Real-Time Universal Transfer Function Synthesizer

Izabella Pastrana

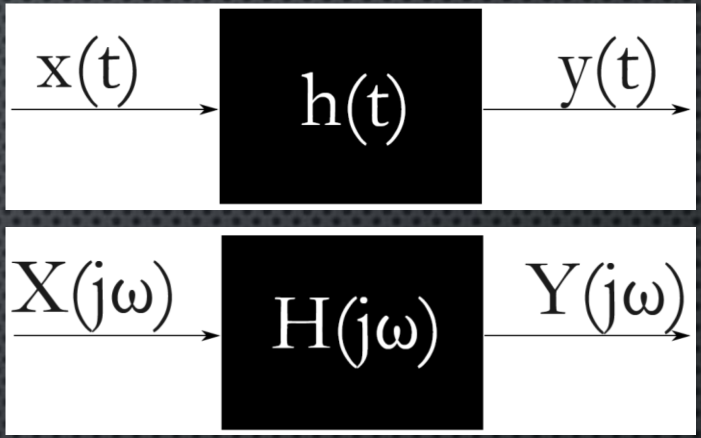
Washington University in St. Louis

Advisors: Johannes Eichholz, Rana Adhikari, Christopher Wipf

Project Outline

- Measure transfer functions with high accuracy
- Complex modeling and fitting
- Implementation in embedded system (FPGA)

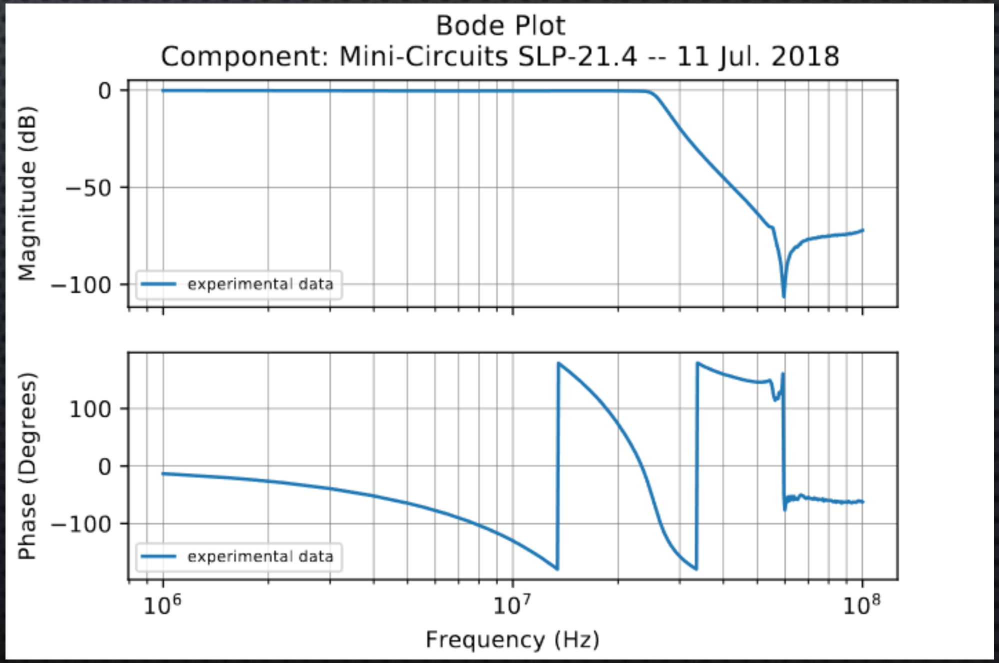
Transfer Functions



- Laplace transform of the impulse response of a linear-time invariant (LTI) system when initial conditions are set to zero
    - Relates output to input in Laplace/Fourier domain
    - H(s) = Y(s) / X(s)

