2019-10

Impulsive Noise Mitigatin Using Turbo Coding and Hybrid Nonlinear Preprocessing In Ofdm-Plc Systems

Gebeyehu, Girma

http://hdl.handle.net/123456789/10963

Downloaded from DSpace Repository, DSpace Institution's institutional repository
BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF RESEARCH AND GRADUATE STUDIES
FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

IMPULSIVE NOISE MITIGATION USING TURBO CODING AND HYBRID NONLINEAR PREPROCESSING IN OFDM-PLC SYSTEMS

GIRMA GEBEYEHU AMLAKU

BAHIR DAR, ETHIOPIA
October 14, 2019
IMPULSIVE NOISE MITIGATION USING TURBO CODING AND HYBRID NONLINEAR PREPROCESSING IN OFDM-PLC SYSTEMS

Girma Gebeyehu Amlaku

A thesis submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree of Masters in the Communication Systems Engineering in the faculty of Electrical and Computer Engineering.

Advisor Name: Fikreselam Gared (PhD.)

BahirDar, Ethiopia
October 14, 2019
DECLARATION

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation/ fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student: Girma Gebyehu Signature ______________________
Date of submission: 14/10/2019
Place: Bahir Dar

This thesis has been submitted for examination with my approval as a university advisor.

Advisor Name: Fikreselam Gared (PhD.) ____________________
Advisor’s Signature: ______________________
Bahir Dar University
Bahir Dar Institute of Technology-
School of Research and Graduate Studies
Faculty of Electrical and Computer Engineering

THESIS APPROVAL SHEET

Girma Gebeyehu
Name

14/10/2019
Date

The following graduate faculty members certify that this student has successfully presented the necessary written final thesis and oral presentation for partial fulfillment of the thesis requirements for the Degree of Master of Science in Communication Systems Engineering.

Approved By:
Advisor:
Fikreselam Gared (PhD.)
Name

14/10/2019
Date

External Examiner:
Yihenew Wondie (PhD.)
Name

14/10/2019
Date

Internal Examiner:
Prof. Pushparaghavan Annamalai (PhD.)
Name

14/10/2019
Date

Chair Holder:
Amare Kassaw (MSc.)
Name

14/10/2019
Date

Faculty Dean:
Solomon Lulie (MSc.)
Name

14/10/2019
Date
Acknowledgments

First of all, I would like to thank my advisor Dr. Fikreselam Gared for the continued guidance and constructive comments he gave me from title selection up to the completion of this thesis work. I would also like to take this opportunity to acknowledge my instructors for giving me constructive comments from proposal to second progress presentations.

Last but not least, I would also like to thank my family and friends for their continued support throughout the work.
Abstract

Power Line Communication (PLC) uses already existing powerlines to communicate data. As power lines are designed to carry electric power at low frequency, they are not suited for high speed data communication. Multipath induced frequency selective fading and Impulsive Noise (IN) are the primary challenges in PLC. To reduce the effect of multipath fading and IN, Orthogonal Frequency Division Multiplexing (OFDM) is commonly used in PLC. The success of OFDM to reduce IN effects is limited to low levels of IN energy, and for higher levels of IN energy, other IN mitigation techniques need to be applied.

In this thesis, hybrid nonlinear preprocessing and Turbo coding are applied together to reduce the effect of IN in OFDM-PLC systems. Hybrid preprocessors considered are of two type; conventional hybrid preprocessing (CHP) and adaptive hybrid preprocessing (AHP). Error performances of the proposed dual mitigation techniques are studied via simulations using MATLAB.

Bit Error Rate (BER) performance of Turbo coding paired with CHP and AHP is compared with the performance of Turbo coding paired with blanking and clipping separately, taking different levels of channel impulsiveness. The obtained results indicate that the proposed IN mitigation scheme results in reduced BER compared to other previously applied combinations of Turbo coding and nonlinear preprocessing methods. It is also noted that, comparing all pairs, AHP combined with Turbo coding has the lowest BER for all channel conditions considered, and as a result the performance of OFDM-PLC systems would be better.

