Channel Equalization using GNU Radio

Joshua Morman
FOSDEM 2020

perspecta LABS

Agenda

Introduction

Need for Equalization

Equalizer Theory and Mechanics

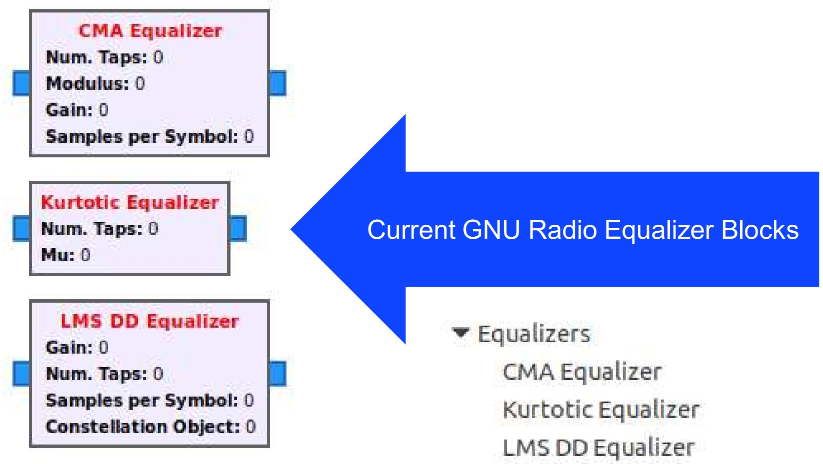
Types of Equalizers

GNU Radio Implementation

Future Work

Introduction

- Josh Morman
  - Senior Research Scientist
  - Perspecta Labs (NJ, USA)
  - GNU Radio Officer
- Motivation
  - Equalizers play a vital role in wireless communication systems
  - GNU Radio has some very good, functional equalizers, but was looking for more features than currently available
    - Equalize on training sequences
    - Expanded adaptive algorithms
    - Support for burst processing



Setup

- The examples shown in this presentation are here:

https://github.com/perspectalabs/gr-equalizers

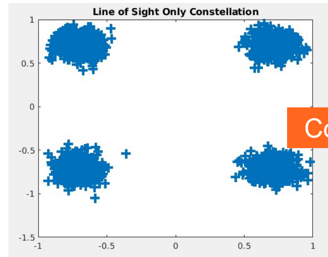
- GNU Radio 3.8 compatible on master branch

```
git clone https://github.com/perspectalabs/gr-equalizers
mkdir build
cd build
cmake ../
make && make install
```

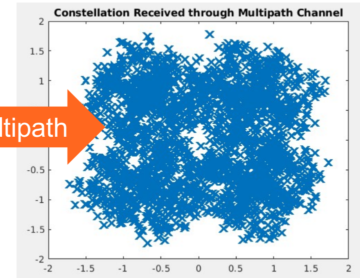
Channel Effects

Multipath channels

- Multipath channels
  - cause linear effects that make reception more difficult due to Inter Symbol Interference (ISI)
- A well designed equalizer can compensate for these effects and restore the expected constellation

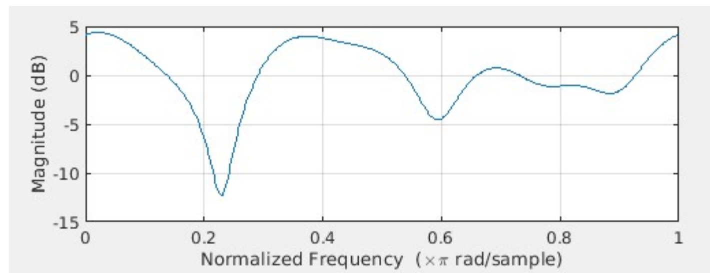


Compared with Multipath

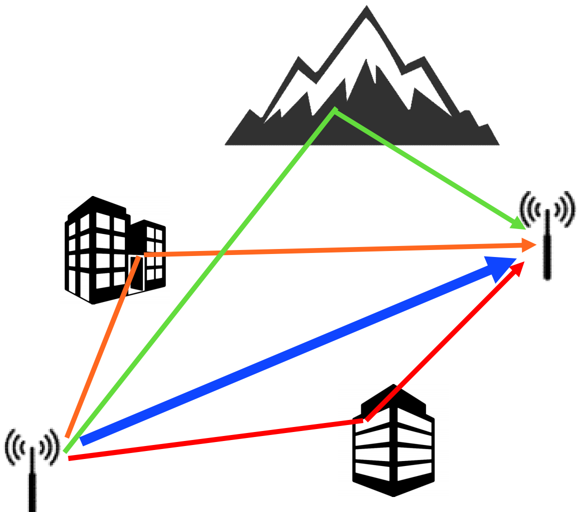



```
chan = [1.0,0,0,0,.3-.15j,0,.2+.33j, 0,0,0,0,0,.1+.03j ];
```

Notional Channel



Frequency Response of the channel will change as transmitter, receiver, or objects in the environment move around



