Channel Equalization in Digital Communication Systems

Kananelo Chabeli
Department of Electrical Engineering
University of Cape Town
Cape Town, South Africa
chbkan001@myuct.ac.za

***Abstract*—Channel equalization is the process for mitigating distortions introduced by digital channels with the intent of improving the transmission performance. The major challenges facing digital communication systems are inter-symbol interference and multi-path interference. This paper presents and implementation of adaptive channel equalizer and simulates it on software to validate its performance. The results show that the overall bit error rate and speech quality of the equalized output at the receiver are much better than that of the unequalized.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Digital communication systems have significantly outperformed their analog counterparts in terms of efficiency and reliability for data transmission,[2]. However, as the demand for higher data rates escalates, these systems face challenges such as signal distortion, increased bit-error rate(BER),and reduced signal-to-noise ratio mainly because of inter-symbol interference (ISI). ISI arises significantly from the band-limited nature of a digital channel and, in context of cellular networks, from multi-path propagation.[2],[4]. ISI is defined as the phenomenon in which signals (symbols) are smeared out such that pulses belonging to different symbols are indistinguishable[3]

Channel equalizers play a pivotal role in this context by counteracting the channel distortions and recovering the transmitted information. Ideally, the impulse response of the channel equalizer system should be an inverse of the channel frequency response, ensuring that cascade effect of channel and equalizer results in the system function of z^{δ}, where δ is some delay, but effectively restoring signal clarity [3].

This study investigates the impact of an adaptive equalizer on signals transmitted through noisy, linear, and time-invariant digital channels. It is being hypothesized that applying an adaptive equalizer reduces the bit error rate,and improves speech quality of audio effectively mitigating channel distortion effects.

The rest of the paper is organised as follows: section II presents a review of the literature in digital communications and channel equalization. Next, section III provides the method of investigation taken in the project. It highlights the procedure followed, simulation tools and how the system was tested to verify the proposed hypothesis. In section IV the overall system design is presented together with detailed description of adaptive algorithm used. Furthermore, sections V and VI presents the simulation results and their analysis. The last section draws conclusions based on the results found.

II. Literature Review

A. Introduction to Literature Review

In this section, a brief review of the literature on channel equalization and digital communication systems is provided. The section starts by looking into digital systems, the pros and cons, and then looks in to equalizer systems that have been proposed in literature.

B. Digital Communication Channels

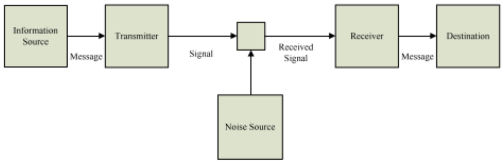
A typical communication channel consists of a transmitter, channel and a receiver. The purpose of a digital communication system is to transmit data or information from the transmitter to the receiver at lowest possible latency . Factors that need to be considered when designing communication systems include type of signals to be transmitted(analog or digital), receive power, and error performance [1].

1) Digital Channel Models: The most widely used channel models in literature are *additional noise channel* and *band-limited FIR filter channel* models [7]. An additive channel is one which is completely modelled by white Gaussian Noise. On the other-hand, filter channel is one modelled with a filter, usually low-pass FIR filter added to additive Noise. These models are shown in the fig.1.

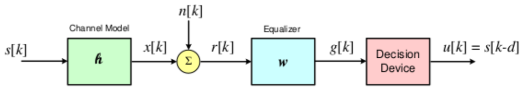
2) Challenges facing Digital Communication Systems: The most commonly occurring challenges are inter-symbol interference and non-linearity of physical channels. To get a sense of what ISI is, consider a digital signal $x[n]$ transmitted over a digital channel characterised by a linear finite-impulse response filter $h[n]$ and additive noise $v[n]$. the received signal,$r[n]$ is given by $r[n] = h[n] * x[n] + v[n]$. which can be written as:

$$r[n] = x[n] * h[0] + \Sigma_{k=-\infty,k\neq 0}^{\infty} h[k]x[n-k] + v[n] \quad (1)$$

The term $\Sigma_{k=-\infty,k\neq 0}^{\infty} h[k]x[n-k]$, is is the ISI [5].