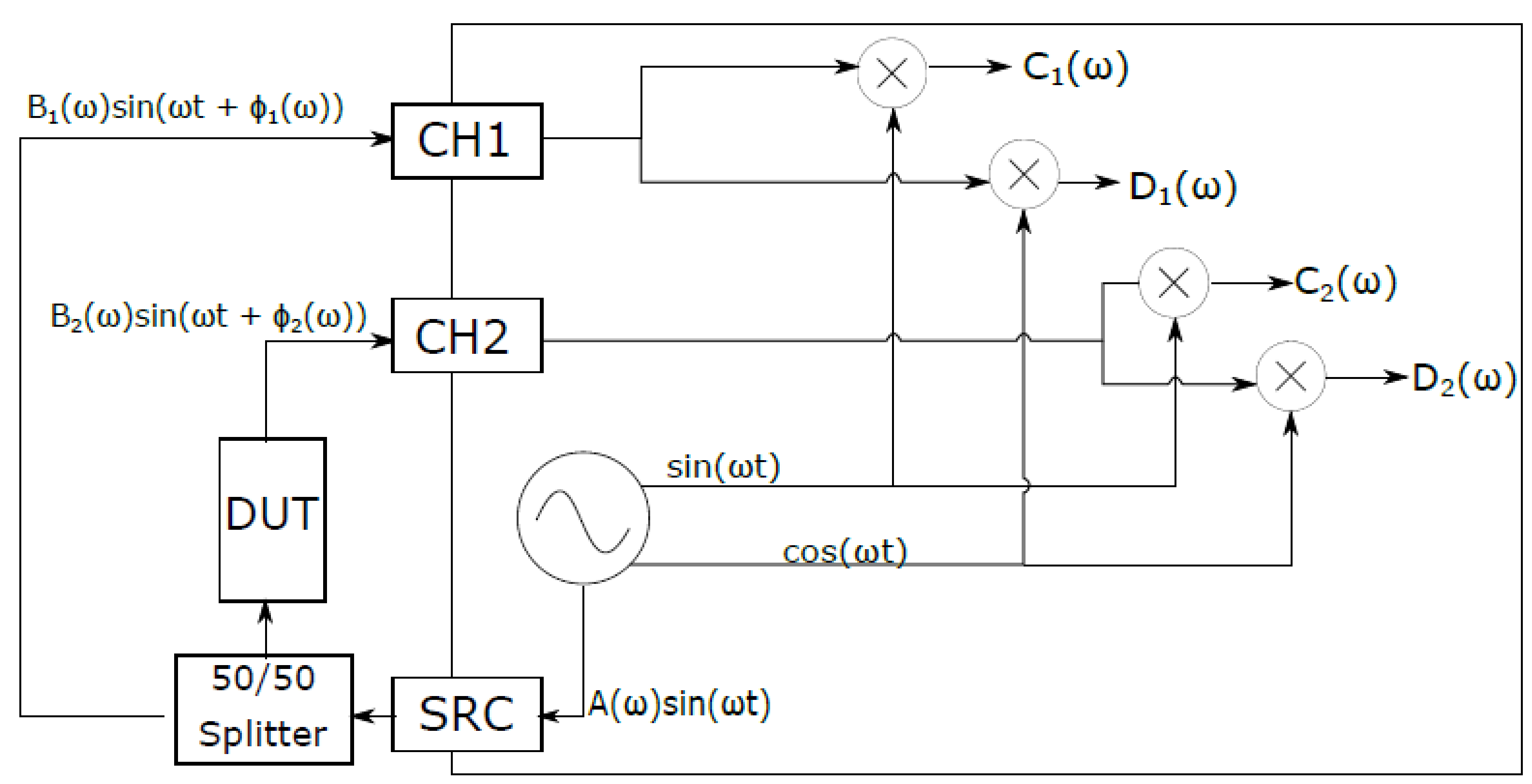
$$H(s) = K\frac{(s-z_1)(s-z_2)...(s-z_{n-1})(s-z_n)}{(s-p_1)(s-p_2)...(s-p_{n-1})(s-p_n)}$$

$$s = \sigma \pm j\omega \longrightarrow s = j\omega$$



- Bode plots – show gain and phase response of system
    - Plot magnitude (dB) and phase (degrees) against logarithmic frequency