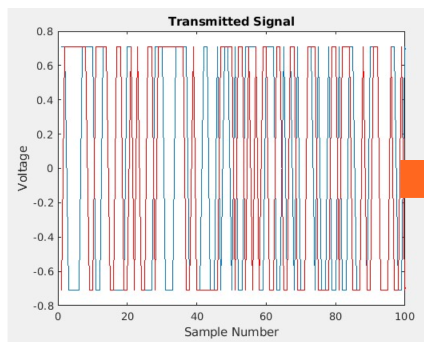
$[d_0 d_1 d_2 d_3 d_4 d_5 d_6 \ldots]$

$+ \quad [d_0 d_1 d_2 d_3 d_4 d_5 d_6 \ldots]$

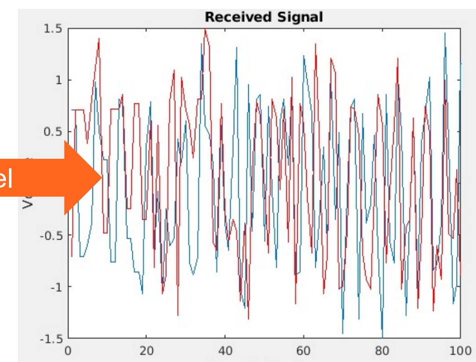
$+ \quad\quad [d_0 d_1 d_2 d_3 d_4 d_5 d_6 \ldots]$

$+ \quad\quad\quad [d_0 d_1 d_2 d_3 d_4 d_5 d_6 \ldots]$

$[u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8 u_9 u_{10} u_{11} u_{12} u_{13} \ldots]$



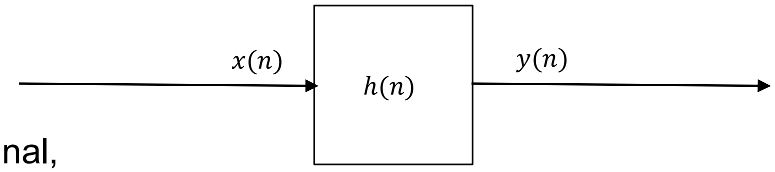
Across Channel



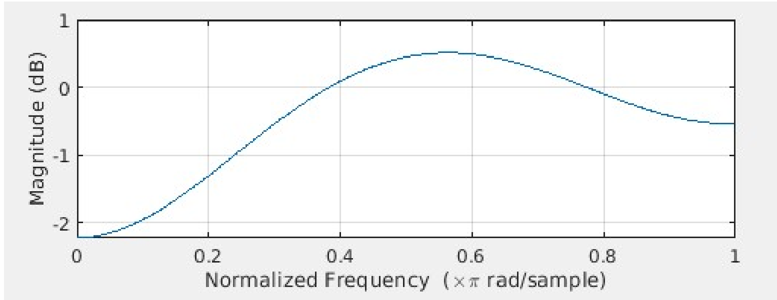
perspecta LABS

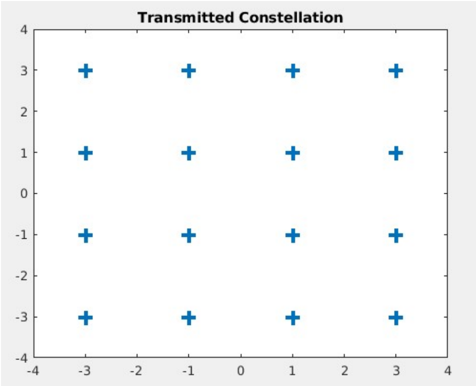
Channel Effects

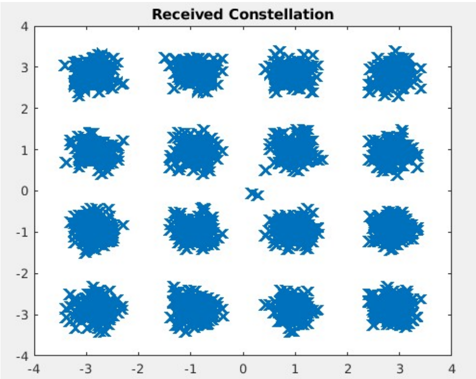
Hardware Filters



- When actual hardware is used to transmit and receive our signal, the frequency response of the filters, amplifiers in our transmit and receive signals also contribute to signal degradation
- May be non-linear effects as well (e.g. amplifier distortion)







Even the slight low frequency roll-off from this notional filter can significantly impact the signal

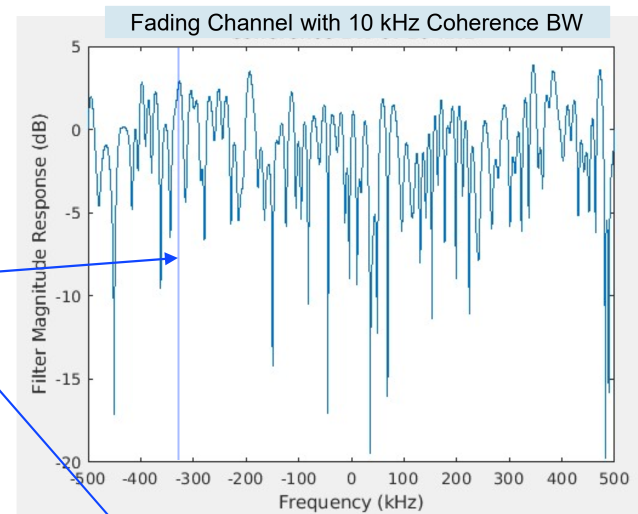
Channel Effects

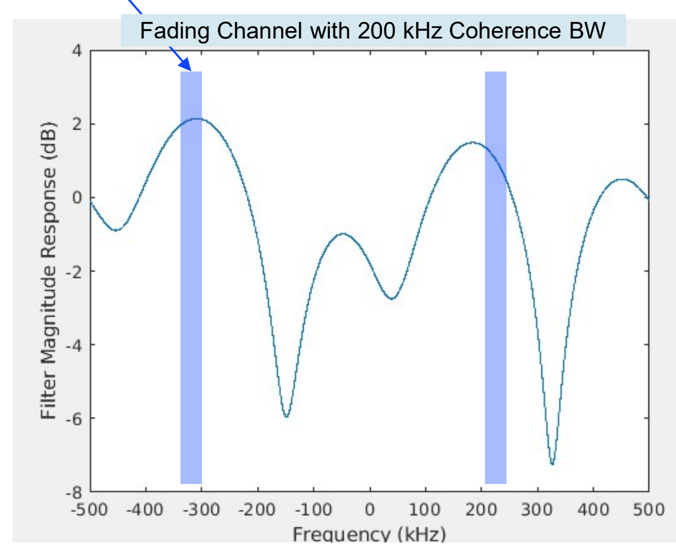
Frequency Selective Channels

Coherence Bandwidth ~ (1/Maximum Delay Spread)

