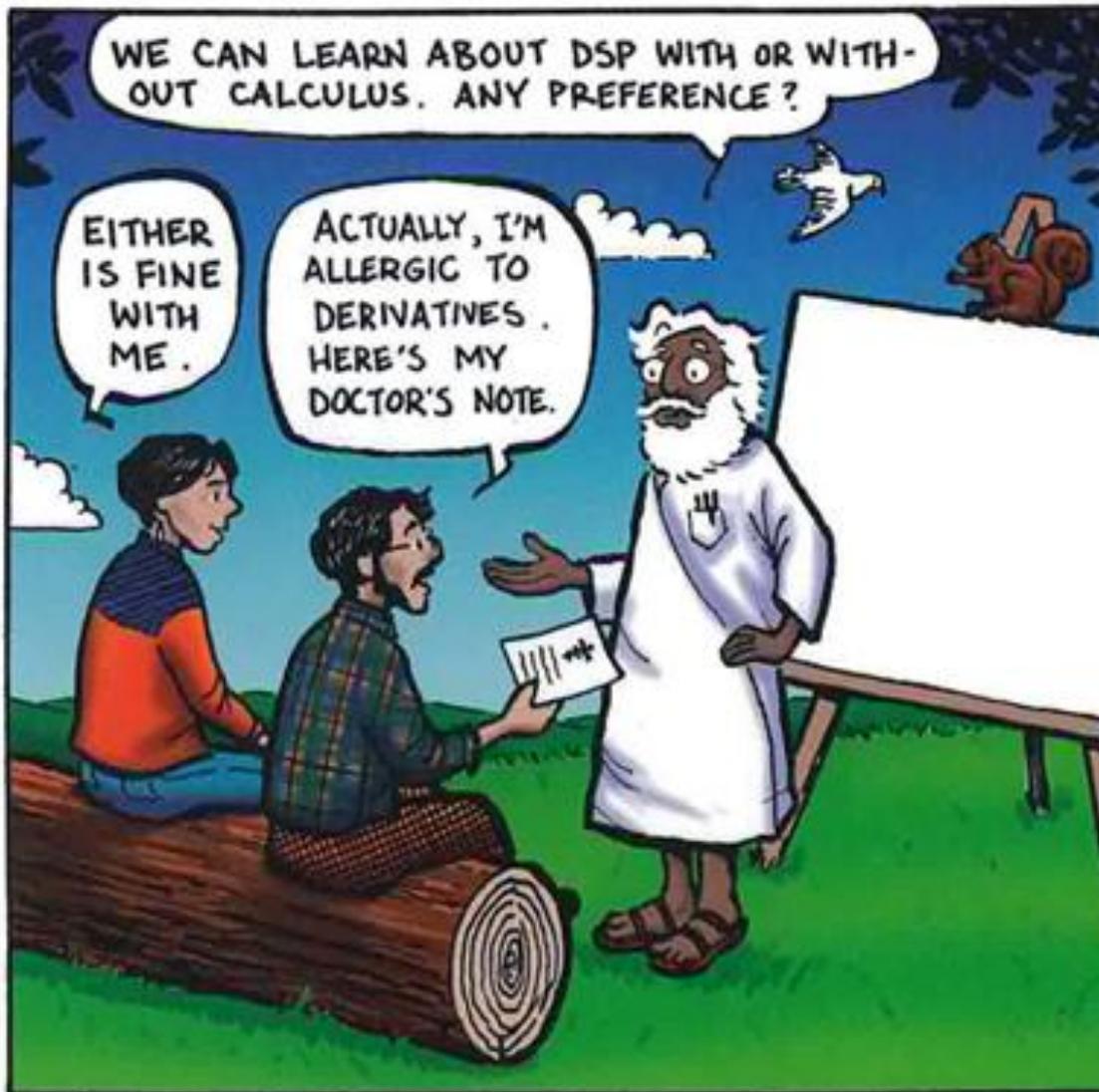




# The 29<sup>th</sup> Annual ARRL and TAPR Digital Communications Conference

## DSP Short Course **Session 1: DSP Intro and Basics**

Rick Muething, KN6KB/AAA9WK



**Dale Grover & John R. Deller**

Illustrations by Jonathan Roth

# Session 1 Overview

- What is DSP?
- Why is DSP better/different than analog?
- What *can* we do with DSP
- What *can't* we do with DSP
- The DSP Cornerstone: The Fourier Transform
- Sampled data systems .... The DFT
- The FFT algorithm to implement the DFT
- Some sample waveforms and their DFT transforms
- Important Transform Properties (caution! Math ahead)
- Floating point or Fixed?
- The NFL Theorem (No Free Lunch)
- Summary

# What is DSP?



- **DSP stands for Digital Signal Processing. The processing of time or frequency signals on a digital computer or dedicated DSP chip.**
- **All useful processing we might want to do to a signal can be expressed mathematically**
- **Doing this with a digital computer takes many thousands of computations...what digital computer are good at!**

# Why is DSP Better/different than analog?



- DSP is done in a computer or dedicated signal processor... the DSP task is program controlled ... more flexibility
- DSP can achieve higher performance. Better filters, reduced noise, better linearity, more agility.
- Analog processing requires active and passive components (Transistors, caps, resistors, ICs, inductors) Component tolerance and drift are issues!
- Digital electronics is advancing at a faster rate than analog. Calculations get Faster, more Precise, Cheaper.
- DSP is our opportunity to get back into “experimenting” with ham radio. (Look at the advancements of projects like the Flex Radio and HPSDR and the increase in “sound card” modes)

