CHAIN: Week of June 10

June 10

- Goals:
  - Finish assigned tasks
- What I learned:
  - Nyquist theorem for complex numbers
  - More about Parks McClellan algorithm

h[n] vs h[t] vs H[n]

1. Impulse Response h[n]

- **Definition**: h[n] is the impulse response of a discrete-time system.
- **Discrete-Time**: The impulse response h[n] describes how a discrete-time system responds to an impulse input (a sequence where one element is 1 and the rest are 0).
- **Notation**: n represents discrete-time indices (integers).
- **Usage**: Used in digital signal processing (DSP) to describe filters and systems that operate on discrete-time signals.

2. Impulse Response h[t]

- **Definition**: h[t] is the impulse response of a continuous-time system.
- **Continuous-Time**: The impulse response h[t] describes how a continuous-time system responds to an impulse input (a Dirac delta function).
- **Notation**: t represents continuous-time indices (real numbers).
- **Usage**: Used in analog signal processing and control systems to describe filters and systems that operate on continuous-time signals.

3. Frequency Response H[n]

- **Definition**: H[n] typically represents the frequency response of a discrete-time system.
- **Frequency Domain**: The frequency response H[n] is the Fourier transform of the impulse response h[n]. It describes how the system responds todifferent frequency components of the input signal.
- **Notation**: n often represents discrete frequency indices.
- **Usage**: Used to analyze and design filters in the frequency domain. It shows how the system modifies the amplitude and phase of sinusoidal components at different frequencies.

h[n] vs h[t] vs H[n]

- **DTFT Relationship**:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}$$

Here, $H(e^{j\omega})$ is the DTFT of $h[n]$, providing the frequency response of the discrete-time system.

- **Inverse DTFT**:

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega$$

- **Continuous-Time Fourier Transform (CTFT) Relationship**:

$$H(j\omega) = \int_{-\infty}^{\infty} h[t]e^{-j\omega t}dt$$

Here, $H(j\omega)$ is the CTFT of $h[t]$, providing the frequency response of the continuous-time system.

- **Inverse CTFT**:

$$h[t] = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(j\omega)e^{j\omega t}d\omega$$

How Does One Ensure Linearity?

Fundamental Properties of Frequency Sampling Methods

Linearity

- **Definition**: Linearity in signal processing means that the response of a system to a weighted sum of inputs is equal to the weighted sum of the responses to each individual input.
- **Linearity in Filters**: For a filter designed using frequency sampling methods to ensure linearity, the system must satisfy the principle of superposition: if $x_1[n]$ and $x_2[n]$ are inputs, and $y_1[n]$ and $y_2[n]$ are the corresponding outputs, then the response to $a \cdot x_1[n] + b \cdot x_2[n]$ should be $a \cdot y_1[n] + b \cdot y_2[n]$ for any constants a and b.

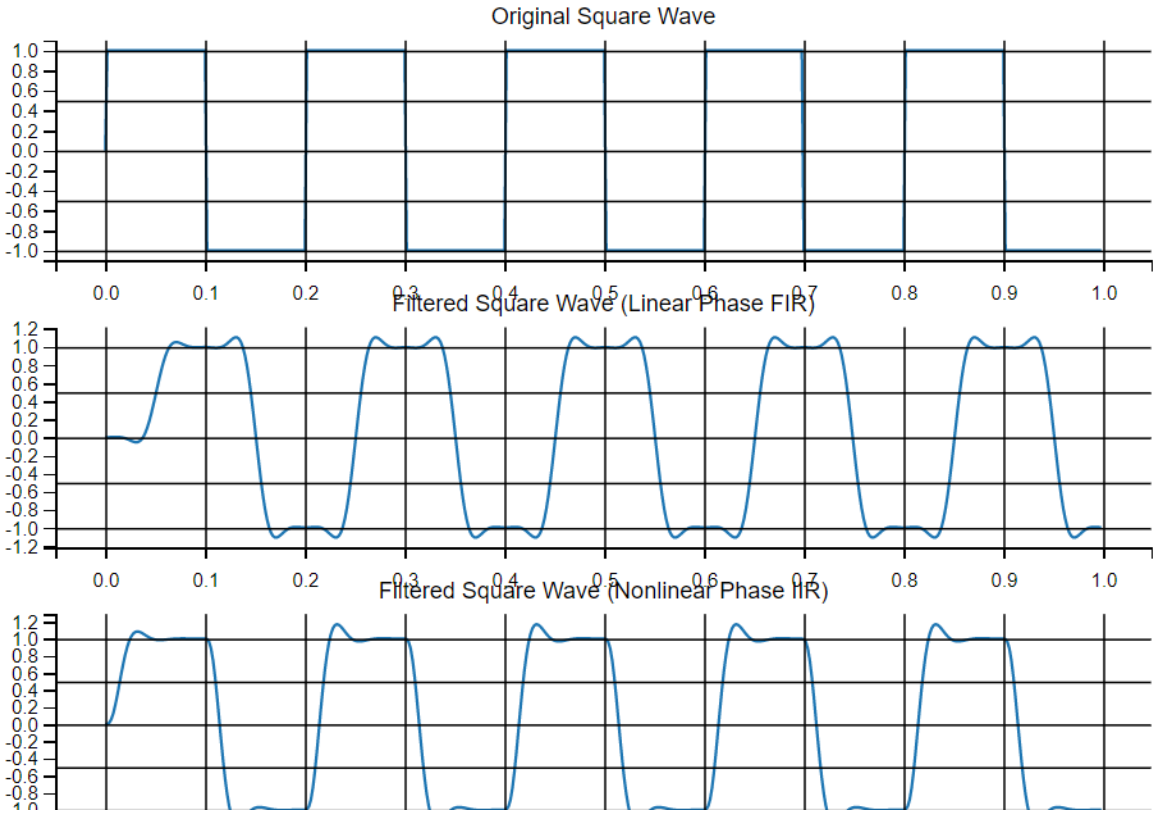
Frequency Sampling Methods and Linearity

