The 29th Annual ARRL and TAPR Digital Communications Conference

DSP Short Course

Session 4: Tricks of the DSP trade

Rick Muething, KN6KB/AAA9WK

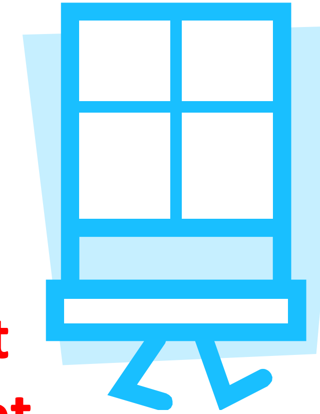
Recap

- We’ve surveyed the roots of DSP and some of the tools we’ll need to work with DSP.
- Some of the more useful DSP “components” have been defined. We’ve looked at some examples of combining these in a DSP processing diagram to do something useful!
- This final session with cover a few of the finer points of DSP and some of the “DSP tricks of the trade” that make solutions practical.

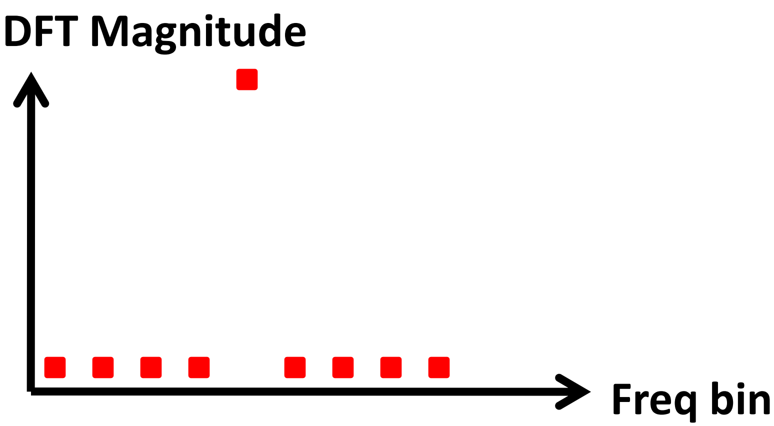
Session 4 Overview

- DFT “leakage” and Windowing
- Bin interpolation for better Frequency Resolution
- I Q sampling and how to obtain it
  - Hilbert Transform
- Single Tone Detection/Single bin DFT
- Sample Rate Conversion
  - Decimation and Interpolation
- Envelope approximation
- Sliding DFT
- FM Detection
- Your next Steps in DSP!

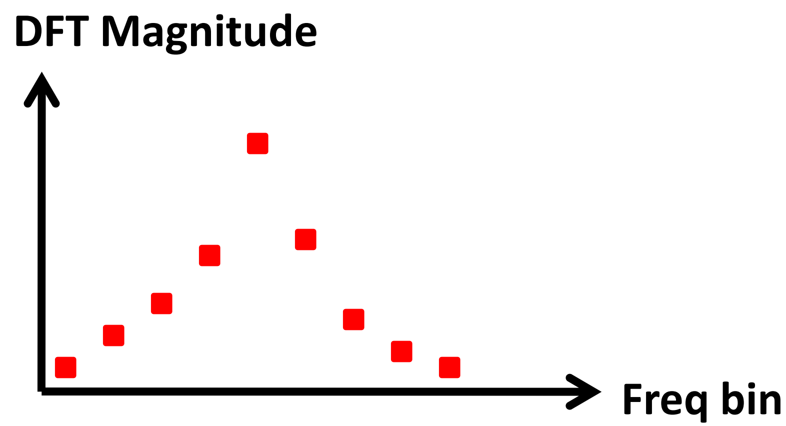
“DFT “Leakage” and Windowing



When the frequency components of the input signal don’t Lie exactly on the center of the DFT frequency “bin” we get An approximation of the true spectra because of a mechanism we call “bin leakage”



