BER Performance of Different Equalizers

B.Tech. Lab Project Report on

WIRELESS COMMUNICATIONS(19EC3016)

Name: V Lavanya

ID : 190040550

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

KONERU LAKSHMIAH EDUCATION FOUNDATION

VADDESWARAM, GUNTUR

June, 2021

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Project Report entitled "**BER Performance of Different Equalizers**", submitted by **V Lavanya** bearing with **ID No.190040550** in partial fulfilment of the requirements for the award of the Bachelor of Technology in Electronics & Communication Engineering and submitted to the Department of Electronics & Communication Engineering of KLEF, Vaddeswaram, Guntur is an authentic record of my own work carried out during a period from June 2021 to November 2021.

Signature of Candidate

V Lavanya

ACKNOWLEDGEMENT

It is great pleasure for me to express my gratitude to our honourable President Sri. Koneru Satyanarayana, for giving the opportunity and platform with facilities in accomplishing the project-based laboratory report.

I express the sincere gratitude to our principal **Dr. K SUBBA RAO** for his administration towards our academic growth.

I express sincere gratitude to our Coordinator **Dr. Vipul Agrawal** for his leadership and constant motivation provided in successful completion of our academic semester.

I record it as my privilege to deeply thank our pioneer **Dr. M. Suman HOD** for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor for his novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Sincerely,

V Lavanya

190040550

ABSTRACT

This Project shows the BER performance of several types of equalizers in a static channel with a null in the passband. The example constructs and implements a linear equalizer object and a decision feedback equalizer (DFE) object. It also initializes and invokes a maximum likelihood sequence estimation (MLSE) equalizer. The MLSE equalizer is first invoked with perfect channel knowledge, then with a straightforward but imperfect channel estimation technique.

As the simulation progresses, it updates a BER plot for comparative analysis between the equalization methods. It also shows the signal spectra of the linearly equalized and DFE equalized signals. It also shows the relative burstiness of the errors, indicating that at low BERs, both the MLSE algorithm and the DFE algorithm suffer from error bursts. In particular, the DFE error performance is burstier with detected bits fed back than with correct bits fed back. Finally, during the "imperfect" MLSE portion of the simulation, it shows and dynamically updates the estimated channel response.

TABLE OF CONTENTS

INTRODUCTION

BER performance of several types of equalizers in a static channel with a null in the passband. The example constructs and implements a linear equalizer object and a decision feedback equalizer (DFE) object. It also initializes and invokes a maximum likelihood sequence estimation (MLSE) equalizer. The MLSE equalizer is first invoked with perfect channel knowledge, then with a straightforward but imperfect channel estimation technique.

As the simulation progresses, it updates a BER plot for comparative analysis between the equalization methods. It also shows the signal spectra of the linearly equalized and DFE equalized signals. It also shows the relative burstiness of the errors, indicating that at low BERs, both the MLSE algorithm and the DFE algorithm suffer from error bursts. In particular, the DFE error performance is burstier with detected bits fed back than with correct bits fed back. Finally, during the "imperfect" MLSE portion of the simulation, it shows and dynamically updates the estimated channel response.

OBJECTIVE

Comparing the BER Performance of Different Equilizers.

Software Used:

Matlab Software(2021 Version)

Theory:

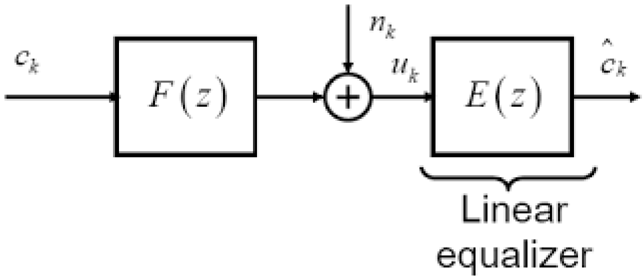
Linear Equalizer:

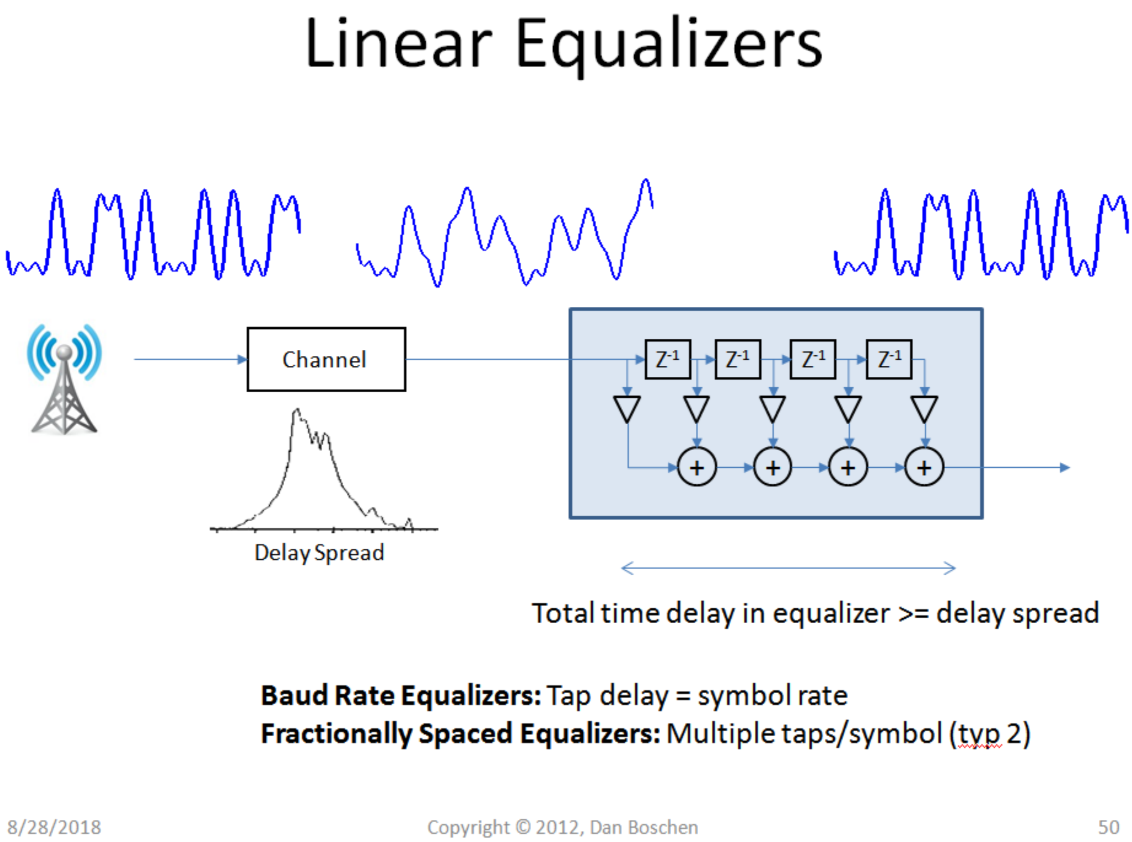
If the reconstructed Signal d(t) is not used in the feedback path to adapt the equilizer then the equalization is linear.

The equalizer is a device that attempts to reverse the distortion incurred by a signal transmitted through a channel. In digital communication its purpose is to reduce inter symbol interference to allow recovery of the transmit symbols

Linear Equalizer:

It processes the incoming signal with a linear filter.