# What *can* we do with DSP?

- **Virtually any standard signal processing including:**
  - Building “DSP components” Oscillators, mixers, filters, etc
  - Modulation, demodulation, noise reduction, mixing, detection, averaging, peak detection, AGC etc.
  - Time base adjustments, Doppler correction
  - Propagation analysis and simulation
  - Create “stealthy” signals (Spread spectrum)

# What *can't* we do with DSP?

- Create stealthy signals (for purposes of obscuring)
- Use DSP to exceed basic theoretical limits (e.g. Shannon's limit of channel capacity)
- Use DSP to predict random processes (e.g. You can't truly cancel out random noise)
- You can't make a silk purse from a sow's ear! If your sound card has only 16 bits you can't get 120 dB of dynamic range.
- You are often limited by how much DSP computing power is available. Real time concerns.
- While there is a DSP concept of negative frequency you can't use that to go back to the future!



# The DSP Cornerstone...The Fourier Transform

- Just who was Joe Fourier anyway?
- A French mathematician and physicist 1768-1830 that showed any periodic function could be represented by a sum of harmonically related sine and cosine waves.
- The Fourier Transform

$$\hat{f}(n/T) = c_n = \int_{-T/2}^{T/2} e^{-2\pi i n x/T} f(x) dx$$

[http://en.wikipedia.org/wiki/Fourier\\_transform](http://en.wikipedia.org/wiki/Fourier_transform)

Joseph Fourier



Jean Baptiste Joseph Fourier

# Sampled Data Systems

- Instead of representing signal by continuous waveforms (e.g. sine wave) we represent the signal as a series of samples in time.
- The Nyquist-Shannon sampling theorem states that if a function of time consists of no frequencies above  $B$  Hz then it can be represented by a series of samples spaced at  $1/2B$  seconds apart.