- When the coherence bandwidth of the channel is:
  - larger than the bandwidth of the signal, we have a “flat fading“ channel
    - e.g. Narrowband signals
      - Time dispersion of the multipath is mostly contained within a symbol duration
  - smaller than the bandwidth of the signal, we get "frequency selective fading“ channel
    - e.g. Wideband signals
      - Time dispersion spans multiple symbols – large Intersymbol Interference (ISI) – the information from one symbol interferes with subsequent symbols

Flat Fading over this region



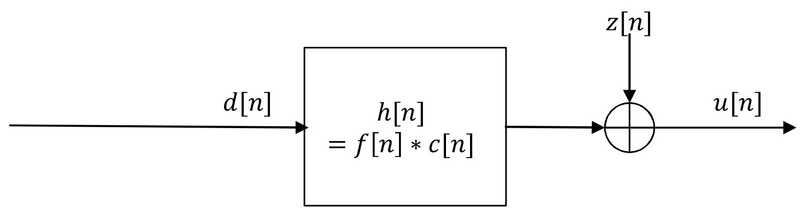


perspecta LABS

Equalizer Theory

Signal Model

- Suppose we have received a signal that has been passed through a composite linear channel which includes all the multipath and hardware filter responses, and experiences AWGN at the receiver



$$h(n) = \sum_{n=0}^{N-1} \alpha_n \delta(n-k)$$

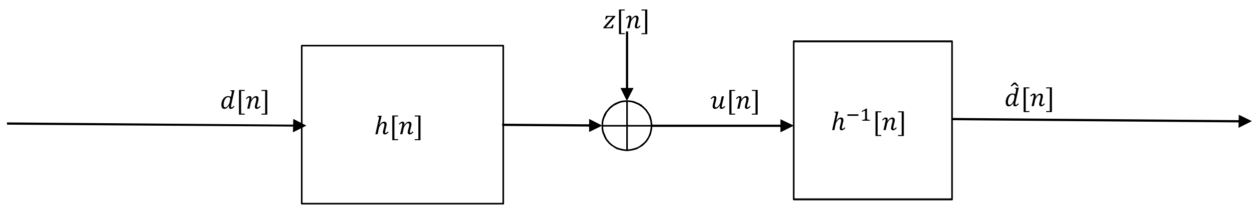
Linear Time Invariant channel $\mathbf{h}$ is the composite of the over the air channel and any hardware filters

- We consider the received signal our transmitted signal convolved with the impulse response of the channel filter $h(n)$
- For now we will assume Linear Time Invariant (LTI) channels

Equalizer Theory

Now equalize it … not so fast

- So it should logically follow that if we can cancel out the effect of h(n) by passing it through the inverse filter, then we can recover the original signal



- This is known as the **zero forcing equalizer**, and is difficult to do in practice
  - We must first be able to estimate H(f)
  - Then we must come up with an FIR approximation to $H^{-1}(f)$
    - This will be imperfect and therefore lead to imperfect cancellation
      - The response of the inverse filter will be infinite
  - Strong nulls in frequency domain will mean strong gain at those frequencies in the inverse filter
    - ZF filter will amplify the noise in those areas

Equalizer Theory

MMSE Criterion

- The optimal filter for reversing ISI is the Maximum Likelihood Sequence Estimator (MLSE) which requires finding the most likely path through a trellis of possibilities – Viterbi Algorithm
  - Computational complexity grows exponentially with length of channel response
- Define our available signals
  - $u(n)$: received signal; $e(n)$: error signal; $d(n)$: training symbols; $y(n)$: equalized signal; $w(n)$: FIR filter
- Let's look at a more reasonable criterion – **Minimum Mean Squared Error (MMSE)**
  - $e(n) = d(n) - y(n) = d(n) - \sum w_i u_{n-i}^*$

← our error is the difference between the correct decision and the received/filtered symbol

- Set up a cost function to minimize
  - $J_{min} = \mathrm{E}[|e(n)|^2]$
  - Come up with an optimum filter such that $\mathrm{E}[u(n-k)e^*(n)] = 0$, k=0,1,2,…
    - For the cost function to attain its minimum value, the estimation error needs to be *orthogonal* to each input sample
- Wiener-Hopf Equations - for FIR filter matrix formulation $\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$, $\mathbf{p} = E[\mathbf{u}(n)\mathbf{d}^*(n)]$
- $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$ <---- The optimum MMSE filter