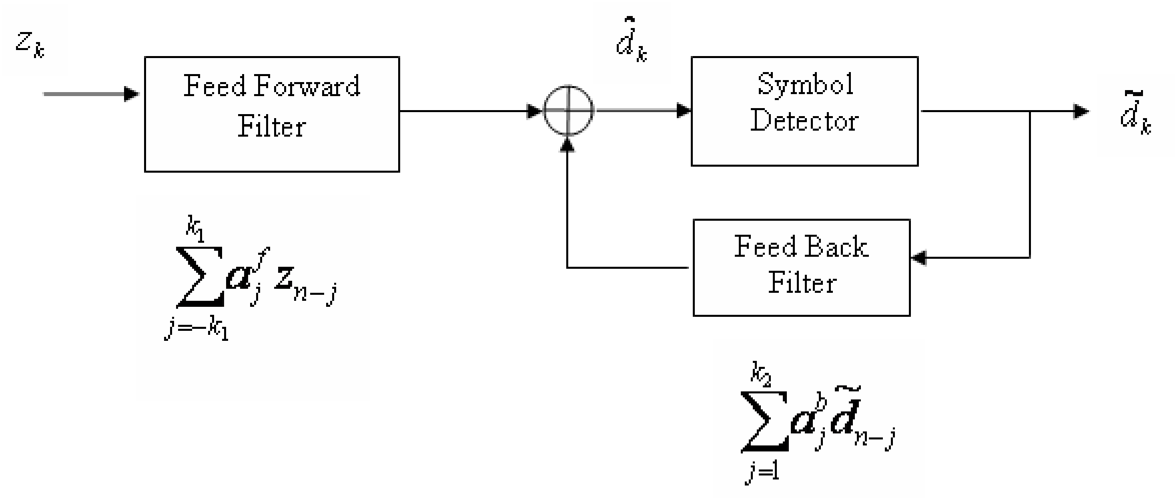




Decision FeedBack Equalizer:

The Decision Feedback Equalizer (DFE) is a form of non-linear equalization which relies on decisions about the levels of previous symbols (high/low) to correct the current symbol.

ts main advantage over linear equalizers is the ability to cancel inter-symbol interference (ISI) without amplifying the noise.



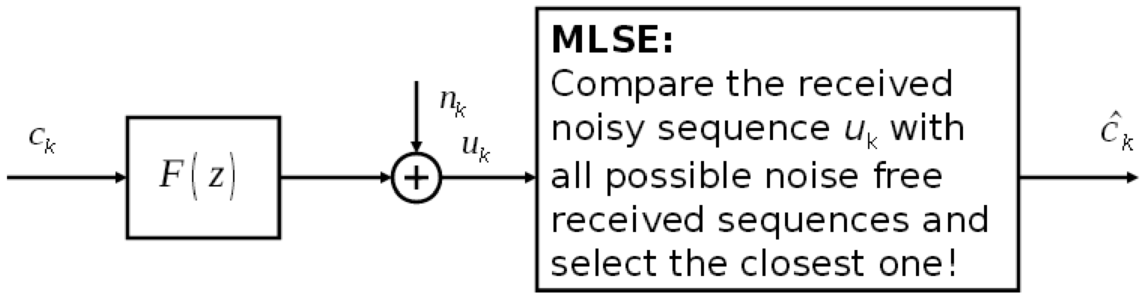
MLSE Equalizer:

Maximum-Likelihood Sequence Estimation (MLSE) equalizers provide optimal equalization of time variations in the propagation channel characteristics. However, MLSE equalizers are sometimes less appealing because their computational complexity is higher than Adaptive Equalizers.

To decode a received signal, the MLSE equalizer:

1. Applies the FIR filter to the symbols in the input signal. The FIR filter tap weights correspond to the channel estimate.
2. Uses the Viterbi algorithm to compute the traceback paths and the state metric. These values are assigned to the symbols at each step of the Viterbi algorithm. The metrics are based on Euclidean distance.
3. Outputs the maximum likelihood sequence estimate of the signal as a sequence of complex numbers corresponding to the constellation points of the modulated signal.

An MLSE equalizer yields the best theoretically possible performance, but is computationally intensive.



METHODOLOGY

Code Structure:

functions to perform link simulations over a range of Eb/No values.

eqber_adaptive.m - a script that runs link simulations for linear and DFE equalizers.

eqber_mlse.m - a script that runs link simulations for ideal and imperfect MLSE equalizers

eqber_siggen.m - a script that generates a binary phase shift keying (BPSK) signal with no pulse shaping, then processes it through the channel and adds noise.

eqber_graphics.m - a function that generates and updates plots showing the performance of the linear, DFE, and MLSE equalizers.