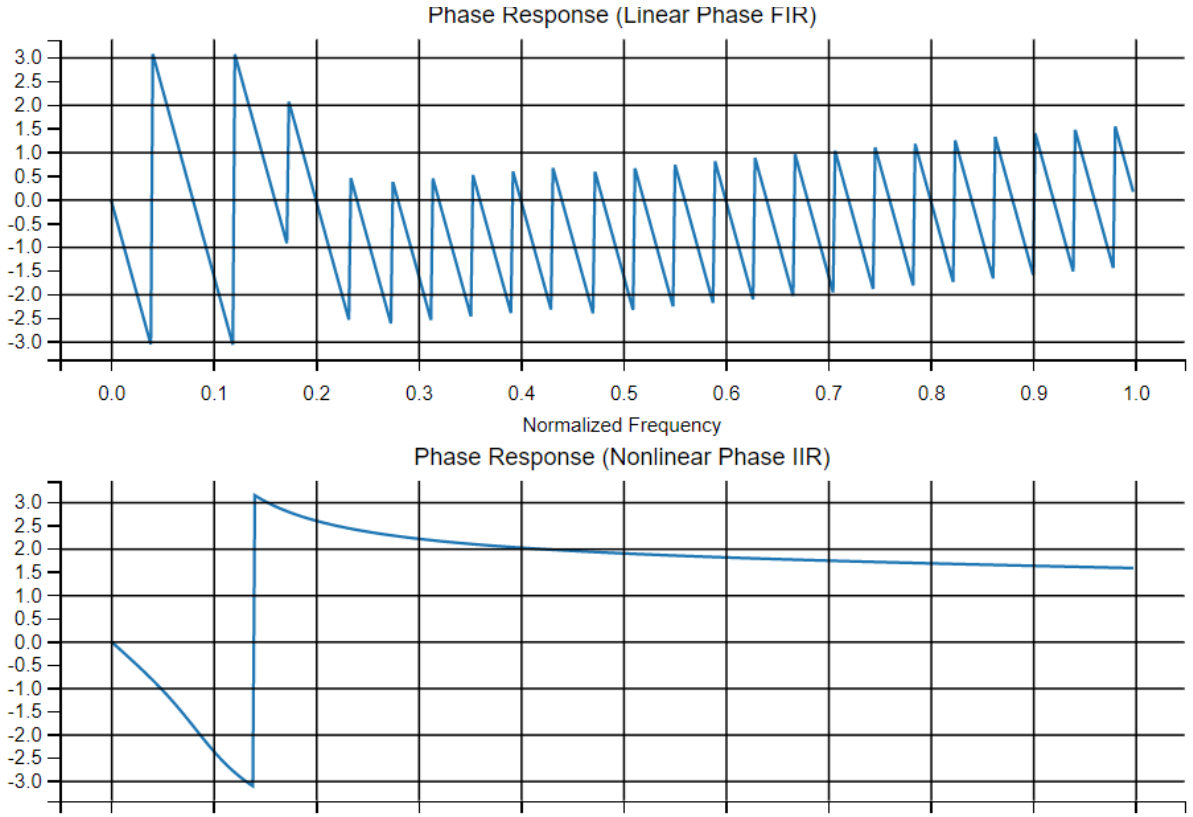
- Not all frequency sampling methods ensure linearity
  - Modifies phase response (nonlinear phase response)
    - Linear Phase Response
      - **Definition**: A filter has a linear phase response if the phase response is a linear function of frequency. This implies that all frequency components of the input signal are delayed by the same amount of time, preserving the wave shape of signals.
      - **Characteristics**:
        - **Phase Response**: $\phi(\omega) = -\omega\tau$
        - **Impulse Response**: For FIR filters, having a symmetric (or anti-symmetric) impulse response ensures a linear phase.
  - Warps frequency axis
- Examples:
  - IIR frequency responses
  - Asymmetric impulse responses
  - Frequency warping techniques