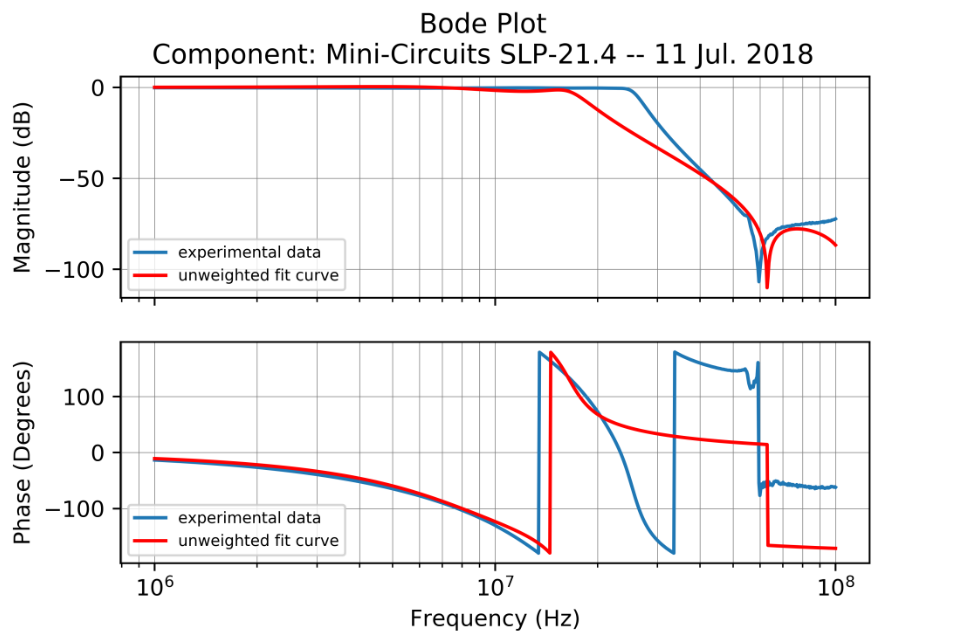
Measuring TFs

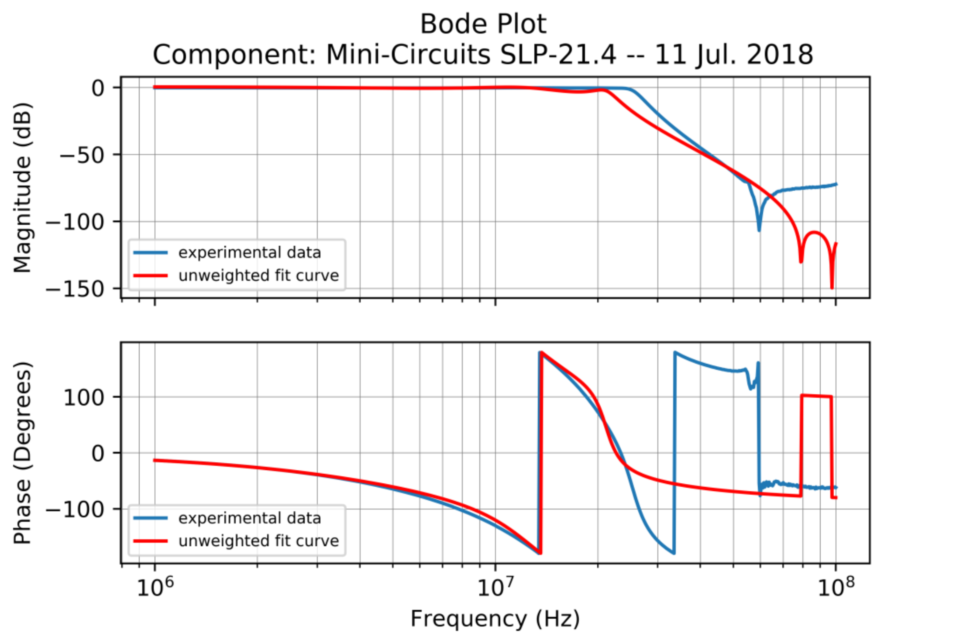


Agilent 4395A Vector Network Analyzer

Modeling TFs

- TF fitting to pole-zero model of a 4^{th} order elliptic low-pass filter
  - Fit model parameters by minimizing root mean square error between model and data
  - More interested in fitting to passband and transition region than stopband



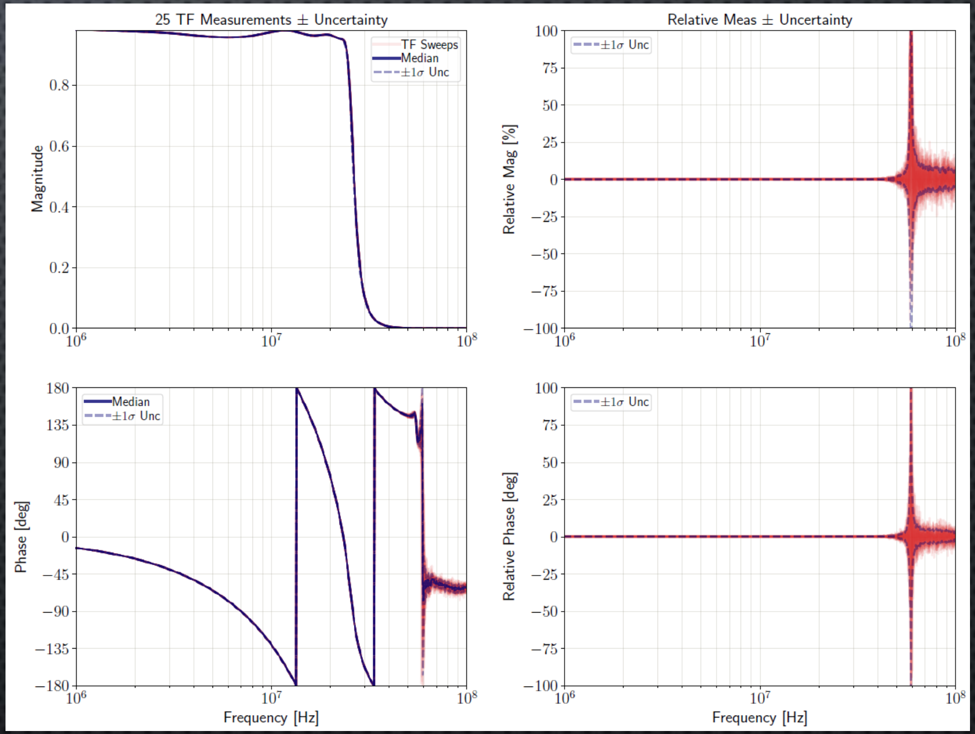


Modeling TFs

- Tradeoff: accuracy v. time
  - Feature-rich data
    - Problem: under-sampling
  - Noisy data
    - Problem: trust

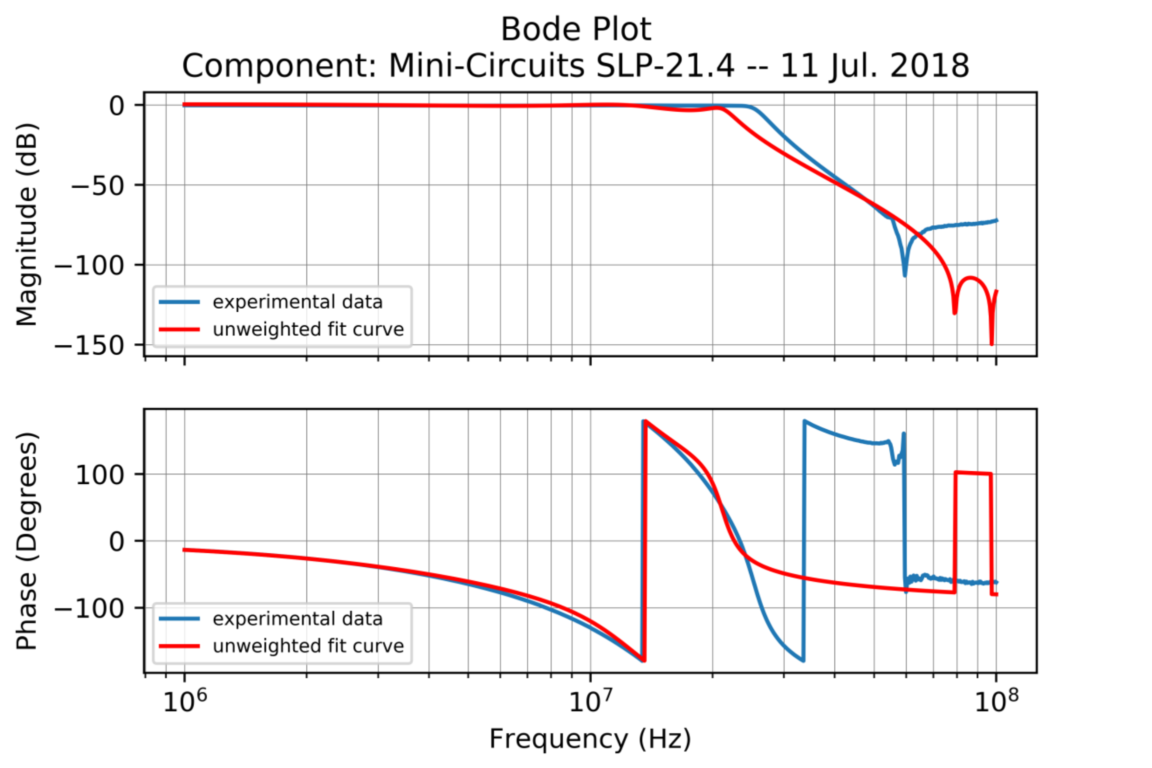
- Uncertainties!