(a) Additive noise channel model [1].



(b) Filter Channel Model

Fig. 1: Common Channel Models

C. Channel Equalization

1) Zero-forcing Channel Equalizer: The Zero-Forcing (ZFE) equalizer stands as one of the simplest and earliest channel equalization techniques, characterized by its approach of inversely mirroring the communication channel's frequency response to counteract distortions [1]. This is achieved by transmitting a noise-like training sequence rich in spectral content through the channel, thereby learning its impulse response to accurately derive the Finite Impulse Response (FIR) of the equalizer [1][7].Despite its foundational status, the ZFE equalizer harbors significant drawbacks, including attenuation of signals at higher frequencies and a tendency to amplify channel noise due to its disregard for additive noise present in the channel [1].

2) Adaptive Equalization Techniques: An adaptive equalizer combats inter-symbol interference by dynamically learning channel characteristics and adjusting its filter coefficients. It comprises two phases: the filtering process, which cleans the corrupted input signal, and the adaptive process, adjusting coefficients to minimize a cost function. This facilitates two operational modes: training mode, utilizing a known signal to learn channel characteristics, and decision-directed mode, applying adjustments for current conditions, thus adapting to the channel's changing nature [5] [3] [6]. Common adaptive equalizers are the Least Mean Square (LMS) and Recursive Least Squares (RLS) algorithms. LMS minimizes mean square error between desired and actual outputs, making it easy to implement but may have slow convergence. Conversely, RLS accelerates convergence by including historical error values in its computation, enhancing efficiency.

III. Methodology

A. System Design Choices

The design process commenced with the selection of an appropriate equalization algorithm. The Least Mean Square (LMS) Adaptive equalizer was chosen for its simplicity, facilitating a detailed analysis of the hypothesis metrics.

For the implementation and simulation phase, the selection of suitable tools was critical. Jupyter Notebook was selected due to its flexibility, notably its feature that allows executing individual code cells. Python, known for its extensive range of signal processing packages, was utilized within the Jupyter Notebook environment to enhance the development and testing phases.

B. System Implementation Procedure

The implementation began with the modeling of the digital channel, which was represented as a Finite Impulse Response (FIR) filter with additive zero-mean noise. To enable the easy variation of noise intensity, the noise variance was kept adjustable. The channel coefficients were generated using the pyFDA tool, chosen for its ease of use and availability at no cost.

The simulation was structured into two stages to efficiently gather results:

Training Stage: This involved transmitting a known signal at the receiver to learn the channel characteristics, allowing for the proper adjustment of the equalizer coefficients.

Decision-Directed Stage: Following the adjustment of the equalizer coefficients, this stage entailed applying the channel to various signals and subsequently measuring the resulting bit error rates and Perceptual Evaluation of Speech Quality (PESQ) scores.

C. Approaches of System Validation

Validation of the system was performed using two principal metrics: Bit-Error Rates (BER) and Perceptual Evaluation of Speech Quality (PESQ) for the transmitted audio signals. The BER was calculated by comparing the bit representations of transmitted, received, and equalized signals under different channel conditions, achieved by altering the noise variance.

For the PESQ evaluation, the pesq Python package was employed to compute scores for the audio signals. This approach provided a quantitative assessment of the speech quality as perceived by listeners, further affirming the system's effectiveness in practical scenarios.

IV. System Design

A. Introduction

This section presents the detailed system design specifications and implementation of the algorithm. A thorough derivation of the LMS is also presented. The sections starts by specifying the metrics tested by the project as per the hypothesis. Next process of simulation set-up and some of the python functions used are presented.