Why is Linearity Important?

- Preserves signal shape
- No intersymbol interference
- Consistent group delay
  - Group delay ensures accurate timing and synchronization
- Accurate frequency representation
- Reduced signal by-products
  - Minimizes by-products such as ringing, overshoot, etc.
    - Ringing: oscillations that follow a sharp transition in a signal
    - Overshoot: output signal exceeds final steady-state value in response to step input

Linear vs Non-Linear





Intersymbol Interference (ISI)

- Form of distortion of a signal in which one symbol interferes with subsequent symbols.
- Objective for receivers and transceivers is to minimize ISI
- Occurs when a pulse spreads out in such a way that it interferes with adjacent at the sample input.

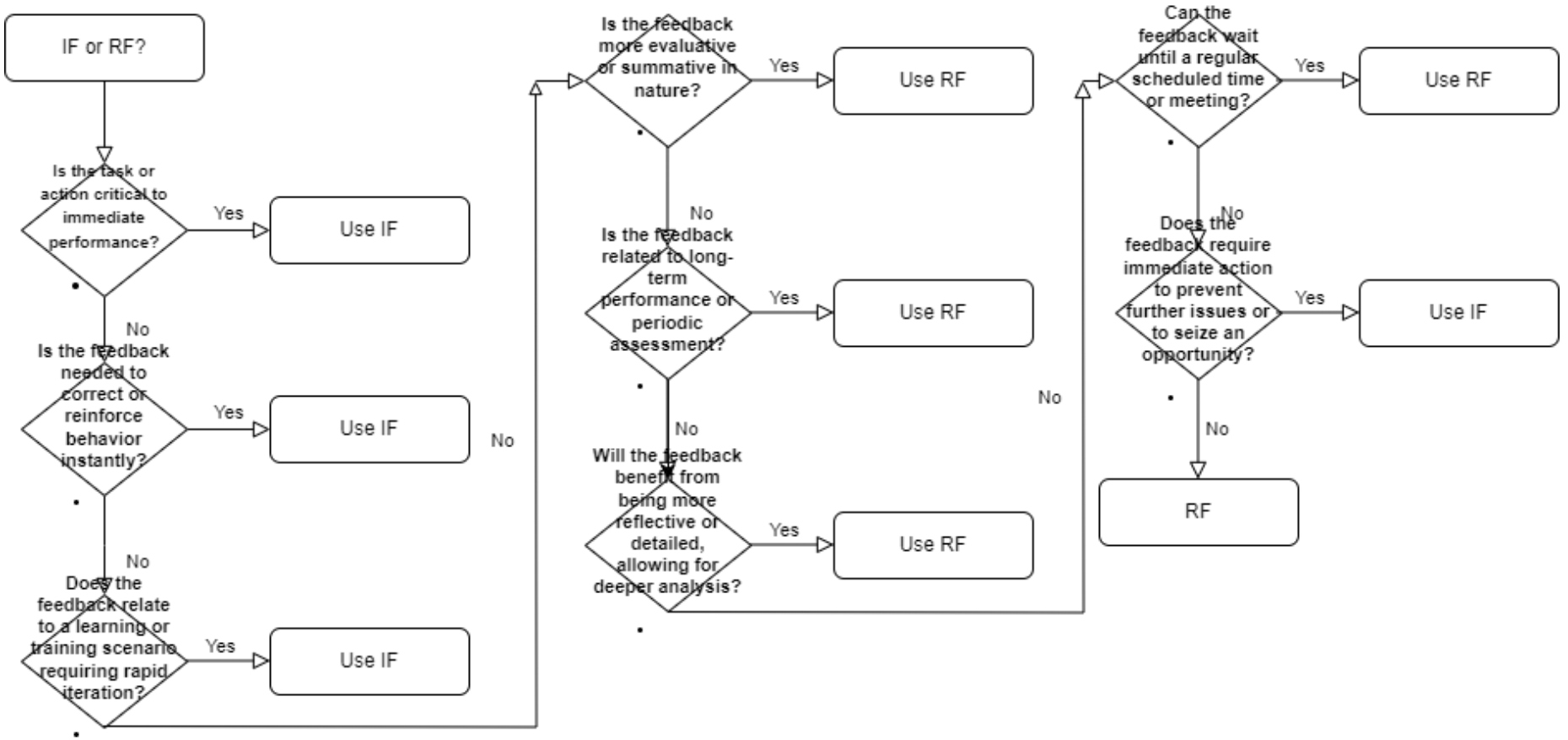
Causes:

- **Multipath Propagation**: In wireless communication, the signal can take multiple paths to reach the receiver due to reflections, scattering, and diffraction. These multiple paths can cause the delayed versions of the signal to arrive at the receiver, overlapping with subsequent symbols.
- **Bandwidth Limitation**: If the transmission channel has a limited bandwidth, it cannot perfectly transmit the signal, causing it to spread out in time. This spreading causes symbols to overlap, resulting in ISI.
- **Dispersive Channels**: In wired communication, like optical fibers and copper cables, different frequency components of the signal may travel at different speeds, causing parts of the signal to arrive at different times.

Effects:

- **Signal Distortion**: ISI causes distortion of the received signal, making it harder to accurately detect the transmitted symbols.
- **Increased Bit Error Rate (BER)**: As the symbols interfere with each other, the probability of incorrect symbol detection increases, leading to a higher BER.
- **Degradation of Communication Quality**: Overall communication quality degrades, affecting the reliability and efficiency of the communication system.

IF vs RF


IF or RF?
Is the task or action critical to immediate performance?
Yes
Use IF
No
Is the feedback needed to correct or reinforce behavior instantly?
Yes
Use IF
No
Does the feedback relate to a learning or training scenario requiring rapid iteration?
Yes
Use IF
No
Is the feedback more evaluative or summative in nature?
Yes
Use RF
No
Is the feedback related to long-term performance or periodic assessment?
Yes
Use RF
No
Will the feedback benefit from being more reflective or detailed, allowing for deeper analysis?
Yes
Use RF
No
Can the feedback wait until a regular scheduled time or meeting?
Yes
Use RF
No
Does the feedback require immediate action to prevent further issues or to seize an opportunity?
Yes
Use IF
No
RF

Nyquist Theorem for Complex Numbers

- **Nyquist Rate**: The Nyquist theorem states that the sampling rate fs must be at least twice the highest frequency component of the signal. This highest frequency component is often referred to as fmax .
  - $f_s \geq 2f_{max}$
- Real numbers exhibit symmetry in their positive and negative domains. If you know your spectrum from 0 to fs/2, then you can conclude what the other half is.
- A complex signal sampled at rate fs can unambiguously contain content from $-f_s/2$ to $f_s/2$
- Each complex sample contains two components (real and imaginary), so while you require half as many samples, each requires twice the amount of data storage, which cancels out any immediate benefit.

Original vs Image Frequencies

Original Frequencies

- **Definition**: Original frequencies refer to the frequencies present in the original continuous-time signal before any sampling process. These are the true frequencies of the analog signal, and they can range from 0 Hz up to the highest frequency component present in the signal.

Image Frequencies

- **Definition**: Image frequencies, or aliases, are frequencies that appear in the sampled (discrete-time) signal as a result of undersampling. They are not present in the original signal but arise due to the sampling process when the sampling rate is not high enough to capture the original signal's highest frequency components without distortion.

Parks McClellan Algorithm

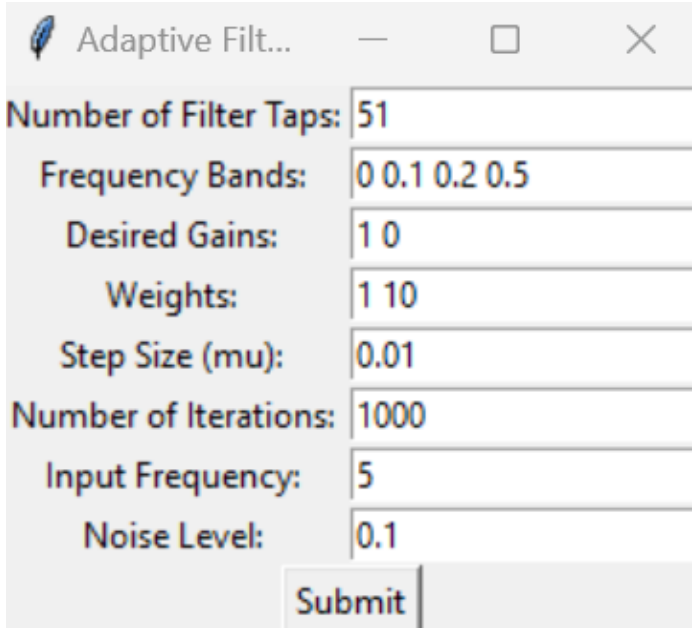
- The Parks-McClellan algorithm performs an iterative process known as the Remez exchange, where it alternates between approximating the desired frequency response and minimizing the approximation error.
- During each iteration, the algorithm exchanges the error function's extrema, ensuring that the filter response closely matches the desired response within the specified tolerance.
- The exchange process continues until the approximation error reaches an acceptable level or the maximum number of iterations is reached. The result is a set of filter coefficients that meets the desired frequency response with minimal error, creating a high-performance FIR filter.