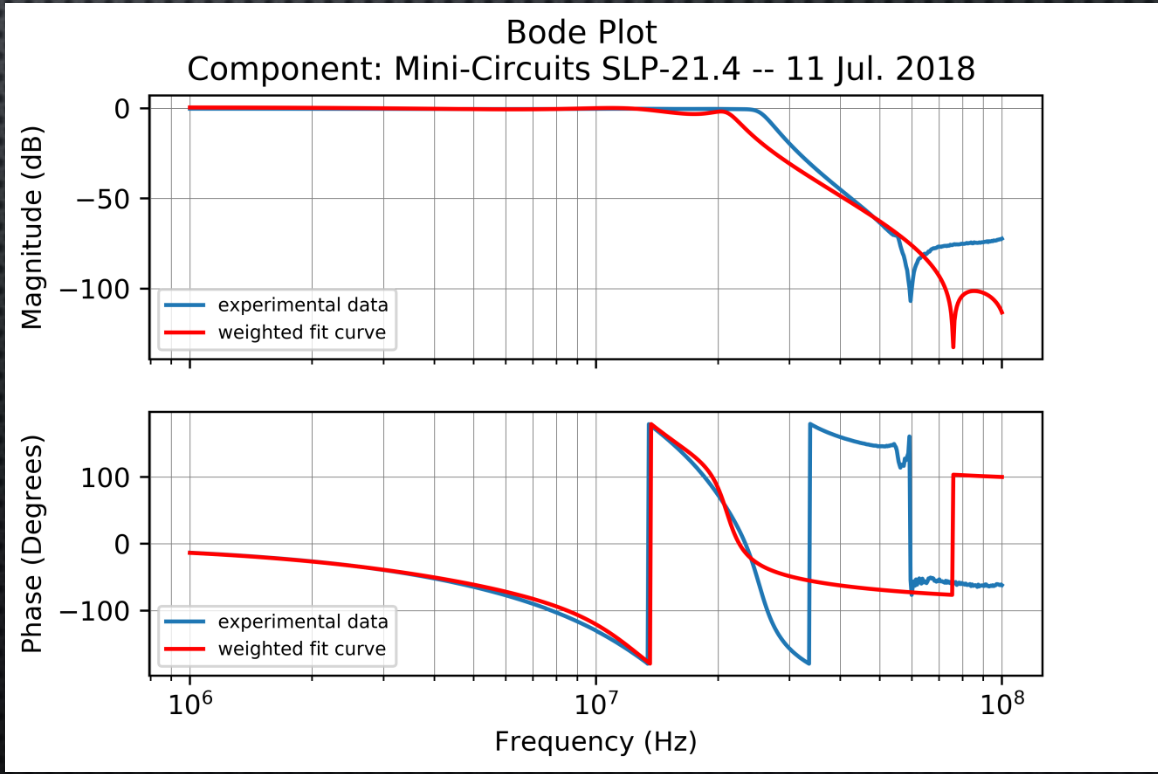
- IRIS software – Craig Cahillane (https://git.ligo.org/40m/labutils/tree/master/iris)
  - Visualization tool
  - Plots individual sweeps and spread in values



Modeling TFs

- TF fitting to pole-zero model – now with weights!
  - Modified IRIS to write out median complex TF and uncertainty measurements for use in fitting process





ADAPTIVE SWEEPING

- Some network analyzers have auto-resolution features
  - Hard programmed into device
  - Could emulate in software, but data transfer rate too slow
- Developing own auto-resolution algorithm
  - Euclidean distance – measuring in higher resolution