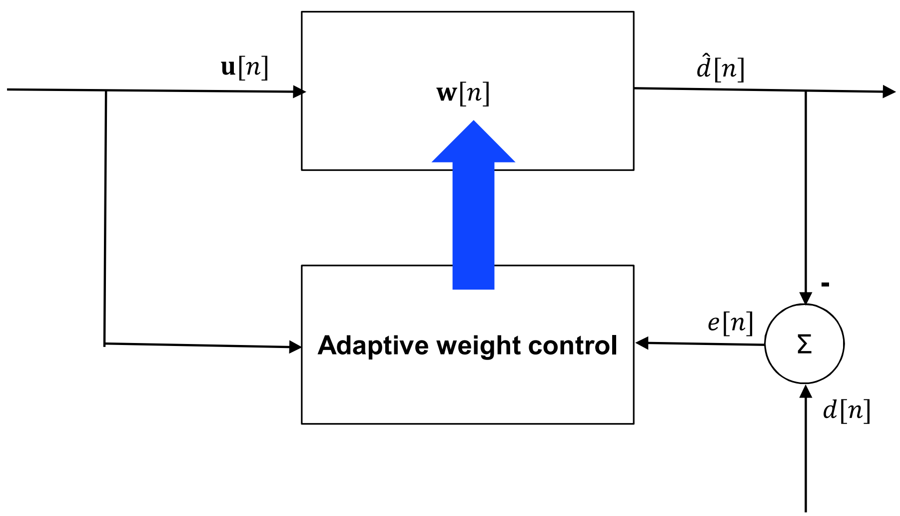
perspecta LABS

Equalizer Structures

Linear Equalizer

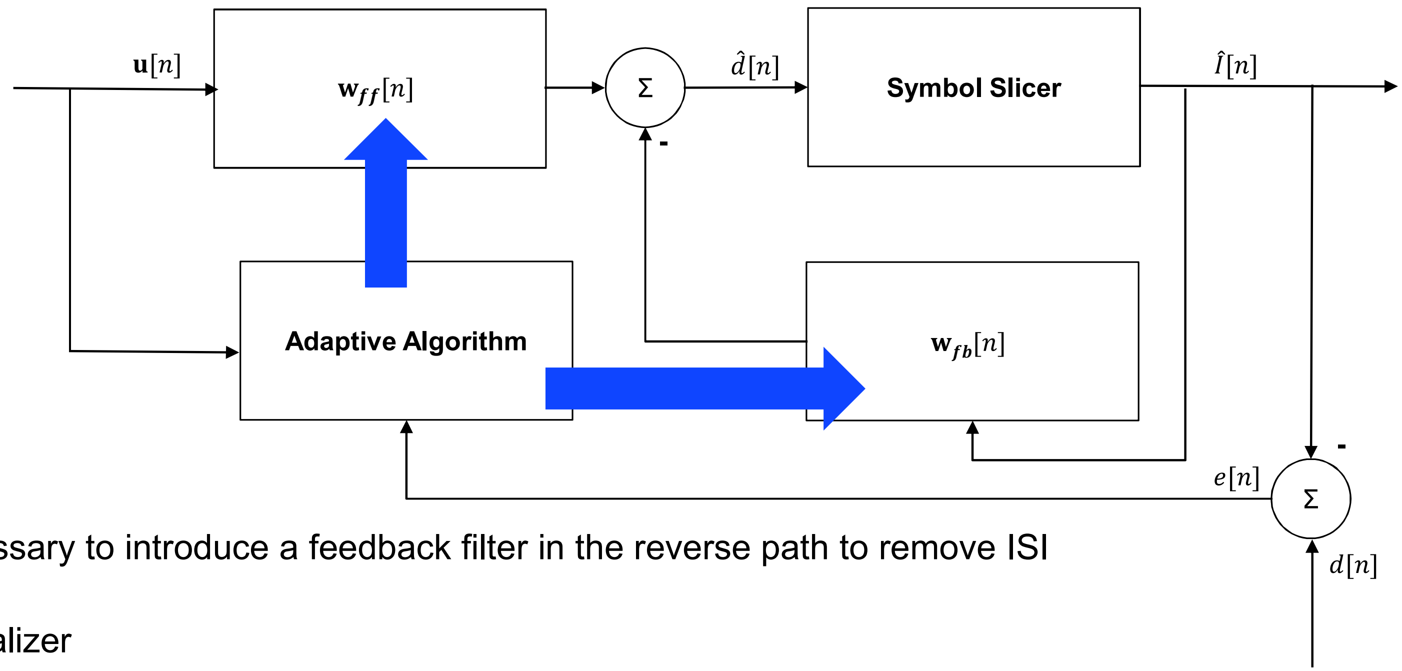
- The linear adaptive equalizer structure involves an FIR filter which gets periodically updated with new weights according to the generated error signal and how it relates to the current input sequence $\boldsymbol{u}(n)$
- Error is estimated as the difference between the filtered signal $\widehat{\boldsymbol{d}}(n)$ (estimate of the transmitted signal) and the known* transmitted sequence $\boldsymbol{d}(n)$
  - *In Decision Directed (DD) equalizers, $\boldsymbol{d}(n)$ is estimated as the closest constellation point to the filtered symbol

To start, we don't know anything about w[n] → depending on the adaptive algorithm we can set this to all 0's or an impulse



Equalizer Structures

Decision Feedback Equalizer (DFE)

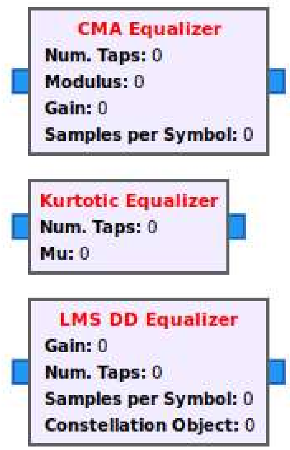


- For non-linear channels, it is necessary to introduce a feedback filter in the reverse path to remove ISI from previously detected symbols
  - This becomes a nonlinear equalizer
  - Feedback filter happens after symbol decisions are made
- Linear equalizers can emphasize noise in severely distorted channels – strong nulls in the frequency response

GNU Radio Equalizers

Existing blocks

- Let's revisit the existing GNU Radio Equalizer Blocks
- Having seen the Linear and DFE structures, we need adaptive algorithms to perform the weight updates
  - But the existing structure is not particularly modular
  - The Adaptive algorithm is baked into the block work() function
  - Would require many permutations of blocks/algorithms to cover some minor changes that we need to add additional algorithms and structures



Current GNU Radio Equalizer Blocks

We can see from the wiki page that the equalizers were indeed in need of some love

Kurtotic Equalizer

Implements a kurtosis-based adaptive equalizer on complex stream.