Resulting DFT when Input Frequency = the center of a frequency bin

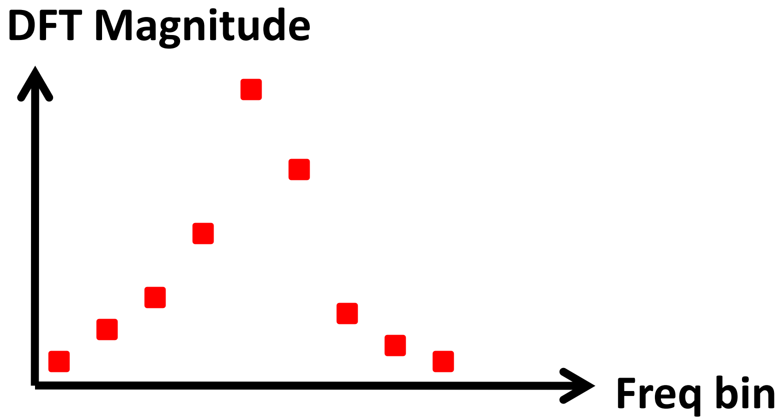


Resulting DFT when Input Frequency is NOT in the center of a frequency bin

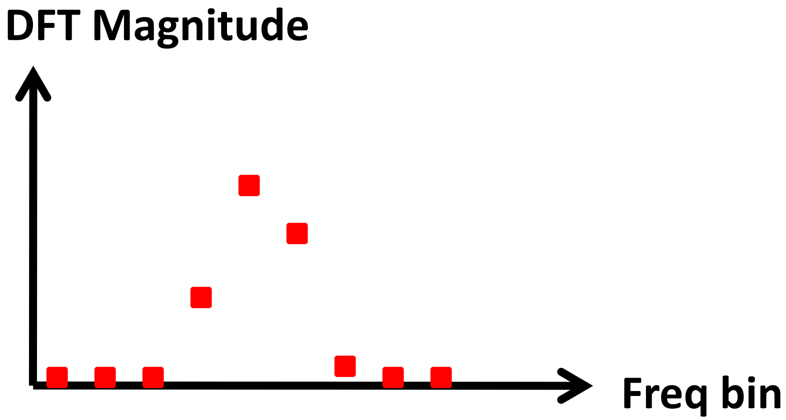
"DFT "Leakage" and Windowing

We can't eliminate leakage but we can minimize the effects of leakage Using a technique called "windowing"

We window (scale) the input sequence with a function that reduces the magnitude of the samples near the beginning and end of the sample sequence.
e.g. Hamming Window = .54 - .46(Cos(2*Pi*n/(N-1)))



Without Windowing



With Windowing

“DFT “Leakage” and Windowing

- But “windowing” is a tradeoff (remember the NFL theorem?). Windowing will reduce the magnitude of the side lobe leakage but it also spreads the main lobe (worsens frequency resolution)
- Depending on how the DFT output is used may require different window functions or no windowing at all.
- One of the best ways to get a feel for DFT Windowing is to use the Scope DSP utility, apply different window functions to the same waveform and observe the resulting spectrum.

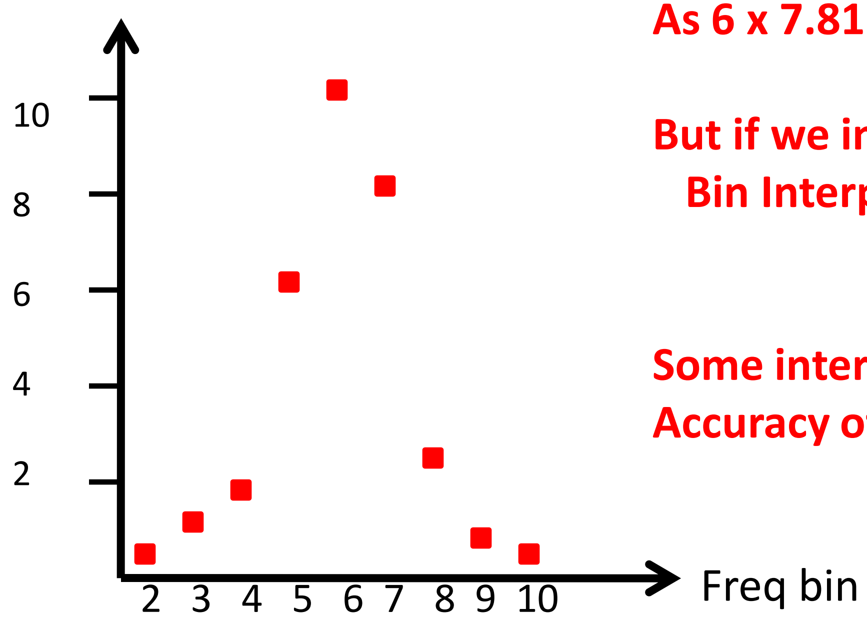
http://en.wikipedia.org/wiki/Spectral_leakage

Bin interpolation for better Frequency Resolution

What if we need better frequency resolution than the DFT Frequency bins?