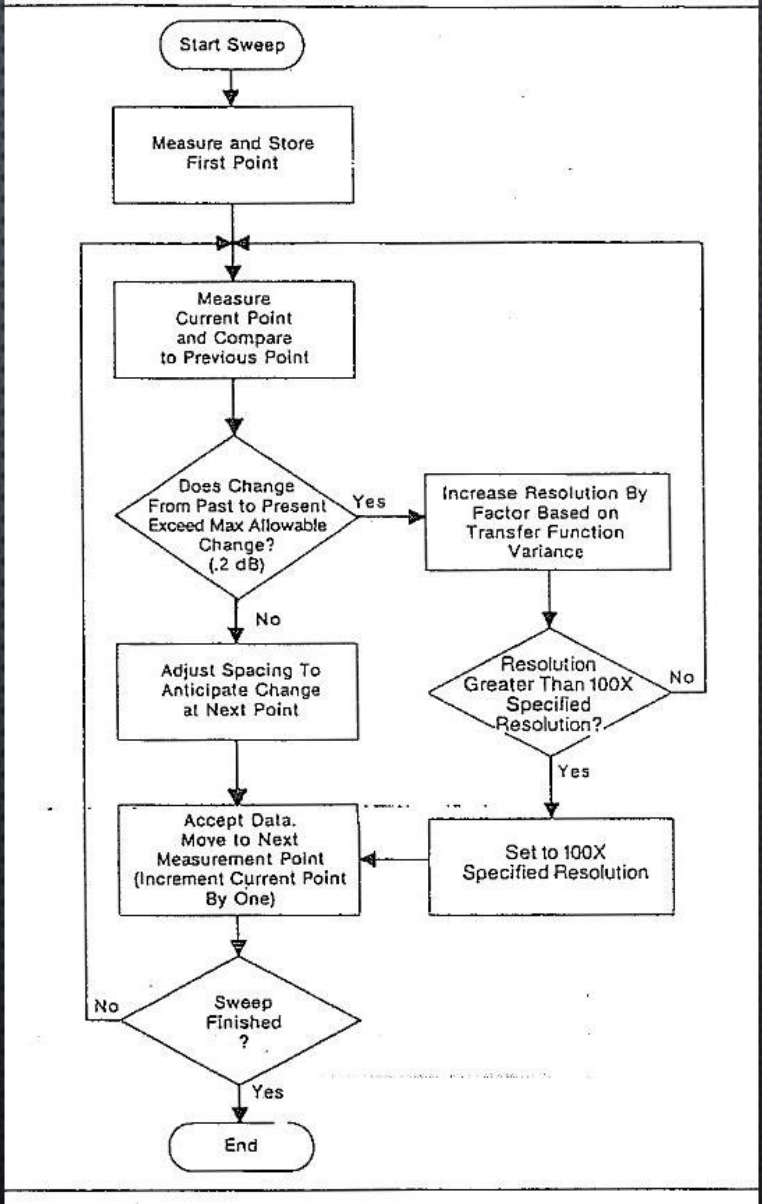
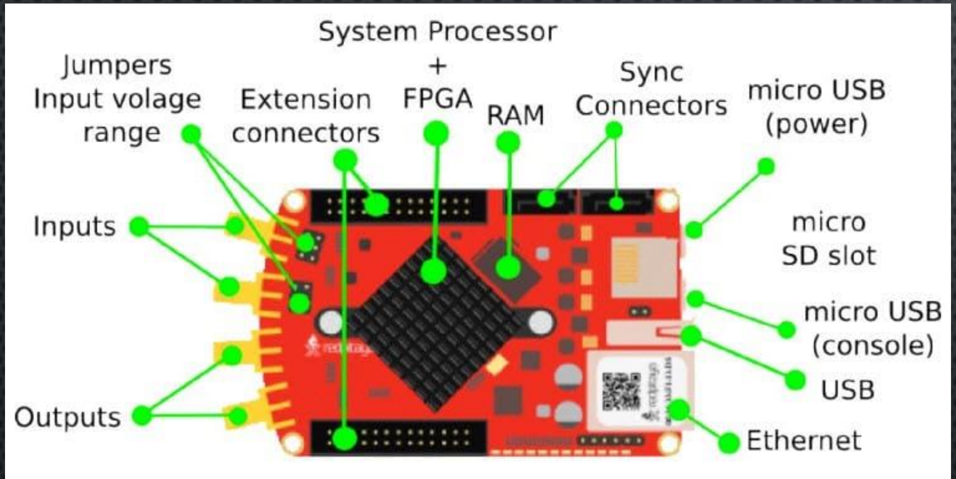