Warning: This block does not yet work (As of 2013). Is it still true? No real change since then. Need info on this.

```
"Y. Guo, J. Zhao, Y. Sun, "Sign kurtosis maximization based blind equalization algorithm," IEEE Conf. on Control, Automation, Robotics and Vision, Vol. 3, Dec. 2004, pp. 2052 - 2057."
```

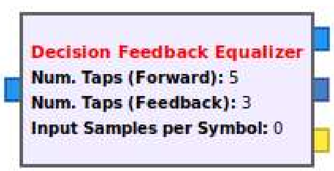
perspecta LABS

gr-equalizers

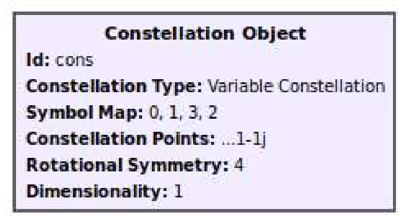
Some new Equalization blocks!!

- Adaptive Algorithm separated from the Equalizer Structure
    - More flexibility to add additional adaptive algorithms
- Linear Equalizer

- Decision Feedback Equalizer



- Algorithm object modeled off of digital::constellation object
- Holds a reference to a constellation object for making slicing decisions



gr-equalizers

- Both Linear and DFE
  - •Acts as a filter (derives from filter) same as stock GNU Radio CMA and LMS DD Equalizer
  - •Fractional spaced equalizer – decimates the signal by the configured samples per symbol
  - •Calculates error signal and updates filter taps
    1) If a tag is received as specified indicating start of training sequence
    2) If "Adapt After Training" option is True following training sequence
    3) If no training sequence is provided, will operate in DD mode

- Adaptive Algorithm
  - provides:
    - weight initialization method
    - tap update method
    - error estimation method
      - both for training and decision directed

Adaptive Equalization

Overview

- The adaptive algorithm defines how the weights of the equalizer will be updated according to the error signal and current estimates
- Many variations of adaptive algorithms that can be used in this context
  - Each with benefits and tradeoffs in different scenarios
    - MMSE – direct matrix inversion – not adaptive
    - LMS – Least Mean Square
      - Slower convergence, computationally simple
    - NLMS – Normalized LMS
    - RLS – Recursive Least Squares
      - Adapts more quickly, more computationally intensive
    - CMA – Constant Modulus Algorithm
      - Limited to constant modulus constellations (e.g. PSK)

Adaptive Equalization

Least Mean Square (LMS) Adaptive Algorithm

- Parameters
  - μ – step size
  - M – number of taps
- Initialize $\hat{\mathbf{w}}(0) = \mathbf{0}$ if no knowledge about the initial filter vector
- **At time n:**
  - $\mathbf{u}(n)$ *- Mx1 signal vector*
  - $d(n)$ *– data symbols (training or decision directed)*
  - Compute:
    - $e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$ — Error calculation
    - $\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n)$ — Tap weight to be used at the next step

We defined a cost function to minimize, which in M-dimensional space creates a surface

$$J_{min} = \mathrm{E}[|e(n)|^2]$$

Push $\hat{\mathbf{w}}$ in the steepest direction toward the minimum of J_{min}

NLMS has a slight modification of the weight update:

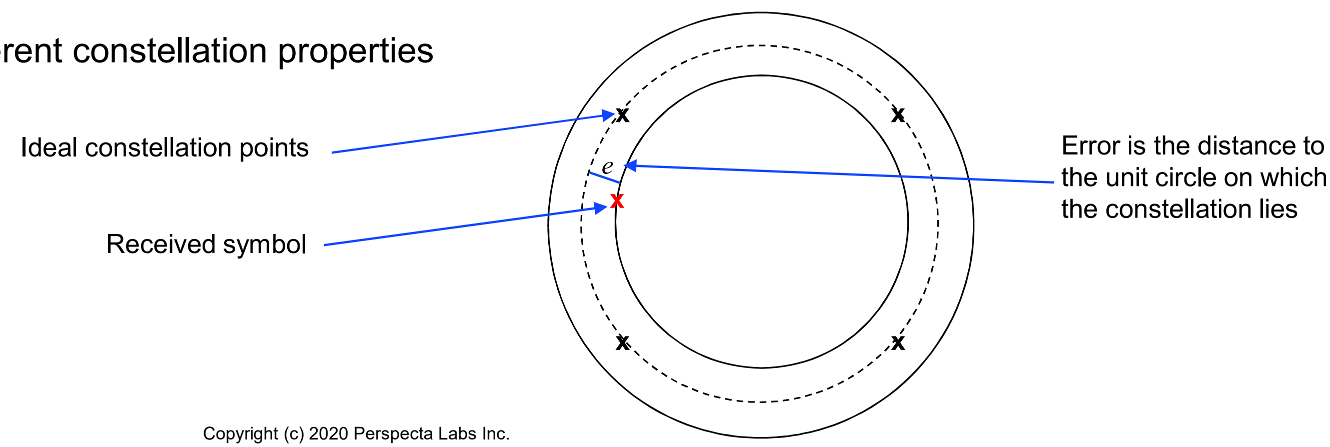
$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)e^*(n)$$