**DSP Heros!**

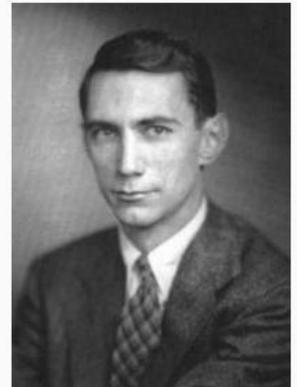
Harry Nyquist



Scanned at the American  
Institute of Physics

Harry Nyquist (1889-1976)

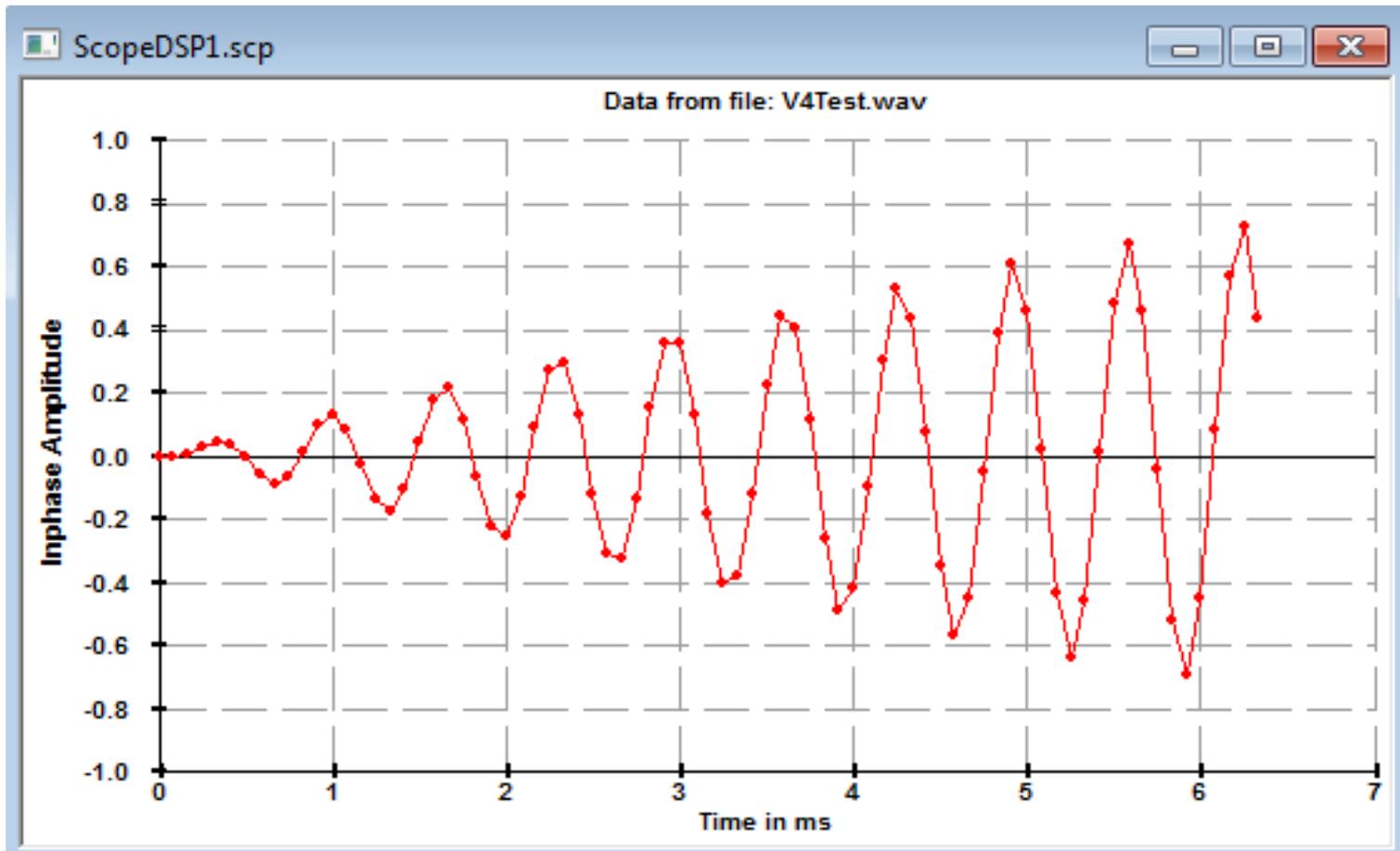
Claude Shannon



Claude Elwood Shannon (1916-2001)

# Example Sampled .WAV File

- This .wav file plot shows the first few samples of a two tone waveform sampled at 12000 samples/sec. The Dots are the actual sample points. The wave form plotter fills in the lines to make it easier to view.