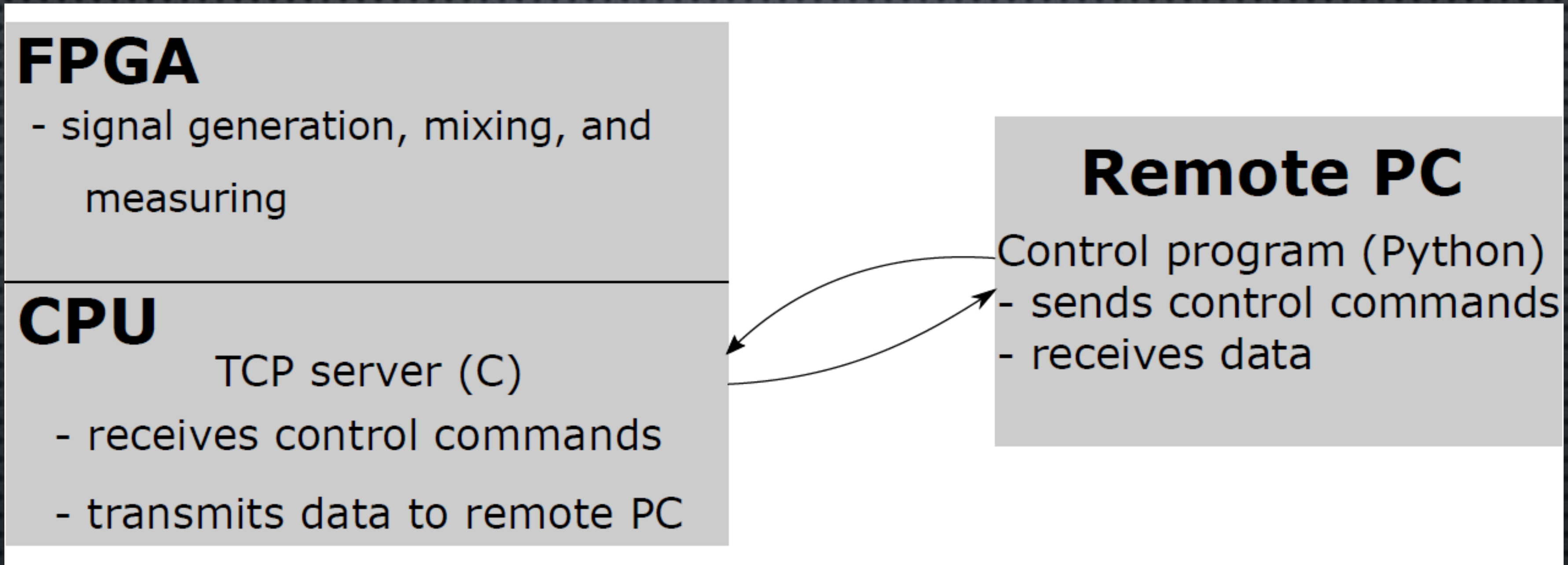
Figure 3-3 Auto Resolution

The redpitaya



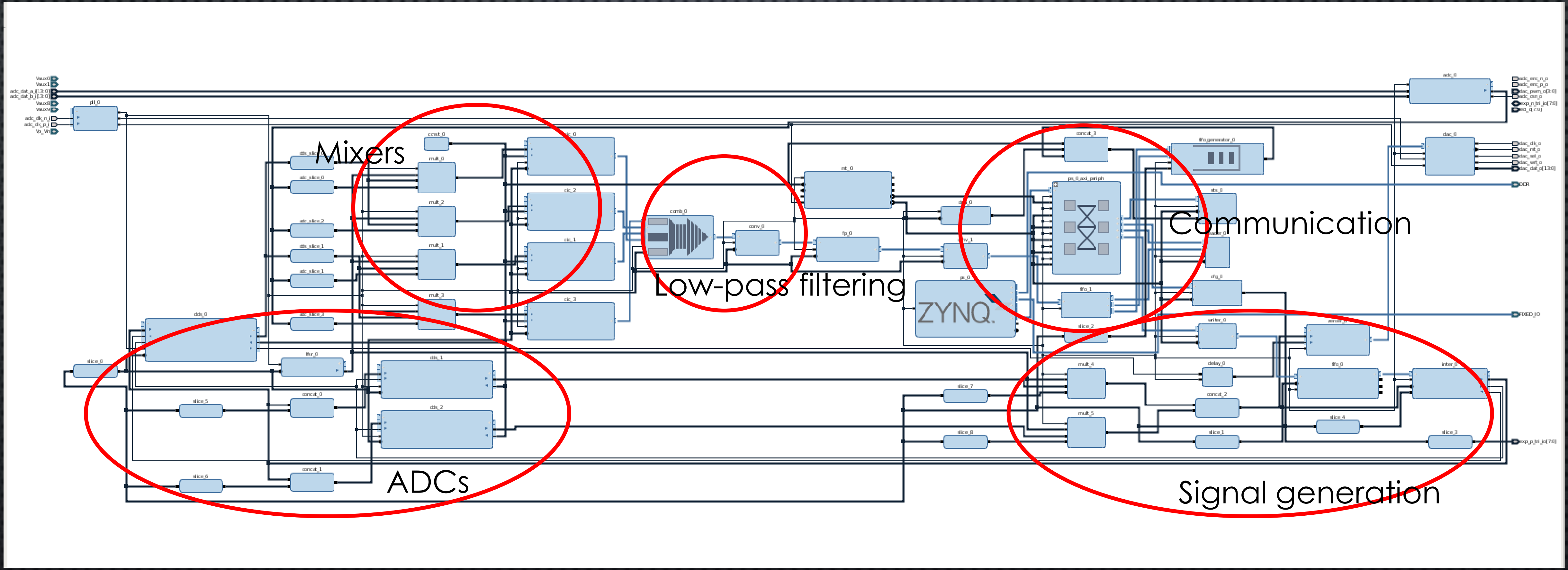
- A customizable data acquisition and signal generation platform
    - ½ FPGA + ½ CPU
- Advantages
    - Cheaper
    - Faster data transfer (TCP/IP rather than GPIB)
    - Highly flexible use
    - Red Potato
- Disadvantages
    - Cheaper for a reason

THE RED PITAYA AS TF MEASURING DEVICE



- Modified to run and take sweep parameters by command line (for easier automation)
- Automatically take multiple sweeps
- Pass data to IRIS
- Re-sweep at new frequencies

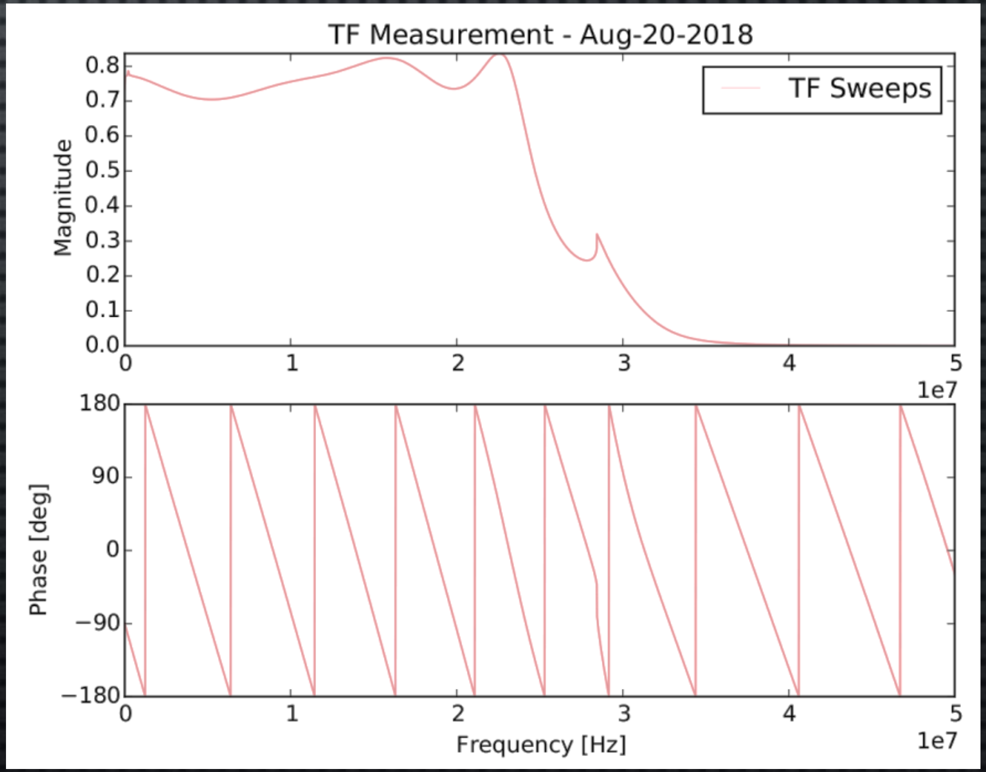
The Red Pitaya as TF Measuring Device



Pavel Demin (https://github.com/pavel-demin/red-pitaya-notes)