perspecta LABS

Adaptive Equalization

Constant Modulus Algorithm (CMA) Adaptive Filter

- Blind equalization scheme
- When we know some properties of the transmitted signal, such as it is constant envelope (e.g. PSK)
- Similar to Decision Directed equalization, we can estimate the error as a distance from the unit circle of a constant modulus constellation
- Assumes AGC prior to equalization, but don't care about phase
  - Do phase correction later
- Perform same weight update as LMS
- Other variants of CMA based on different constellation properties



perspecta LABS

Adaptive Equalization

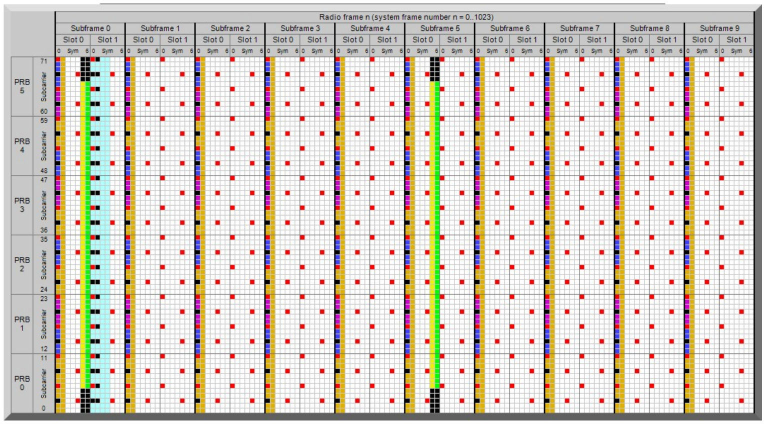
Recursive Least Squares (RLS) Adaptive Filter

- Parameters
  - λ – forgetting factor
  - δ – positive constant, set small for high SNR, large for low SNR
- Initialize $\hat{\mathbf{w}}(0) = \mathbf{0}$, $\mathbf{P}(0) = \delta^{-1}\mathbf{I}$
- **At time n:**
  - $\boldsymbol{\pi}(n) = \mathbf{P}(n-1)\mathbf{u}(n)$
  - $\mathbf{k}(n) = \frac{\boldsymbol{\pi}(n)}{\lambda + \mathbf{u}^H(n)\boldsymbol{\pi}(n)}$
  - $\xi(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}(n)$ — Error calculation
  - $\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\xi^*(n)$ — Tap weights to be used at the next step
  - $P(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1)$

Adaptive Equalization

Further Extensions

- Good news about having separate algorithms from equalizer structure, is that now we can add more algorithms and structures
  - Many variations of adaptive algorithms and equalizer structures
- Neural network based equalizers
- OFDM
  - Beauty of OFDM is that each subchannel appears as flat fading channel
  - Delay spread is contained within the confines of the cyclic prefix
  - Simpler per RB equalization in frequency domain across the grid
    - Training symbols spaced in frequency and time
    - Interpolation across blocks



LTE Resource grid [from http://niviuk.free.fr/lte_resource_grid.html]

Burst Processing

Implementation Note

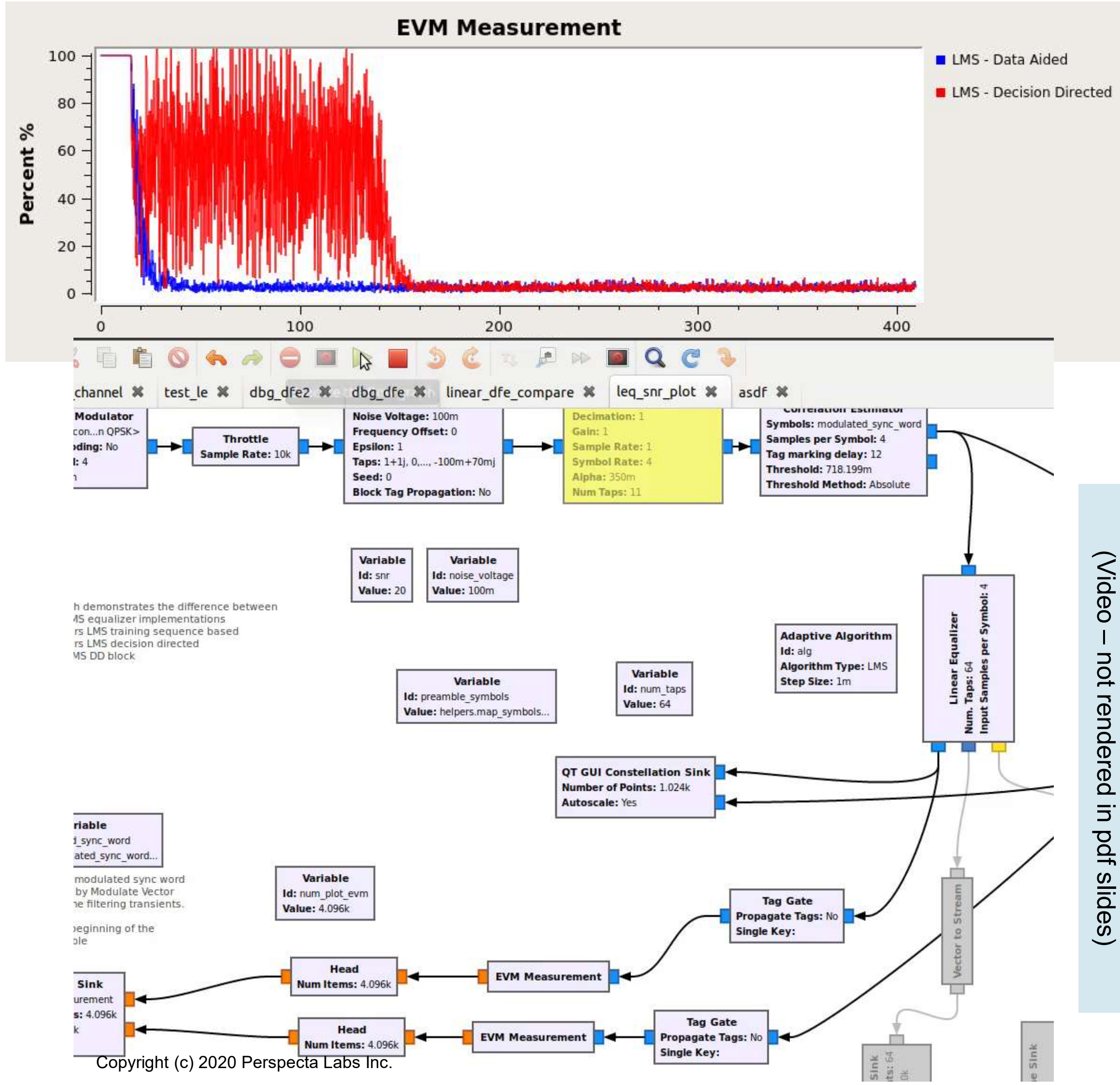
- A key driver of this work was the ability to handle bursty data, as in non-continuous modulated data, it is necessary to implement the blocks in a way that don't make the assumption of tagged stream blocks
- Some GNU Radio modules do a good job of allowing the core algorithm to be called from outside the context of a streaming block
  - E.g. Filters have a separate kernel namespace (as in what is inherited in these equalizer blocks)
- Want to be able to use the same algorithms within different streaming contexts – Streaming blocks, PDU blocks, maybe even outside a flowgraph altogether
- In this implementation, rather than performing all signal processing in work functions, have a separate function that is called from work() that does all non-scheduler specific things
  - Unfortunately the work() function is tied in with the scheduler and can't be called standalone
  - Hopefully this will be addressed in the near future as it will open up much flexibility for gr blocks