B. System Performance Metrics

The system performance metrics, derived directly from he hypothesis are as follows:

- **Bit-Error-Rate(BER):** Computed by representing signals in their raw bits and counting the number of corresponding bits that are not equal and dividing by total bits.
- **Speech Quality:** Evaluated by using the Perceptual Evaluation of Speech Quality(PESQ). This as achieved by using python package **pesq**. The function takes in

reference and degraded audios and return the speech quality score.

C. Channel Model

The digital channel used for simulating the Least Mean Square adaptive algorithm is modelled as a low-pass FIR filter with additive zero-mean white Gaussian noise. The low-pass response was chosen specifically to reflect the band-limited nature of practical communication channels. The following are the filter specification for FIR filter used:

- Must be minimum-order low-pass FIR filter designed using equiripple algorithm.
- Must have cut-off frequency of $\omega_c = 1/3\pi$, $0.1dB$ ripple for $\omega < 0.8\omega_c$ and at least $30dB$ attenuation for $\omega > 1.1\omega_c$.

The filter coefficients were generated using pyFDA tool and were fixed for the entire simulation. Channel characteristics were varied through changing of noise variance.

D. Least Mean Square (LMS) Algorithm

The Least Mean Square (LMS) algorithm's implementation in filter design involves adaptive adjustment of equalizer coefficients ($b_k[n]$) to minimize mean square error (MSE). Key notations include $x[n]$ for the transmitted data symbol at time nT_s, $f[n]$ for the received symbol, $b_k[n]$ for the k^{th} equalizer coefficient, and $y[n]$ for the equalizer output. The error signal is defined as $e[n] = x[n] - y[n]$ and the MSE objective function is formulated as $Mean|x[n] - y[n]|^2 = Mean|x[n] - \Sigma_{k=0}^{N} b[k]f[m-k]|^2$. To minimize MSE, the algorithm employs gradient descent, adjusting $b_k[n]$ iteratively based on the equation $b_k[n+1] = b_k[n] + \mu e[n]f[n-k]$, where μ denotes the step-size, crucial for balancing convergence speed and stability. Typically, $\mu = 0.01$ is used. This process dynamically adapts the filter coefficients to the changing channel conditions, aiming for optimal signal reconstruction. This is shown in the figure below.

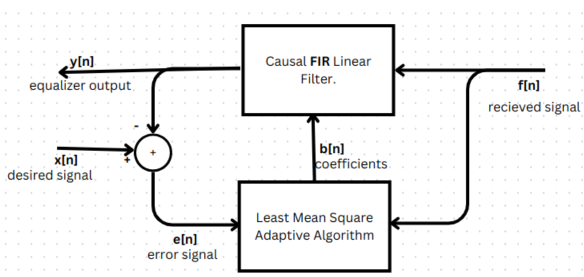


Fig. 2: LMS Equalizer Block Diagram.

E. Simulation Set-up

The system was simulated in a Jupyter notebook. Unlike conventional discrete-time signal processing, here $x[n]$ represents signal amplitude at time nTs, reflecting real-world filtering processes. This approach diverges from theory, which assumes knowledge of all signal symbols for convolution. The python function that takes in a signal data symbol and filters it is shown below:

```
def fir_filter(in_val, current_ins, fir_c):
    output = 0.0;
    if len(current_ins)!= len(fir_c):
        return 1
    #shift the current inputs:
    N= len(current_ins)
    for i in range(N-1,-1,-1):
        current_ins[i]= current_ins[i-1]
    #add thre current input at index 0
    current_ins[0] = in_val;
    #now find the output value
    for i in range(N):
        output+=current_ins[i]*fir_co[i]
    return output;
```

As the code snippet shows, the FIR filter is implemented to reflect how practical filter would be implemented in hardware. This approach gave much flexibility in adding some delays in between the symbols effectively realizing a real communication system.