Unfortunately…

- Much too late in game – realized something was amiss with the Red Pitaya TF measurements
    - Wrapping in phase in weird way
- Got only to point where data can be grabbed from Red Pitaya and incorporated into algorithm
    - Insufficient calibration of data
    - Must run calculations ourselves separately from the Red Pitaya



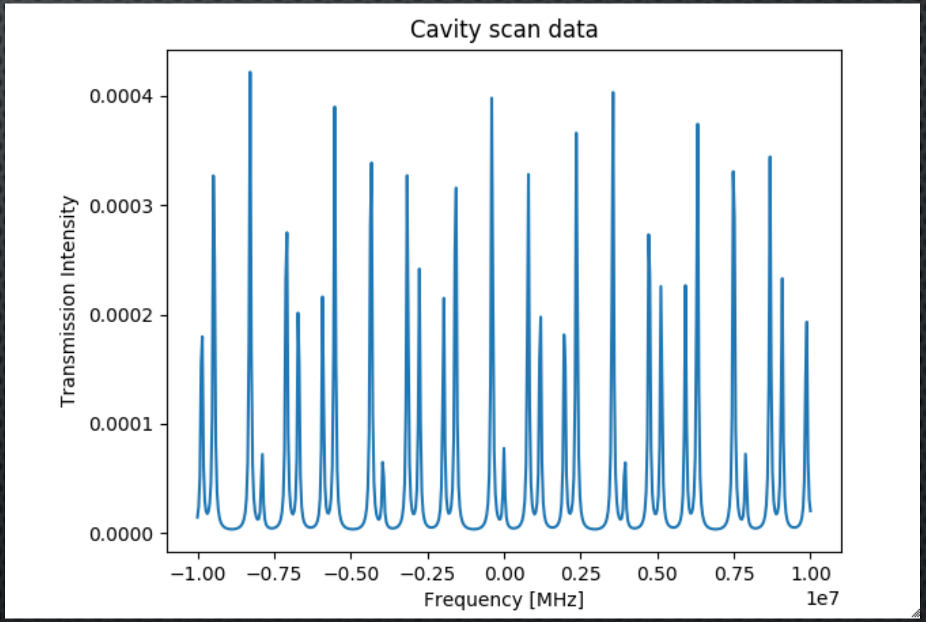
Conclusions, Next Steps, and What Could have Been…

- Engineering useful, functional technology is hard!! Love and appreciate your instrumentation people!!!

- Comment your code!

- Red Pitaya as it currently exists cannot quite yet do adaptive sweeping as desired
  - Adapt FPGA code to better fit needs of auto-res
  - Improve TF fitting model for use with Red Pitaya

- If things had worked, it could have:
  - Increased repetition rate of TF measurements
  - Made cavity arm scans at the 40m Prototype Lab a little easier