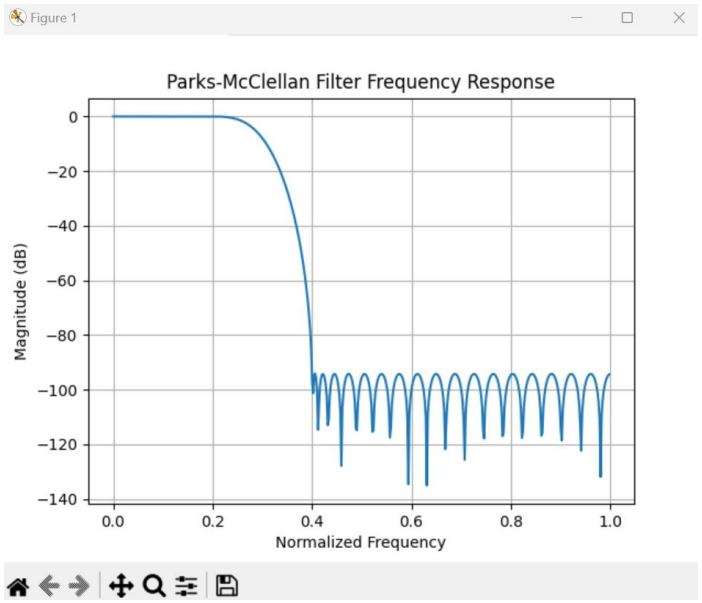
Parameter Ranges

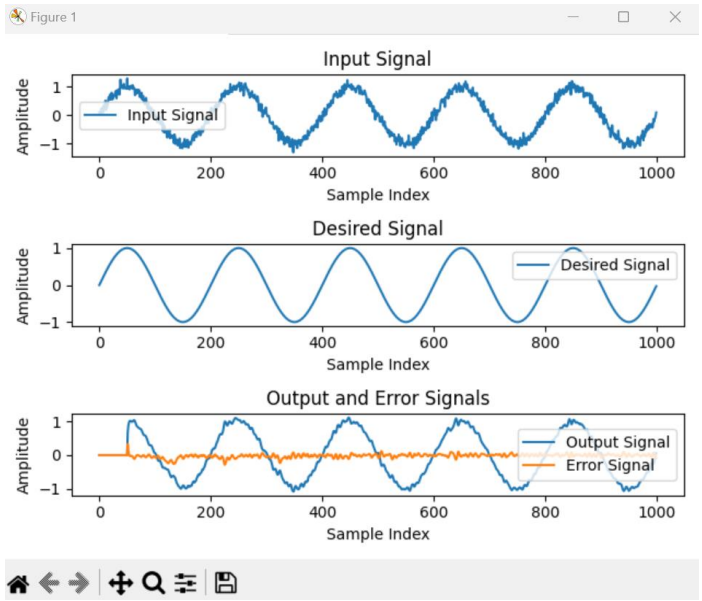
- **Number of Filter Taps (Coefficients)**:
  - Optimal: 20 to 100, depending on the application.
  - Too few: < 20 (may not perform well).
  - Too many: > 200 (often unnecessary and computationally expensive).
- **Frequency Bands and Desired Responses**:
  - Frequency Bands: 0 to 0.5 (normalized frequency).
  - Desired Gains: Typically 0 (stopband) or 1 (passband), but intermediate values are possible.
- **Weights**:
  - Positive values, typically ranging from 1 to 10, depending on the relative importance of each band.
- **Step Size (μ)**:
  - Range: 0.001 to 0.1.
  - Optimal step size should balance between convergence speed and stability.