# The DFT (Discrete Fourier Transform)

- The DFT is a method of doing the Fourier transform on the sampled data. Instead of the integrals of continuous functions of the Fourier Transforms it uses sums of products of the discrete samples.
- In DSP we always use the DFT because we are working with sampled data. (The digital computer also does sums of products much easier than integration!)
- One aspect of the DFT that can be confusing: The DFT assumes the sampled data transformed is basically a single occurrence with an infinitely long period...more on that when we talk about windows in session 4 (no not Bill Gates' Windows)



# So then what is an FFT?

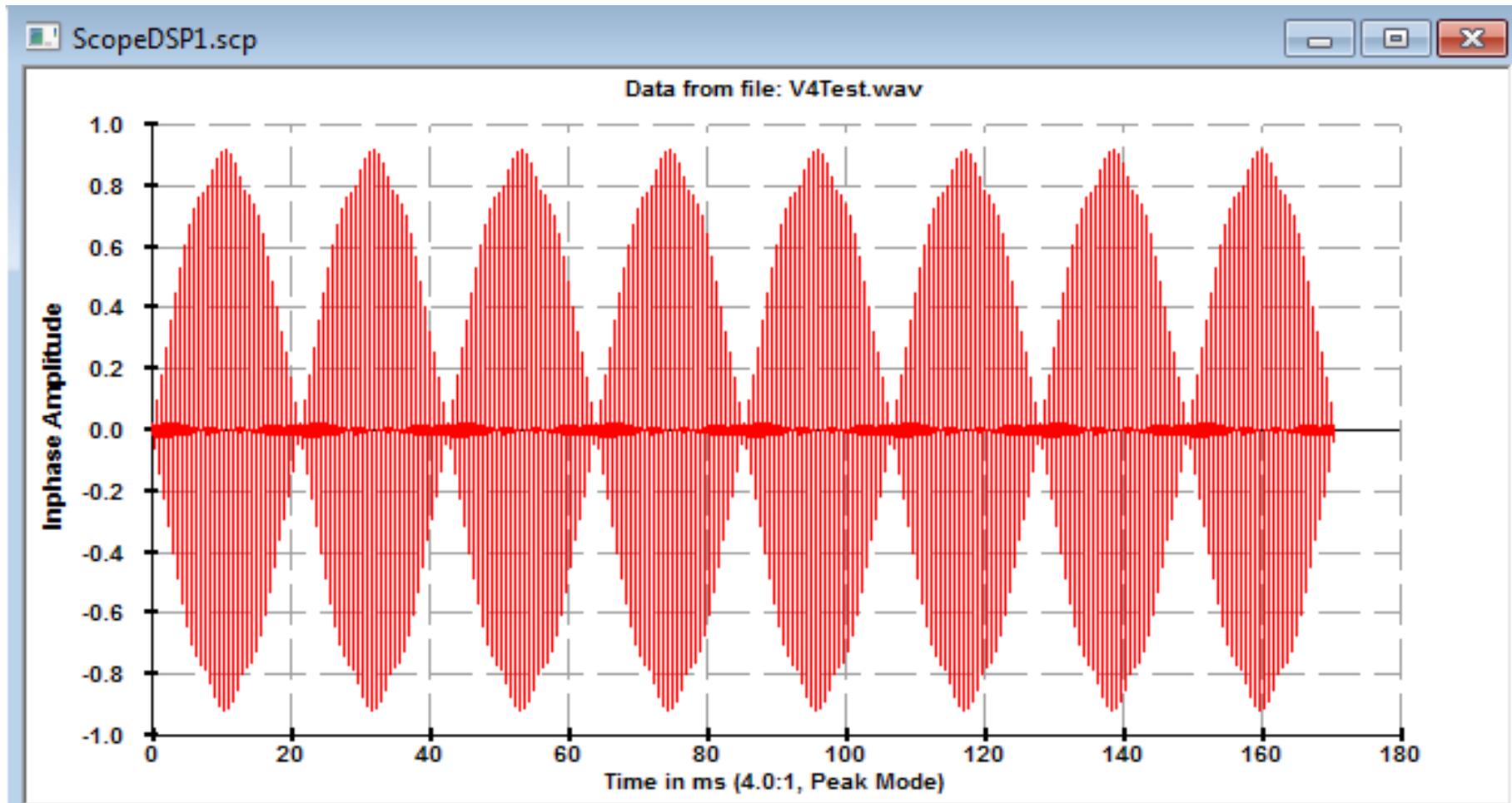


- The FFT (Fast Fourier Transform) is simply an efficient algorithm (Cooley-Tukey) to implement the DFT on a digital computer. It minimizes the number of multiplies and adds needed to do the transform of the sampled data.
- For the FFT the # of samples must be a power of 2
- Even with the FFT it can take a lot of calculations! About  $\log_2(N) * N$  multiply + adds where N is the transform size
- For a 1024 point transform the FFT takes over 10,000 multiply + add operations... You see why DSP chips are characterized in MIPS (millions of instructions/sec)?
- There are other algorithms (e.g. Goertzel) which can be more efficient in certain circumstances...we'll look at those in the last session

[http://en.wikipedia.org/wiki/Butterfly\\_diagram](http://en.wikipedia.org/wiki/Butterfly_diagram)