perspecta LABS

gr-equalizers

Performance Comparison

- LMS gain = 0.001
- In 20dB SNR, we can see that the DD equalizer takes longer to adapt, but once it does, the performance is similar



(Video – not rendered in pdf slides)

Error Vector Magnitude (EVM) Measurement is

$$\sqrt{\frac{P_{error}}{P_{ref}}}x100\%$$

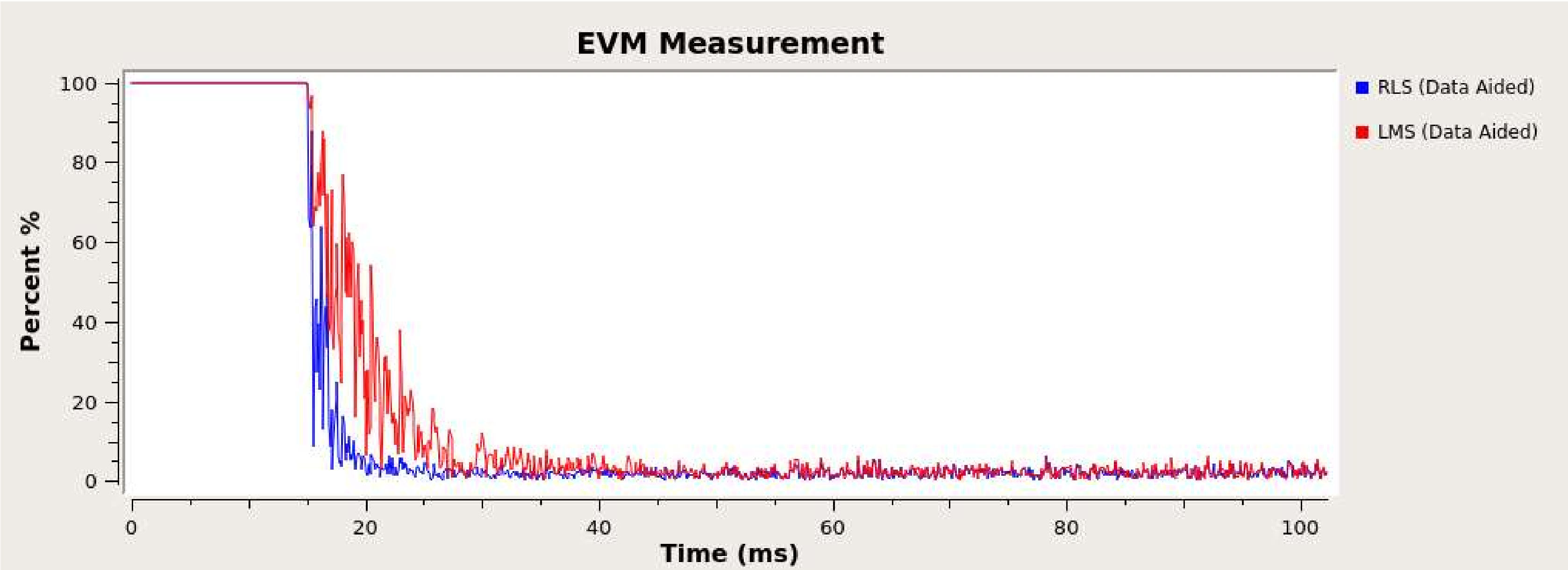
Where P_{ref} is the average power of the constellation