We know the DFT bins are spaced at 1/(total sample time) e.g.

Sample rate of 8000, 1024Point DFT yields bin spacing of 7.8125 Hz



Our Max bin = 6 so we estimate our frequency
As 6 x 7.8125 or 46.87 Hz

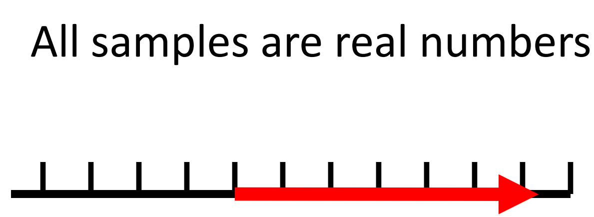
But if we interpolate with the two adjacent bins:
Bin Interp = 5*6 + 6*10 + 7*8 /(6 + 10 + 8)
= 6.08 or 47.53 Hz

Some interpolation algorithms yield good
Accuracy of .1 bin or better!

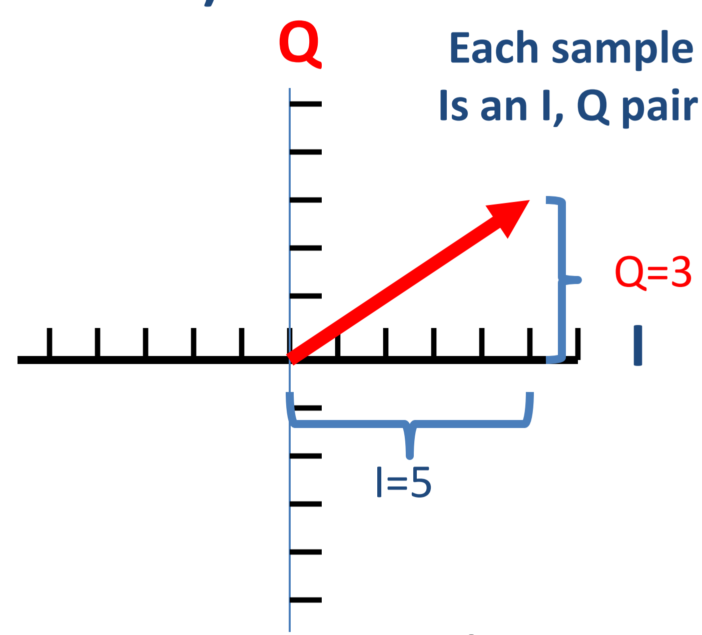
Bin interpolation is a very effective tool when we can't increase the sample time

I Q Sampling …the Holy Grail of DSP

- **We've touched on it but what *is* I Q sampling?**
- **I stands for "In Phase" and Q stands for "Quadrature"**
- **We augment the common "real" samples with samples 90 degrees delayed ("imaginary" samples").**
- **This allows computing many useful functions in DSP. (balanced mixers, envelope detectors etc)**



Real samples only



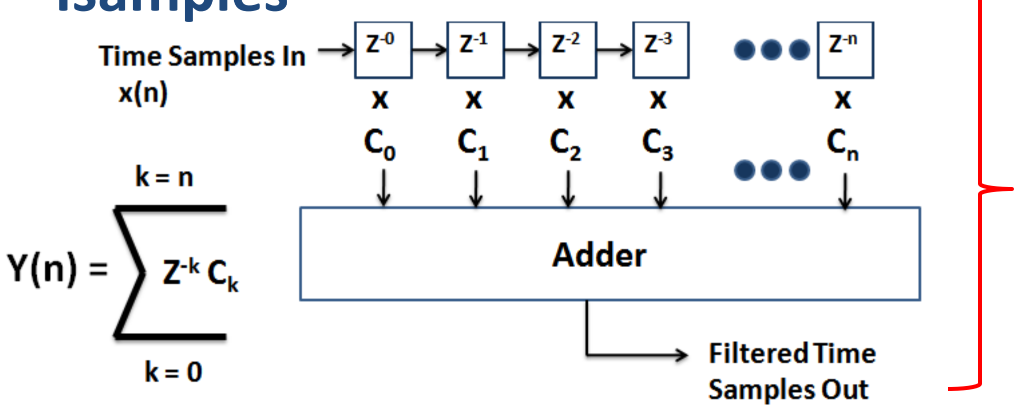
I and Q **samples (real, imaginary)**

How do we get I Q Samples?