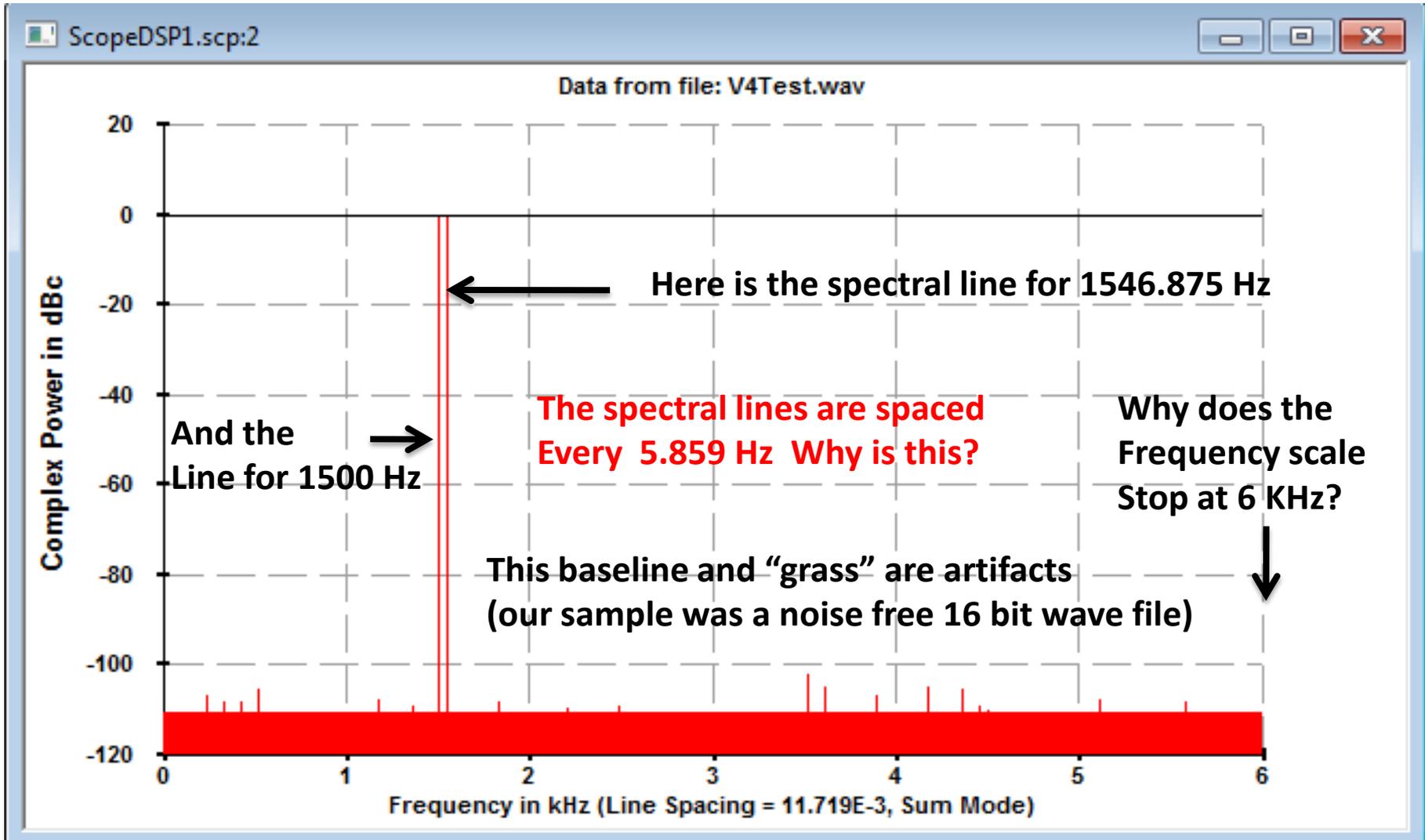
# A sample Waveform and its DFT

Here are 2048 samples of a two tone waveform of 1500 Hz + 1547Hz.  
The sample rate is 12000/sec and the