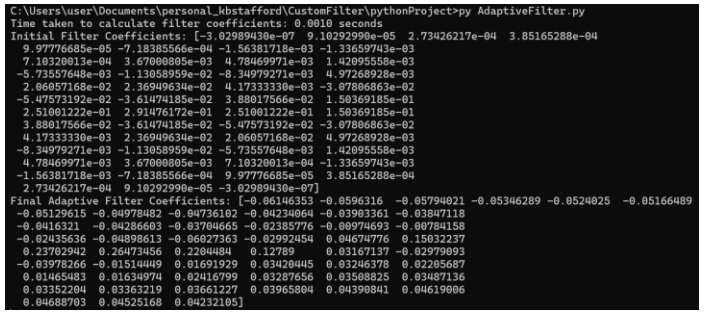
Parks McClellan Algorithm Steps

1. Specification
   - Parameters: pb, sb, ripple, order, etc.
   - Frequency domain
2. Initial Guess
   - Chebyshev polyniomial
   - Follows a deterministic procedure based on initial conditions and the specified constraints
3. Error Function
   - The error function measures the deviation of the actual filter response from the ideal response, aiming to minimize the maximum error (ripple) in the passband and stopband.
   - The error is determined by evaluating the deviation of the filter's frequency response from the desired response at a set of extremal frequencies. The goal is to minimize the maximum error across these frequencies (minimax error).
4. Remez Exchange
   - The Remez exchange algorithm iteratively adjusts the extremal frequencies to achieve an equiripple response. This involves finding a set of N+2N+2N+2 extremal frequencies where the error alternates in sign and has equal magnitude.
   - The algorithm fits a weighted minimax polynomial to the desired frequency response. The weights can be adjusted to prioritize different parts of the frequency spectrum.
     - The algorithm can handle multiple passbands and stopbands, optimizing the filter across all specified frequency bands.
5. Convergence
   - The algorithm iteratively refines the extremal frequencies and the filter coefficients until convergence criteria are met, typically when the change in the error function falls below a specified threshold.
6. Result







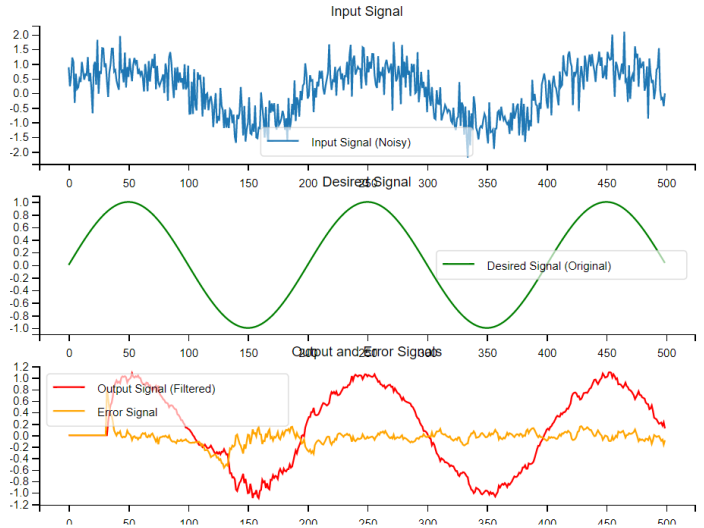


AdaptiveFilter.py

- Parks McClellan algorithm
  - Remez exchange
    - Chebyshev approximations
    - Minimize the maximum frequency-response error-magnitude over the frequency axis
    - Let x be continuous on [a, b] and let ε > 0. Then, there exists a polynomial p for which $|e(t)| = |x(t) - p(t)| \leq \varepsilon$ for every $t \in [a, b]$.
- Adaptive filtering
  - LMS optimization
    - LMS with regard to what? What are you minimizing?
      - Minimize MSE between desired signal and filter output
- Plots frequency response, input signal, desired signal, output and error signals
- Gives choice between GUI implementation and terminal implementation

LMS

- Reduced noise and better alignment with desired signal
- Over time, as the LMS algorithm iterates and adjusts the filter coefficients, the output signal should become closer to the desired signal.



$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n)$$

where $\mathbf{w}(n)$ are the filter coefficients, μ is the step size, $e(n)$ is the error signal, and $\mathbf{x}(n)$ the input signal vector.