F. Equalizer Implementation

The equalizer system was realized by two functions: $equalizer\_filter()$ and $update\_coefficients()$. The latter was used during training and the former during decision-directed stage. Furthermore, the function $equalizer\_filter()$, uses $fir\_filter()$ function with received symbols and learnt equalizer coefficients, passing each symbol one by one into the filter, and recording the output.

V. Simulation Results

A. Training Results

Channel FIR coefficients were generated using pyFDA, and the resulting taps are:

```
channel_coefficients =
np.array([0.014639903439961355,
-0.016192490896558002,-0.00472899759069931,
0.0610807859152884,-0.1421777303627756,
0.21559737364151998,0.7547096850970777,
0.21559737364151998,-0.1421777303627756,
0.0610807859152884,-0.00472899759069931,
-0.016192490896558002,0.014639903439961355])
```

The training signal used in this simulation was a synthesized sinusiod generated as follows:

```
n = np.arange(0,100)
training = 2*np.cos(np.pi/25*n)
+ np.sin(np.pi/4*n)
```

This sequence was used as a training sequence to learn the channel coefficients. It was 'transmitted through the channel' by applying the channel filter and adding additive noise. Figure below shows the plot of the resulting equalizer output.

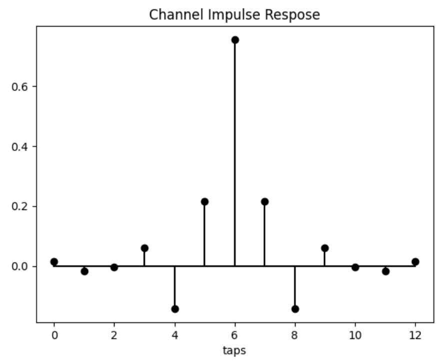


Fig. 3: Channel Impulse Response Filter

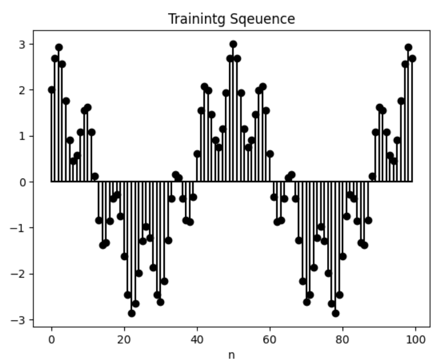


Fig. 4: Plot of the training Sequence.

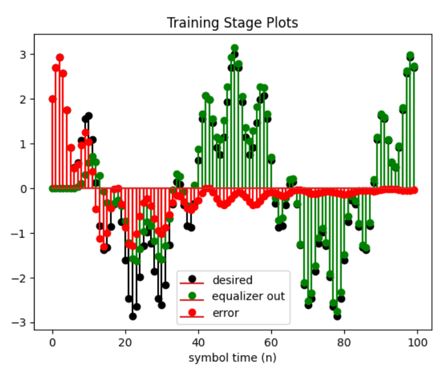


Fig. 5: Plot of signals obtained from the training

These coefficients were then used to filter a test signal passed through the channel.

B. Testing Results

For testing stage, synthesized and real audio signals were used.

1) Synthesized Test Signal Results: The synthesized test signal was a sinusoidal of different frequency. This was generated by the code:

```
test1 = 3*np.cos(np.pi/60*n)+
4*np.sin(np.pi/10*n)
```

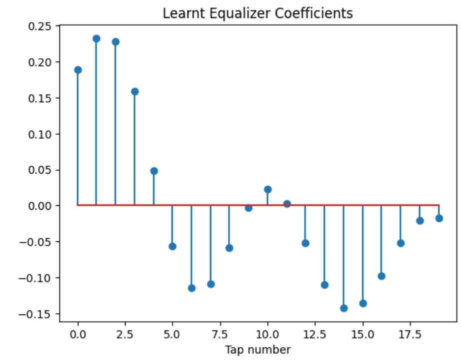


Fig. 6: Channel Coefficient Plot learnt by LSM algorithm

.

channel filter were applied to this signal and additive noise, with same statistical properties as for training stage, were added to obtain the channel output. This output were then filtered with the 'learnt' equalizer coefficients and the figure .7 shows the plot of the resulting signals.