sample time is ~170 ms



# FFT of the Two tone sample

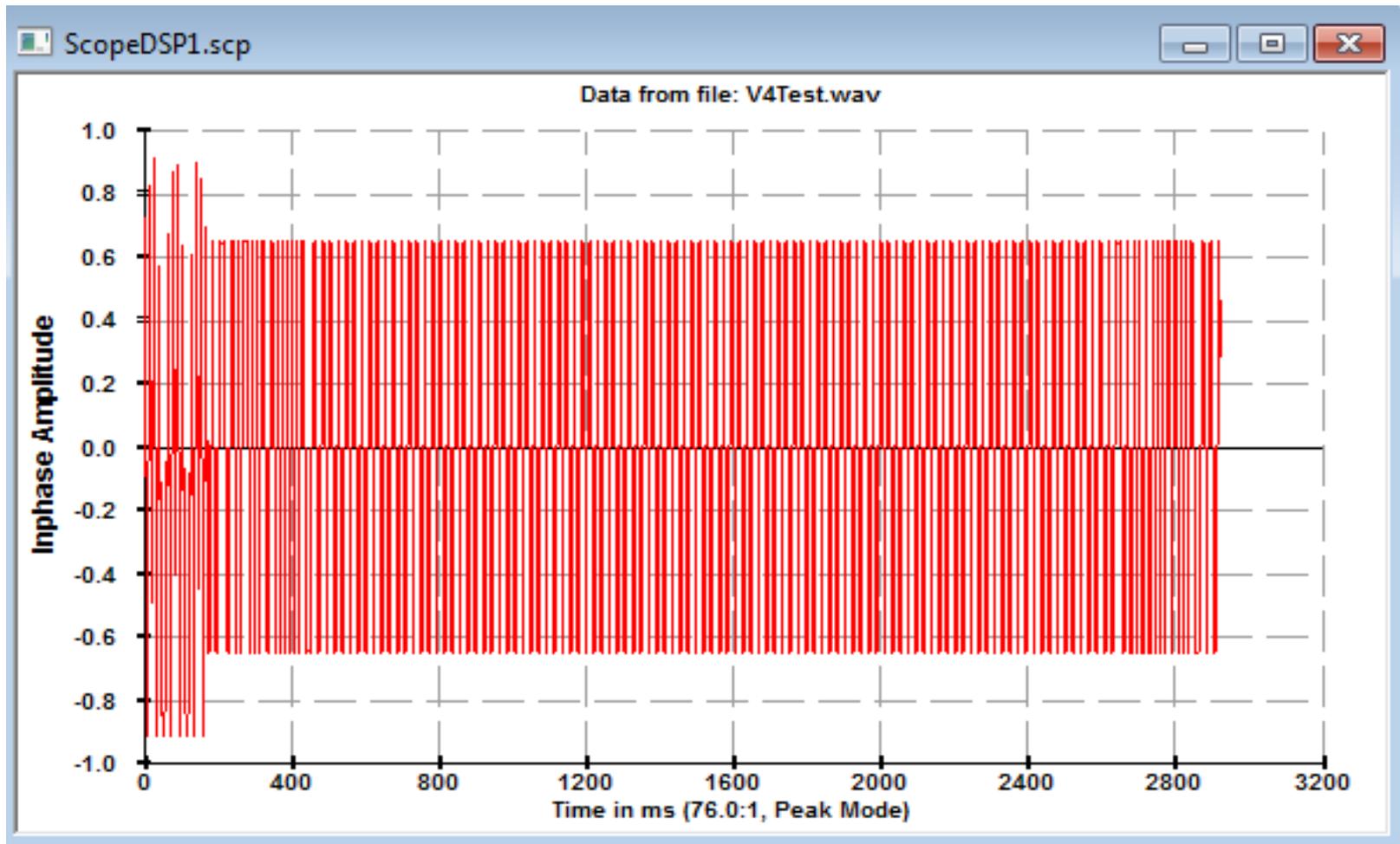
This is the plot of frequency magnitudes of the FFT of the prior wave form.