- Here are two common methods
  - Direct IQ sampling using sample points 90degrees out of phase. (Might be a good fit for stereo sound cards)
  - Generation from real only samples using a Hilbert Transform. Implemented like a FIR filter but with specific Hilbert Transform coefficients.

http://en.wikipedia.org/wiki/Hilbert_transform

Isamples



Here again is our Familiar FIR filter Structure

Qsamples (delayed)

Single Tone detection, Single bin DFT

- **The Goertzel algorithm can implement a single bin DFT or single bin Tone detector using a 2nd order IIR structure.** http://en.wikipedia.org/wiki/Goertzel_algorithm
  - more computationally efficient than the DFT if the number of bins (tones) is limited.
  - N does NOT have to be a power of two!
  - can calculate fractional bin #'s (e.g. bin 31.22)
  - Can be computed as received (no large arrays)
  - E.g. Actual execution time:
    - DFT/Goertzel = 3.3 $\text{Log}_2(N)$

The Goertzel subroutines are on the CD!

This was the "trick" used in the RTTY decoder:

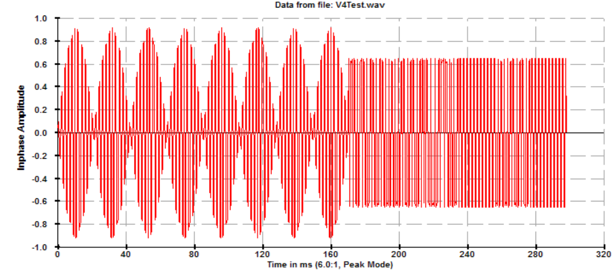
N= 1055 (allowed "fitting" symbol size with sample rate) allows placing bins for optimal orthogonal detection

Sample Rate Conversion

- Needed when streams of different sample rates must be combined or processed (e.g. As in Virtual Audio Cables)
- Can reduce processing demands
- To decrease the rate use Decimation $F_{new} = F_{old}/D$
- To Increase the rate use Interpolation $F_{new} = F_{old} * M$
- Can use combination to translate to any M/D value
- Often the rate conversion must use a FIR low pass or Cascaded Integrator-Comb IIR filters

http://en.wikipedia.org/wiki/Sample_rate_conversion

Envelope Approximation



We often need the envelope of the waveform .

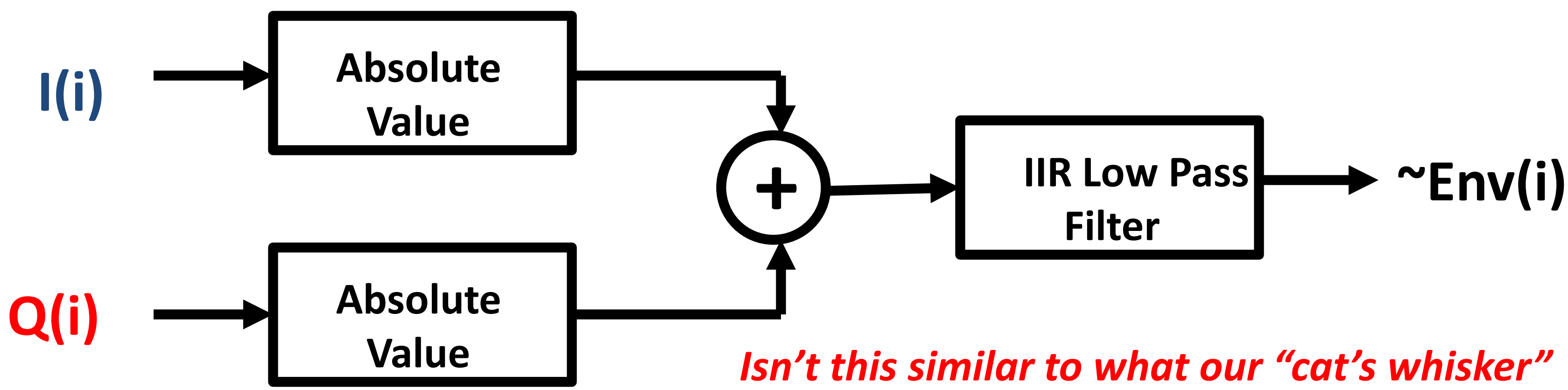
(Recall our example of the DSP Crystal Radio)

The envelope of a sample waveform is : Env(i) = Sqrt(I(i)^2 + Q(i)^2)

This can be a heavy computational load at times.