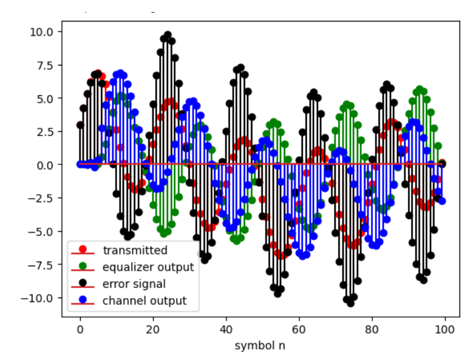


Fig. 7: Plot of signals obtained from training sequence.

2) Real Audio Signal Results: To get an approximate analysis of how the system performs with real signal,audios samples were used . The plot of the simulation with one of the audios is shown below.

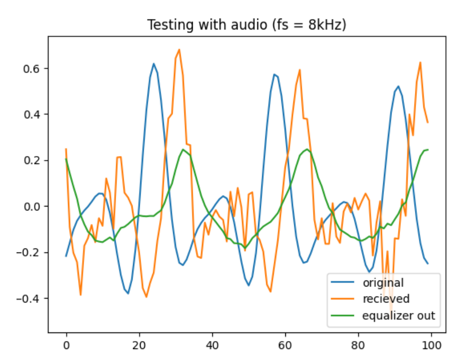


Fig. 8: Testing Results with Audio(zoomed in)

C. System Performance Results

The system performance was measured using Bit-Error Rate and Perceptual Evaluation of Speech Quality. The simulation was run several times with different channel characteristics(different noise variance), and the BER and PESQ were measured in each iteration.

TABLE I: BER data for Synthesized Signals

Noise Standard Deviation	Equalized BER	Unequalized BER
0.001	0.486	0.525
0.05	0.498	0.519
0.1	0.497	0.507
0.15	0.5	0.522
0.2	0.493	0.53
0.25	0.5	0.505
0.3	0.499	0.505
0.35	0.506	0.509
0.4	0.494	0.511
0.45	0.509	0.524
0.5	0.506	0.507

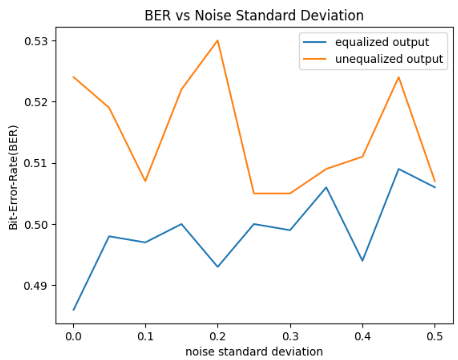


Fig. 9: Noise standard deviation vs BER for synthesized test signals.

1) BER Performance Test:

2) PESQ Performance Test : To quantify how the system performs with real audio, audio signals were transmitted and equalized with the coefficients estimated during training stage. Python package **pesq** were used to obtain PESQ score for equalized and equalized outputs and these are recorded in the table below:

TABLE II: PESQ scores for audio data

Noise Standard Deviation	Equalized PESQ	Unequalized PESQ
0.001	4.401	4.081
0.05	1.98	1.54
0.1	1.58	1.316
0.15	1.449	1.271
0.2	1.37	1.236
0.25	1.284	1.202
0.3	1.283	1.199
0.35	1.269	1.188
0.4	1.268	1.193
0.45	1.258	1.181

VI. Analysis of Results

This paper describes the development and implementation of an adaptive channel equalizer aimed at mitigating the effects of noise and inter-symbol interference. The equalizer's effectiveness was evaluated using a synthesized sinusoidal