June 11

- Goals:
  - Improve content on slides
  - Further foundational knowledge
- What I learned:
  - Parks McClellan steps (further in depth)
  - LMS (further in depth)

June 12

- Goals:
  - Further foundational knowledge
    - Write notes from Foundations of Signal Processing
  - a
- What I learned:
  - Z transform, Laplace transform
  - Localization (frequency-time domain trade-off)
  - Approximation techniques
    - Least-squares, Langrange, Taylor Series, Chebyshev

June 13

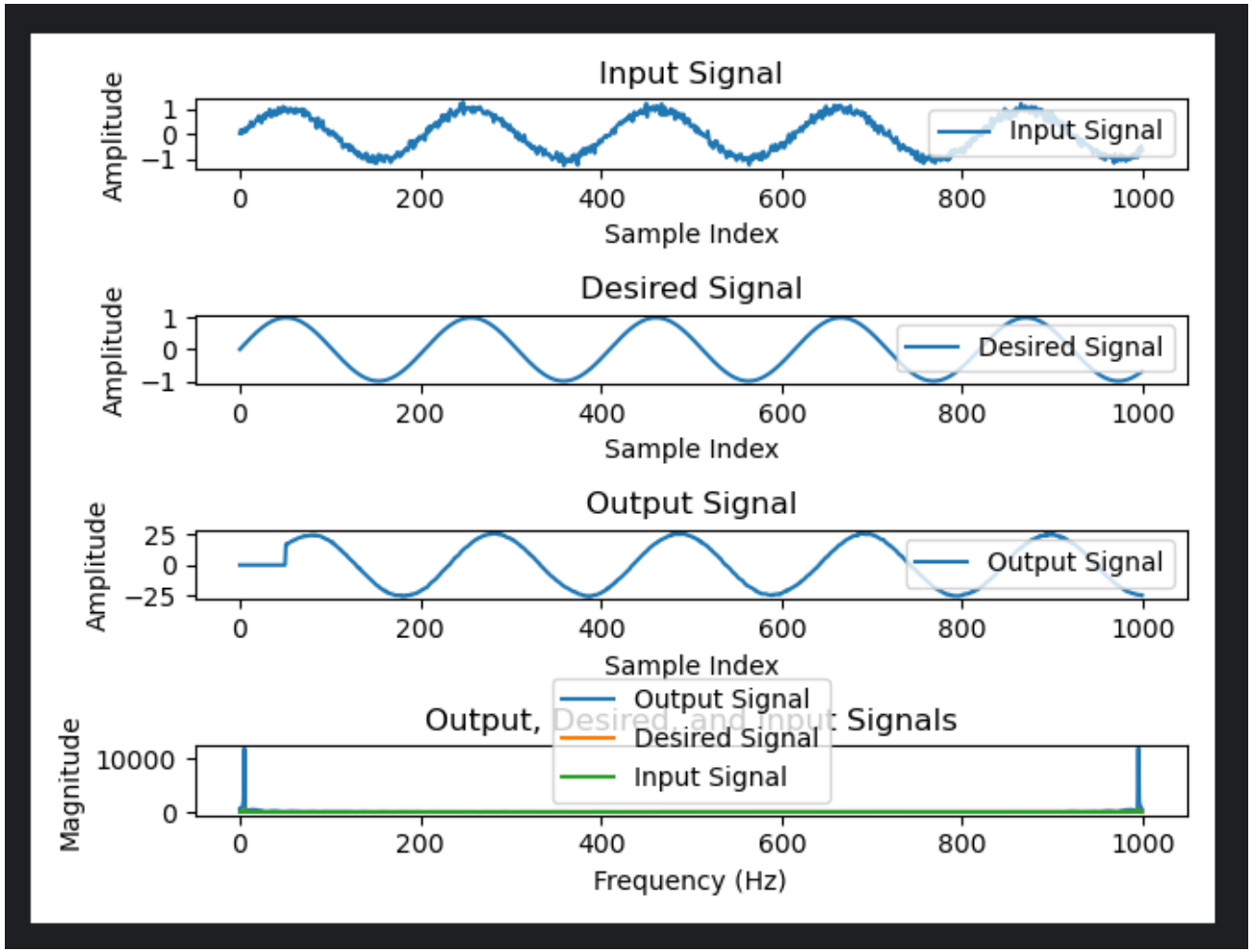
- Goals:
  - Make filter use complex numbers
  - Make filter digitized
  - Rewrite filter using less abstract operations (add, sub, few mult., etc.)
- What I learned:
  - "Dumbing down" a filter
  - Digitization processes
    - Decimate the coefficients based on the median value dynamically

June 14

- Goals:
  - Learn about FPGA filter design
  - FIR filter design in FPGAs, how it compares to other implementations
  - Code in Verilog
  - Figure out why my output is shifted
- What I learned:

test2.py

- Digitized, does not take complex numbers (yet).