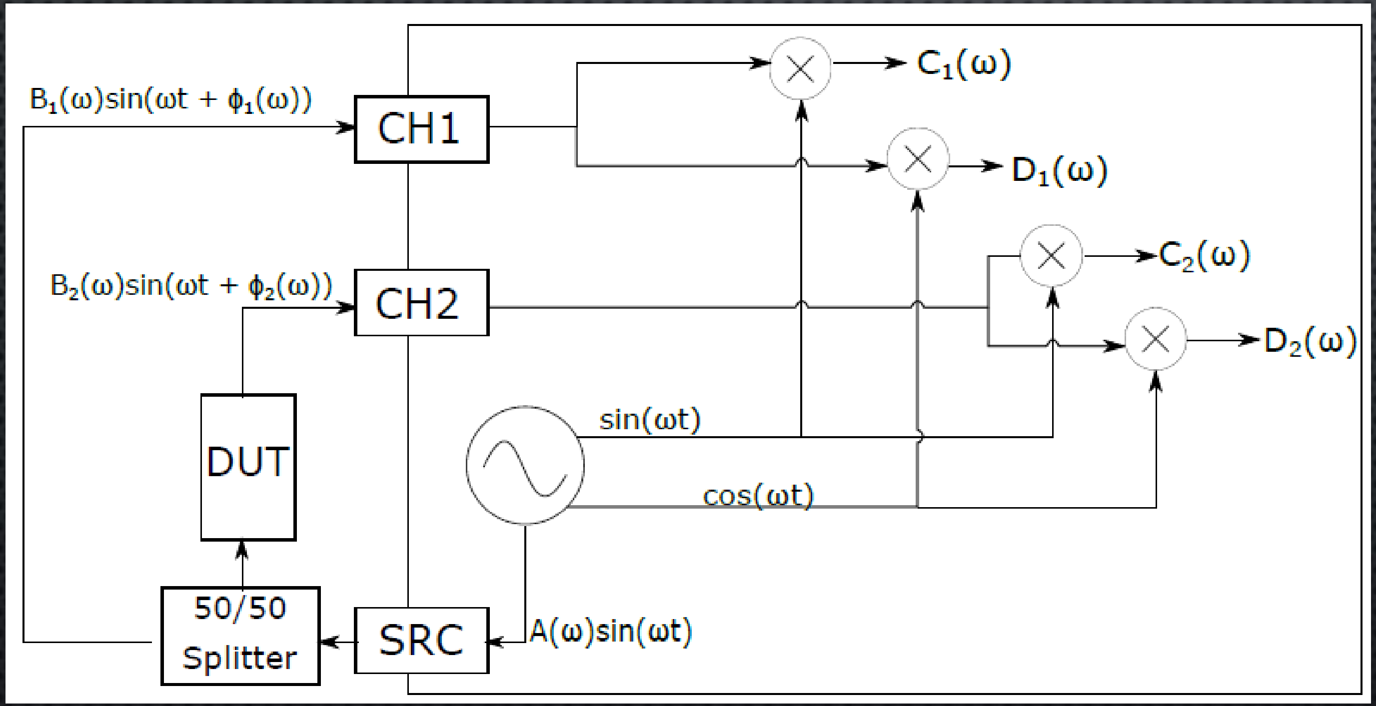
© Keerthana S. Nair

Thank You

The sincerest, most heart-felt, most enthusiastic thanks to

- Johannes Eichholz, Rana Adhikari, Christopher Wipf
- Caltech LIGO SURF
- Viewers like you ☺

TF MEASUREMENT MATH, HOORAY!



Where i = 1, 2

$$C_i = \frac{B_i}{2}[\sin(\phi_i) + \sin(\omega t + \phi_i)]$$

$$D_i = \frac{B_i}{2}[\cos(\phi_i) + \sin(\omega t + \phi_i)]$$

$$C_i^{\,2} + D_i^{\,2} = \frac{B_i^{\,2}}{4}(\sin^2(\phi_i) + \cos^2(\phi_i))$$

$$C_i^{\,2} + D_i^{\,2} = \frac{B_i^{\,2}}{4}$$

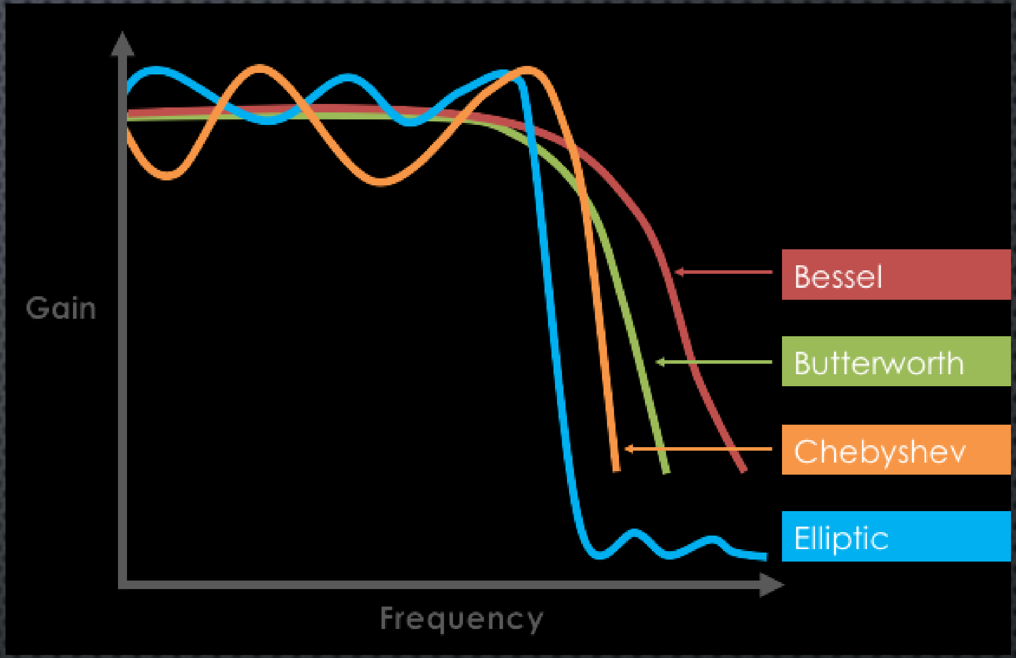
$$\frac{C_i}{D_i} = \frac{\frac{B_i 2}{\sin}(\phi_i)}{\frac{B_i 2}{\cos}(\phi_i)}$$

$$B_i = \sqrt{4(C_i^{\,2} + D_i^{\,2})}$$

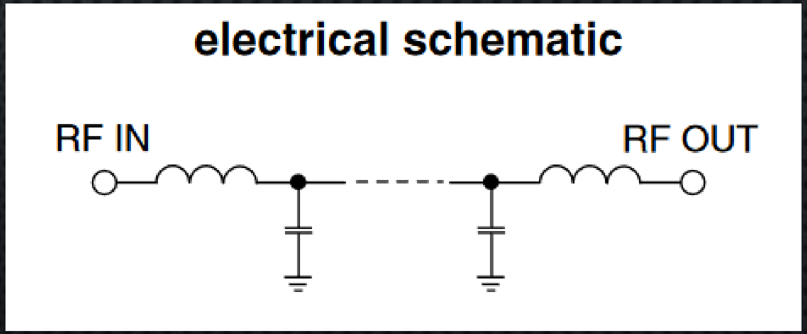
$$\frac{C_i}{D_i} = \tan(\phi_i)$$

Elliptic Low-Pass Filter

- Elliptic filters
  - Ripple behavior in both passband and stopband
  - Fastest transition in gain between pass- and stopband among filters of same order
- Why 4^{th} order
  - Rana



© Biley Technologies



© Mini-Circuits