The scripts eqber_adaptive and eqber_mlse illustrate how to use adaptive and MLSE equalizers across multiple blocks of data such that state information is retained between data blocks.

ALGORITHM:

1. Set parameters related to the signal and channel. Use BPSK without any pulse shaping, and a 5-tap real-valued symmetric channel impulse response.

2. Set initial states of data and noise generators. Set the Eb/No range.

3. Set parameter values for the linear and DFE equalizers. Use a 31-tap linear equalizer, and a DFE with 15 feedforward and feedback taps.

4 .Use the recursive least squares (RLS) algorithm for the first block of data to Ensure rapid tap convergence. Use the least mean square (LMS) algorithm thereafter to ensure rapid execution speed.

5. Set the parameters of the MLSE equalizer. Use a traceback length of six times the length of the channel impulse response. Initialize the equalizer states. Set the equalization mode to "continuous", to enable seamless equalization over multiple blocks of data.

7. The RLS update algorithm is used to adapt the equalizer tap weights and reference tap is set to center tap.

8. Run the all equalizers, and plot the equalized signal spectrum, the BER, and the burst error performance for each data block. Note that as the Eb/No increases, the linearly equalized signal spectrum has a progressively deeper null.

CODE

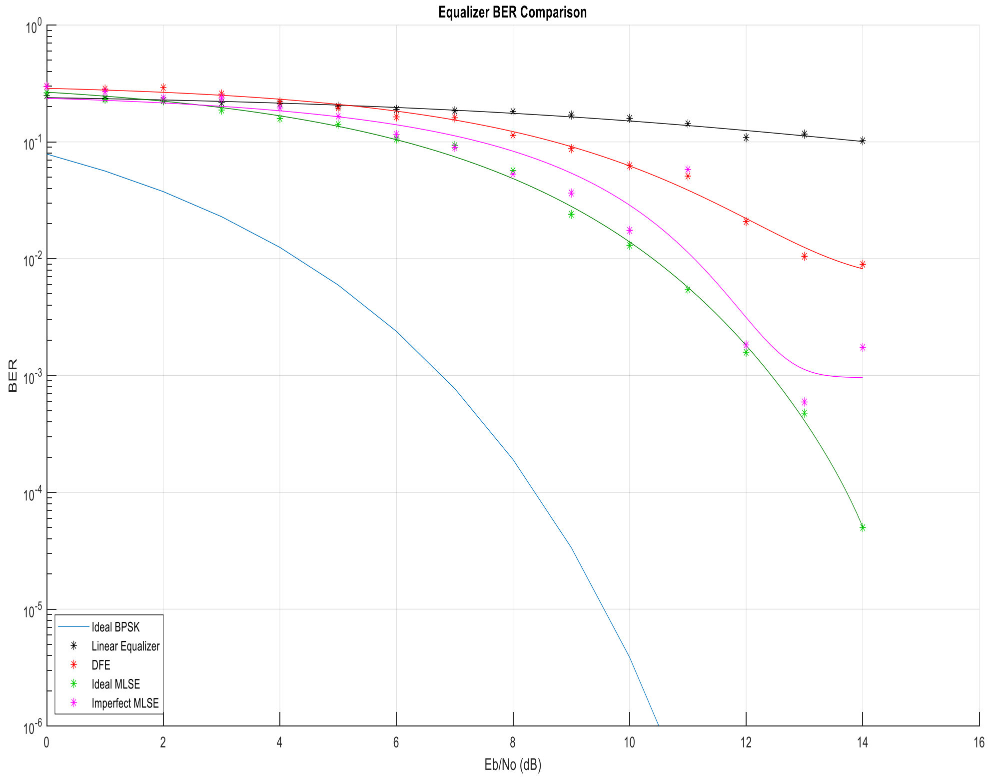
```
% System simulation parameters
Fs = 1;             % sampling frequency (notional)
nBits = 2048;       % number of BPSK symbols per vector
maxErrs = 200;      % target number of errors at each Eb/No
maxBits = 1e6;      % maximum number of symbols at each
Eb/No

% Modulated signal parameters
M = 2;                        % order of modulation
Rs = Fs;                      % symbol rate
nSamp = Fs/Rs;                % samples per symbol
Rb = Rs*log2(M);              % bit rate

% Channel parameters
chnl = [0.227 0.460 0.688 0.460 0.227]';  % channel
impulse response
chnlLen = length(chnl);                   % channel
length, in samples
EbNo = 0:14;                              % in dB
BER = zeros(size(EbNo));                  % initialize
values

% Create BPSK modulator
bpskMod = comm.BPSKModulator;

% Specify a seed for the random number generators to
ensure repeatability.
rng(12345)
% Linear equalizer parameters
nWts = 31;                  % number of weights
algType = 'RLS';            % RLS algorithm
forgetFactor = 0.999999; % parameter of RLS algorithm

% DFE parameters - use same update algorithms as linear
equalizer