Interpreting Fourier Transform Plots

- Axes:
  - X-axis: frequency
  - Y-axis: magnitude/amplitude, or phase
- **Identify the Peaks**: Look for the prominent peaks in the magnitude plot. These peaks correspond to the main frequency components of your signal.
- **Check the Frequency Values**: Note the frequencies at which these peaks occur. This tells you the periodicity in the original signal.
- **Magnitude of Peaks**: Assess the height of these peaks to understand the relative strength of each frequency component.
- **Look for Noise**: A broad, spread-out spectrum indicates noise. Sharp peaks indicate a more periodic signal with distinct frequencies.
- **Analyze Phase (if provided)**: Observe the phase plot if phase information is crucial for your analysis, like in communication signals or when reconstructing the signal.

Research Article

- https://ieeexplore.ieee.org/document/7533688
- Design of signal acquisition and tracking process based on multi-thread for real-time GNSS software receiver
- Authors developed a correction method that allows the use of past acquisition data for real-time signal tracking
- Correction method allowed effective real-time tracking even with delayed acquisition data

Signal Acquisition, Problem

- Requires identifying:
  - Code phase
    - The position within the PRN code sequence at a specific time.
  - Doppler frequency
    - The frequency shift in the received GNSS signal caused by the relative motion between the satellite and the receiver.
  - Visible satellites
- Problem: computationally intensive and slower compared to signal tracking. Delay poses a challenge for real-time operation in software receivers
  - Specifically, acquiring weak GNSS signals requires long signal integration times (1-10 milliseconds). Standard = 1ms.

- The corrected code phase shift (ΔΦ) is computed using the sampling rate (fs), the carrier frequency (fx), the Doppler frequency (Δf), and the time delay (tex):

$$\Delta\Phi = f_s \left(1 - \frac{\Delta f}{f_x}\right) t_{\text{ex}}$$

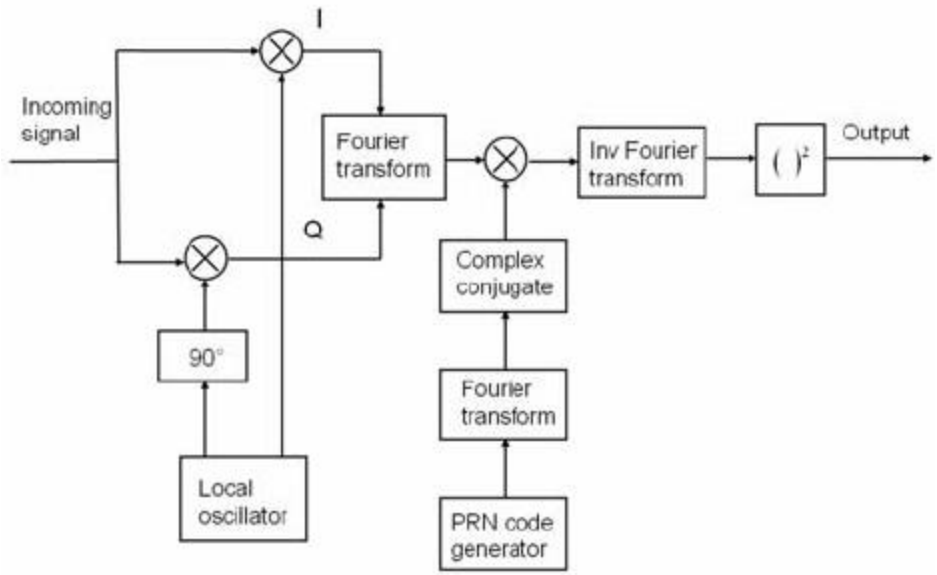
- Considering the Doppler frequency error, the corrected code phase is expressed as:

$$\Delta\Phi = f_s \left(1 - \frac{\Delta f}{f_x}\right) t_{\text{ex}} - \frac{e f_s}{f_x} t_{\text{ex}}$$

Signal Acquisition, Solution

- Solution: two-thread approach
  - **Signal Tracking**: This process runs in real-time on one thread, continuously monitoring the GNSS signal.
  - **Signal Acquisition**: This process runs on a separate thread, taking longer to complete due to the integration time required for accurate detection of weak signals.
- Synchronization - Correction Method: This method adjusts the code phase of the past acquisition results to be usable for current signal tracking. It relies on the Doppler frequency to correct the code phase, ensuring that even delayed acquisition results can be applied effectively in real-time tracking.
- Correction Method: For typical short delays (e.g., up to 10 seconds), the correction method significantly improves the accuracy of the code phase, enabling reliable signal tracking.

Parallel Code Search Acquisition Scheme



TO DO

- Purpose and scope of filters
- How does one use it?
- Verilog implementation: check with Elyas regarding cocotb
- Coroutine simulation setup
- Friday at 2pm next meeting