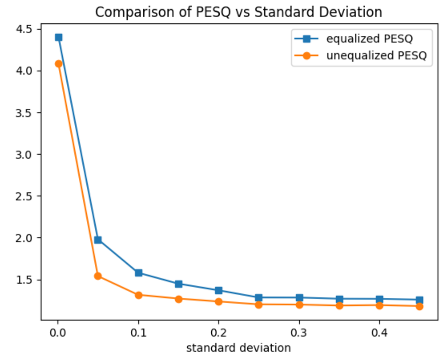


Fig. 10: PESQ Scores of Equalized and Unequalized audio outputs

signal, as shown in Fig.4. The channel outputs and the corresponding errors are depicted in Fig.7. As illustrated by these plots, the discrepancy between the desired signal and the equalized output significantly diminishes, demonstrating the system's capability to reduce channel distortions on the transmitted signal effectively.
Moreover, a channel was applied to a test signal and the results are shown in fig.7. These plots reflect that the system error is high for a different signal.

Furthermore, it was hypothesized that the equalizer system could lower the bit error rate (BER) and improve the quality of audio signals. This was examined by comparing the BER of channel outputs, with and without equalization, under various noise conditions. The resulting BERs, as a function of noise standard deviation, are tabulated in Table I and illustrated in Fig.9. The figure clearly shows that the BER for the equalized output is consistently lower than that of the unequalized output. Additionally, Fig.9 indicates that the overall BER of the output escalates with an increase in noise standard deviation, which aligns with expectations: higher standard deviation implies more noise, causing more distortions and, consequently, an increase in the overall BER.

The system was also tested using real audio signals, with the Perceptual Evaluation of Speech Quality (PESQ) scores of both equalized and unequalized outputs recorded in Table II and illustrated in Fig.10. These results further corroborate the efficacy of the equalizer, as indicated by the higher PESQ scores for the equalized outputs compared to those of the unequalized signals. The findings also reveal an exponential decline in PESQ scores with an increase in noise level, reinforcing the negative impact of noise on signal quality.

VII. Conclusions

This project looked into implementing digital equalizer system to combat the effect inter-symbol interference and noise in digital communication. The hypothesis investigated was that application of equalizer systems at the receiver of communication system could reduce bit-error rate and improve speech quality for voice communication. The report detailed the design procedure of the system and presented simulation results. From these results, it has been shown that equalizing the channel at the receiver reduces bit-error-rate and improves speech quality. However, adaptive system realized has shown to over-fit the training sequence resulting is significantly huge test errors. Also, the overall system performed poorly when transmitting audio signals. These limitations of system call for further improvements. The system can be improved by incorporating modulation techniques, so that both training and test signal have common format.

References

[1] Kazi Muhammed et al. "Channel Equalization in Digital Transmission". In: *International Journal of Computer Science and Information Security* 4.1 2 (2009).

[2] Liaqat Mahmood et al. "Adaptive Filtering Algorithms for Channel Equalization in Wireless Communication". In: *Indian Journal of Science and Technology* 8.17 (2015). DOI: 10.17485/ijst/2015/v8i17/57805.

[3] Garima Malik and Amandeep Singh Sappal. "Adaptive equalization algorithms: an overview". In: *International journal of advanced computer science and applications* 2.3 (2011).

[4] Praneet Soni Agya Mishra. "Kalman Filter Based Channel Equalizer: A Literature Review". In: *International Journal of Eme rging Technology and Advanced Engineering* 4 (5 2014), pp. 165–173.

[5] Cristiano M Panazio et al. "Channel equalization techniques for wireless communications systems". In: *Optimizing Wireless Communication Systems*. Springer, 2009, pp. 311–352.

[6] Jalali Sammuel. "Wireless Channel Equalization in Digital Communication Systems". In: *Claremont, Claremont Graduate University Theses and Disseratations, California* (2012).

[7] Bernard Sklar. "Defining, designing, and evaluating digital communication systems". In: *IEEE Communications magazine* 31.11 (1993), pp. 91–101.