There are a number of approximations that come close to the above Envelope at much lower computational loads. Often these are adequate For envelope detection:

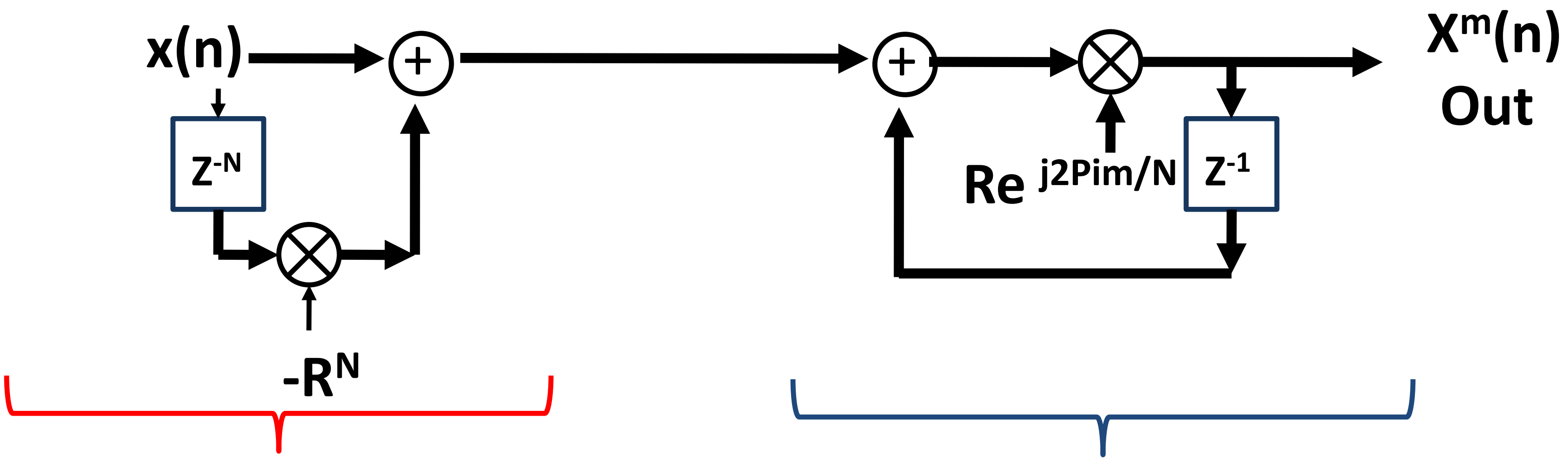
E.g. (simple absolute value approximation uses no multiplies or square root)



Isn't this similar to what our "cat's whisker" diode and cap did in our Crystal Radio?

The Sliding DFT

- The Sliding DFT is useful when we require a new DFT output spectrum every sample or few samples.
- Computationally lower cost than the DFT or the Goertzel
- Uses an IIR structure so we must insure it is long term stable! (R must be close to but < 1 e.g. .9999)

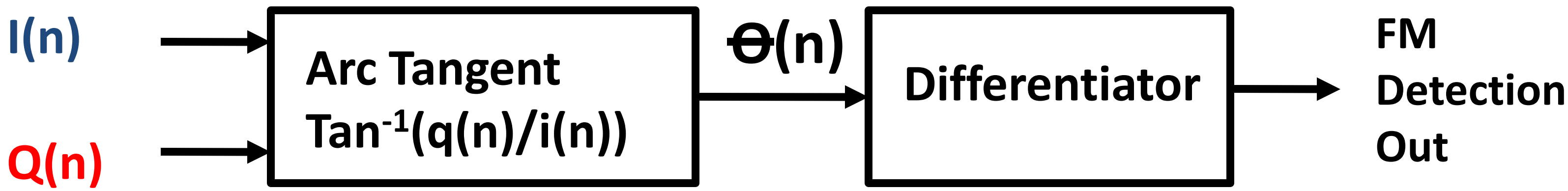


COMB section

One or more complex resonator sections
(One for each output bin)