# A More complex sample

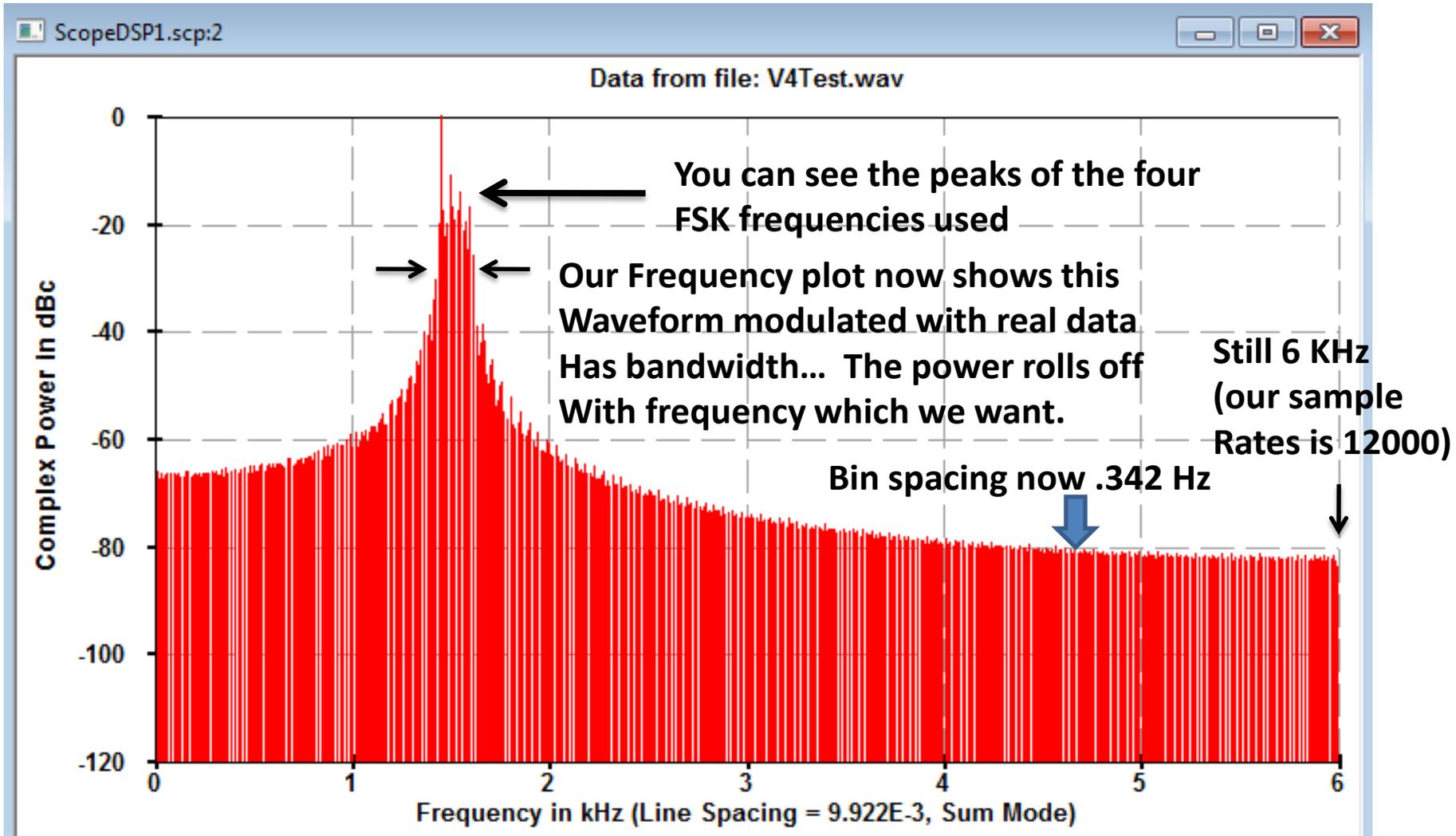


This is about 35000 samples at 12000 Hz sample rate. The waveform is a full Frame (almost 3 seconds) of 4FSK data following a two Tone leader.



# And Here is the FFT

This is the plot of frequency magnitudes of the FFT of the prior wave form.



# FFT and DFT Transform Properties

- **A simplified description:**
  - The Transform is Linear (the transform of  $A + B$ ) = transform of  $A$  + the Transform of  $B$
  - The Transform is invertible. We can go from time to frequency and back and we have what we started with.

For the full and painful description of the DFT properties go to:

[http://en.wikipedia.org/wiki/Discrete\\_Fourier\\_transform](http://en.wikipedia.org/wiki/Discrete_Fourier_transform)

# Would You like that DFT



## Floating or Fixed?



- In the earlier days Floating point was slow and expensive (few floating DSP chips).
- Fixed point offers possible speed improvements and reduction in hardware and power....BUT!
- Keeping track of the scaling with Fixed point is a major headache...With floating point you can forget about scaling, over/underflow, etc.
- Today's High end MPU chips do 64 bit floating point as fast or faster than some fixed point.
- Recommend starting with floating point and optimize for speed/fixed point *only* if needed.

# NFL Theorem



- NFL stands for No Free Lunch!
- All this DSP stuff and capability comes with a price...we can't get something for nothing
- We've had a brief exposure to the NFL Theorem with frequency limitations and bin spacing examples of our sample DFTs. *If you want wide frequency range...sample fast. If you need fine frequency resolution sample long.*
- Just remember in DSP as in life there are always tradeoffs... And No Free Lunches

# Session 1 Summary

- We've defined DSP and looked at what we can and can't do with it.
- We have investigated the Fourier Transform, the DFT and the FFT
- Homage has been paid to Misters Fourier, Nyquist and Shannon for their significant contributions.
- We have looked at sample wave forms and their DFTs ... and reminded ourselves of the limitations of the DFT.
- We briefly looked at the more important properties of the DFT and Fixed and Floating point issues.
- We know there is No Free Lunch when it comes to DSP
- In our next session we'll dig into some of the tools we'll need to understand and do DSP.