nFwdWts = 15;               % number of feedforward weights
nFbkWts = 15;               % number of feedback weights
% MLSE equalizer parameters
tbLen = 30;                           % MLSE equalizer
traceback length
numStates = M^(chnlLen-1);            % number of trellis
states
[mlseMetric,mlseStates,mlseInputs] = deal([]);
const = constellation(bpskMod);       % signal constellation
mlseType = 'ideal';                   % perfect channel
estimates at first
```

```
mlseMode = 'cont';                    % no MLSE resets

% Channel estimation parameters
chnlEst = chnl;           % perfect estimation initially
prefixLen = 2*chnlLen;    % cyclic prefix length
excessEst = 1;            % length of estimated channel
impulse response
% beyond the true length

% Initialize the graphics for the simulation. Plot the
unequalized channel
% frequency response, and the BER of an ideal BPSK
system.
idealBER = berawgn(EbNo,'psk',M,'nondiff');

[hBER,hLegend,legendString,hLinSpec,hDfeSpec,hErrs,hText1
,hText2, ...
    hFit,hEstPlot,hFig,hLinFig,hDfeFig] =
eqber_graphics('init', ...
    chnl,EbNo,idealBER,nBits);
linEq = comm.LinearEqualizer('Algorithm',algType, ...
    'ForgettingFactor',forgetFactor, ...
    'NumTaps',nWts, ...
    'Constellation',const, ...
    'ReferenceTap',round(nWts/2), ...
    'TrainingFlagInputPort',true);

dfeEq =
comm.DecisionFeedbackEqualizer('Algorithm',algType, ...
    'ForgettingFactor',forgetFactor, ...
    'NumForwardTaps',nFwdWts, ...
    'NumFeedbackTaps',nFbkWts, ...
    'Constellation',const, ...
    'ReferenceTap',round(nFwdWts/2), ...
    'TrainingFlagInputPort',true);
firstRun = true;  % flag to ensure known initial states
for noise and data
eqType = 'linear';
eqber_adaptive;
close(hFig(ishghandle(hFig)));

eqType = 'dfe';
eqber_adaptive;
close(hLinFig(ishghandle(hLinFig)),hDfeFig(ishghandle(hDf
eFig)));

eqType = 'mlse';
```

```
mlseType = 'ideal';
eqber_mlse;
mlseType = 'imperfect';
eqber_mlse;
```

OUTPUTS:

Bit Error rate of different equilizers:



Result:

The BER is calculated by comparing the transmitted sequence of bits to the received bits and counting the number of errors. The ratio of how many bits received in error over the number of total bits received is the BER.

A low SNR will have an increased BER. Put simply a strong signal is better than a weak one and has less chance of errors. The reason error increases with SNR is because of noise.

CONCLUSION

BER of different equalizers are

Linear Equalizer>DFE >MLSE>ideal BPSK

References:

https://in.mathworks.com/help/comm/ug/ber-performance-of-different-equalizers.html

END