perspecta LABS

gr-equalizers

Performance Comparison

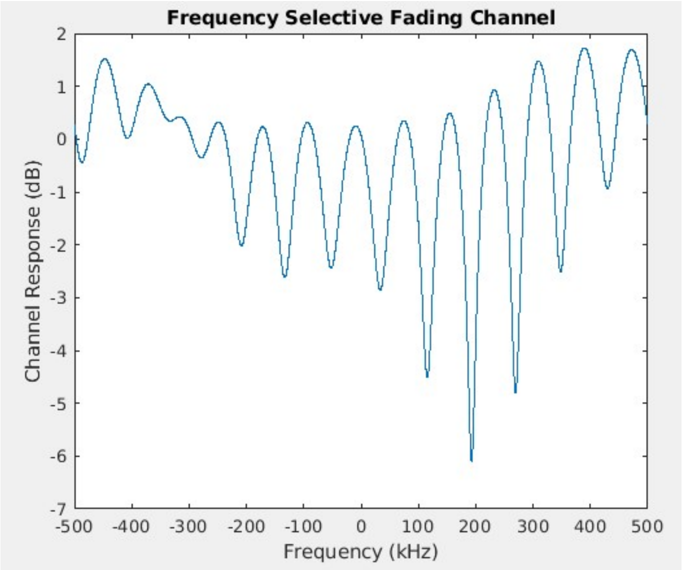
- RLS with \lambda = .999
- LMS with \mu = .001
- RLS converges more quickly but with higher computational cost



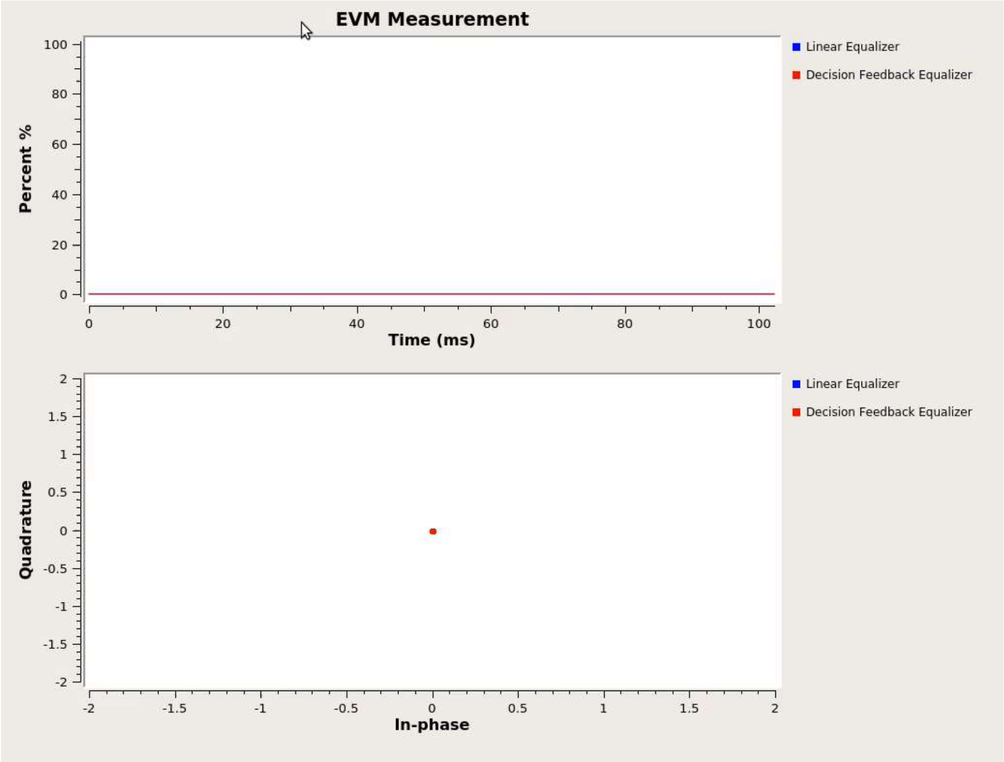
gr-equalizers

Performance Comparison

- When we have a channel with deep frequency nulls, a DFE can outperform a Linear Equalizer
- 10 dB SNR



(Video – not rendered in pdf slides)



Adaptive Equalization

Other Resources

- gr-adapt
  - https://github.com/karel/gr-adapt
  - Excellent implementations of adaptive filters for:
    - Adaptive line enhancer
    - Self-interference cancellation
    - System identification
    - Time delay estimation
- Lectures on MMSE, LMS, RLS, etc:
  - https://www.youtube.com/watch?v=4prlftiKpUY
  - https://www.youtube.com/watch?v=BM7i8LHFwyY
  - https://www.youtube.com/watch?v=gHSiFqO23TE

References

- Haykin, S. Adaptive Filter Theory, Fourth Edition, Prentice-Hall Englewood Cliffs (2002)
- Proakis, Digital Communications. McGraw-Hill, 4th edition (2001)
- Proakis, John G. "Adaptive equalization for TDMA digital mobile radio." IEEE Transactions on Vehicular Technology 40.2 (1991): 333-341.
- Belfiore, Carlos A., and John H. Park. "Decision feedback equalization." Proceedings of the IEEE 67.8 (1979): 1143-1156.
- Tront, R. J. (1984). Performance of Kalman decision-feedback equalization in HF radio modems (Doctoral dissertation, University of British Columbia).
- Takach, Andres, Bryan Bowyer, and Thomas Bollaert. "C based hardware design for wireless applications." Proceedings of the conference on Design, Automation and Test in Europe-Volume 3. IEEE Computer Society, 2005.
- Brink, JW Laura, Anant Sahai, and John Wawrzynek. "Deep Networks for Equalization in Communications." Master’s thesis, University of California, Berkeley (2018).