FM Detection

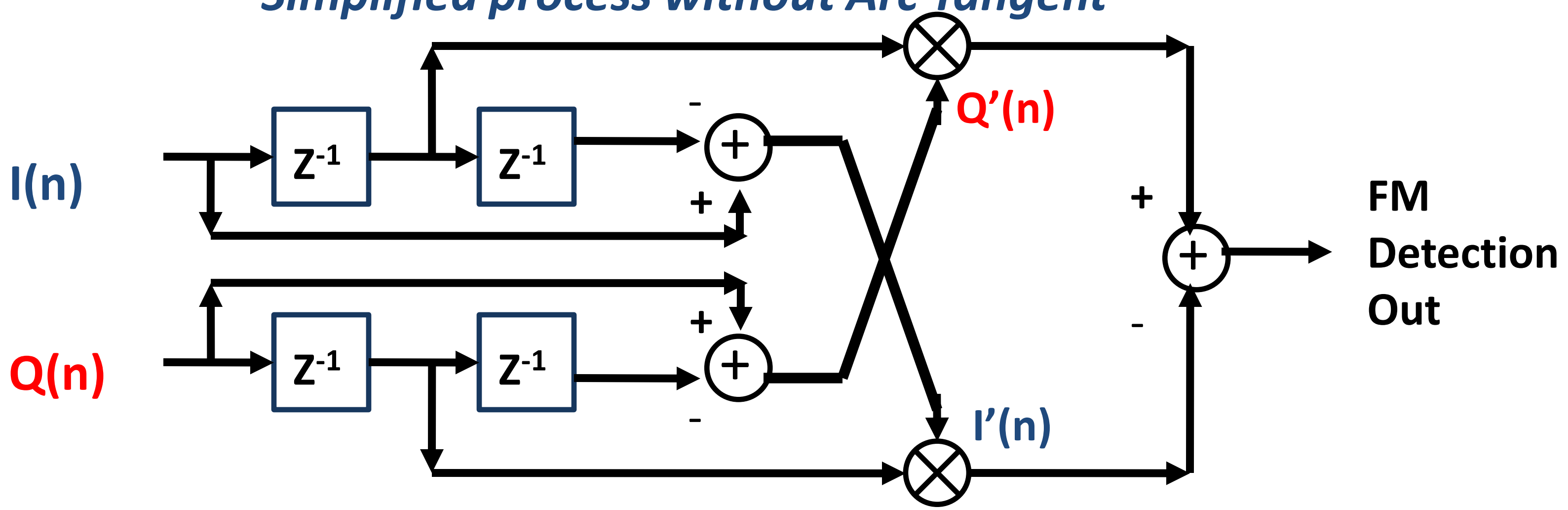
- FM detection involves getting the instantaneous frequency.
- We note that Frequency is rate of change (derivative) of Phase



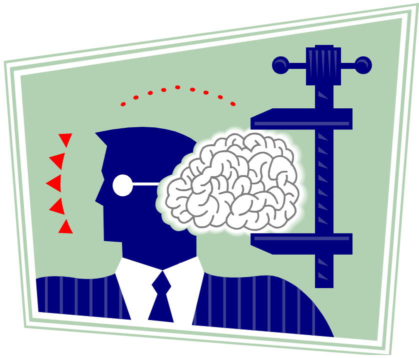
Mathematically straightforward but computationally demanding

(After some calculus identities and simplification)

Simplified process without Arc Tangent



DFT Short Course Summary



- We've packed a lot into a 4 session introduction!
- Session 1 covered the basics of DSP and the important pioneers and basic algorithms
- Session 2 looked at tools and the most common types of filter building blocks
- Session 3 introduced other DSP "components" and how we tie them together to perform some useful task
- Session 4 looked in more depth at the concepts of bin leakage and I Q sampling and some of the more common "tricks of the DSP trade"
- The next step is yours to practice and build some basic applications to improve your understanding and skills.

What's **YOUR** next step on the DSP Road?

- **Some Suggestions:**
  - **Get a good DSP reference (Lyons gets my vote!)**
  - **Write some simple programs to generate**
    Simple waveforms.
  - **Use the Scope DSP utility to view the waveforms in time, Frequency and phase.**
  - **Use Scope DSP to "window" your waveform and observe the affects of the various windows.**
  - **Design a simple IIR or FIR filter using the Scope filter tools**
  - **Try and implement some simple but useful sound card application**

DSP Short Course

- Thank you for your interest and attention
- Dig into more of the details of DSP on your own
- Follow the words in the very first book on DSP
  - Genesis 1:28

Be fruitful and Multiply (and Add)!