Key Words: AHP, Blanking, CHP, Clipping, IN, OFDM, PLC, Turbo coding
# Table of Contents

Declaration ............................................................... i

Acknowledgments ......................................................... iv

Abstract ........................................................................... v

List of Figures .............................................................. viii

List of Tables .................................................................... x

Abbreviations .................................................................... xi

1 Introduction .................................................................... 1
  1.1 Overview of Power Line Communication (PLC) ................. 1
  1.2 Impulsive Noise (IN) .................................................. 2
  1.3 Noise modeling in PLC .............................................. 4
    1.3.1 Poisson noise model ............................................ 4
    1.3.2 Middleton Class A noise model ......................... 5
    1.3.3 Bernoulli-Gaussian noise model ....................... 6
  1.4 IN mitigation techniques ........................................... 8
  1.5 Multipath effect in PLC ............................................ 8
  1.6 Statement of the problem ......................................... 10
  1.7 Objective .................................................................. 11
    1.7.1 General objective ............................................... 11
    1.7.2 Specific objectives ............................................. 11
  1.8 Scope of the Study ................................................... 11
  1.9 Significance of the Study .......................................... 12
  1.10 Methodology .......................................................... 13
  1.11 Organization of the thesis ....................................... 15

2 Literature Review ....................................................... 16

3 Proposed System Model ............................................... 20
  3.1 Orthogonal frequency division multiplexing (OFDM) in PLC .. 21
  3.2 Nonlinear preprocessing ........................................... 28
# Table of Contents

3.2.1 Hybrid nonlinear preprocessing ........................................... 30  
3.2.2 Signal-to-Noise ratio at the output of nonlinear preprocessor 32  
3.3 Turbo codes ................................................................. 33 
  3.3.1 Turbo Encoding ......................................................... 34 
  3.3.2 Trellis Termination .................................................... 37 
  3.3.3 UMTS Turbo code ...................................................... 37 
  3.3.4 Turbo Decoding ......................................................... 39 

4 Results and Discussion .......................................................... 44 
  4.1 Simulation parameters .................................................... 45 
  4.2 Impulsive noise and its model .......................................... 46 
  4.3 Error performance of PLC systems with single and multicarrier modulation ......................................................... 49 
  4.4 Optimization of preprocessor thresholds .............................. 51 
  4.5 Performance analysis of different nonlinear preprocessors for IN mitigation ......................................................... 53 
  4.6 Performance analysis of Turbo coding for IN mitigation ........... 59 
  4.7 Performance analysis of Turbo coding plus nonlinear preprocessing for IN mitigation ......................................................... 61 
  4.8 Performance analysis of Turbo coding plus AHP considering multi-path effect ......................................................... 65 

5 Conclusion and Recommendations ............................................. 68 
  5.1 Conclusion ................................................................. 68 
  5.2 Recommendations for future work ....................................... 69 

References ................................................................................. 71
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Powerline network layout in customer premise [1].</td>
<td>2</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>Noise waveform by a CRT TV [1].</td>
<td>3</td>
</tr>
<tr>
<td>Figure 1.3</td>
<td>Impulsive index ($A$) of three impulses [2].</td>
<td>6</td>
</tr>
<tr>
<td>Figure 1.4</td>
<td>Flow chart approach for overall thesis work.</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>OFDM-PLC system with IN mitigation.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Block diagram of a communication system employing OFDM [3]</td>
<td>22</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>OFDM modulation conceptual diagram [4].</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Three orthogonal subcarriers over one symbol period [4].</td>
<td>25</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Bandwidth saving achieved by OFDM modulation: (a) Conventional FDM and (b) OFDM. [4]</td>
<td>26</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Spectra of multiple OFDM subcarriers showing the orthogonality between them [3].</td>
<td>27</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>Cyclic prefix of an OFDM symbol [4].</td>
<td>27</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Block diagram of OFDM transmission with nonlinear preprocessing [5].</td>
<td>30</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>Generic block diagram of Turbo Encoders [6].</td>
<td>35</td>
</tr>
<tr>
<td>Figure 3.10</td>
<td>A block interleaver [7].</td>
<td>35</td>
</tr>
<tr>
<td>Figure 3.11</td>
<td>Recursive systematic convolutional (RSC) encoder of rate 1/2 [6].</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3.12</td>
<td>Trellis termination of RSC encoder [6].</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.13</td>
<td>UMTS Turbo encoder [6].</td>
<td>38</td>
</tr>
<tr>
<td>Figure 3.14</td>
<td>Turbo decoder [8].</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>IN samples generated using MATLAB according to MCA noise model for different levels of channel impulsiveness.</td>
<td>48</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>PDF plot of IN generated according to MCA noise model.</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Error performance of PLC systems with single carrier and multicarrier modulation schemes.</td>
<td>50</td>
</tr>
</tbody>
</table>
Figure 4.4  Output SNR of blanking, clipping and conventional hybrid pre-processors as a function of threshold values. ($\Gamma = A = 0.1$, SNR=5dB) 52

Figure 4.5  Output SNR of blanking, clipping and conventional hybrid pre-processors as a function of threshold values. ($\Gamma = A = 0.1$, SNR=15dB) 53

Figure 4.6  Output SNR of adaptive hybrid preprocessor as a function of threshold values. ($\Gamma = A = 0.1$, SNR=5dB) 54

Figure 4.7  BER of an OFDM-PLC system with preprocessors for IN mitigation as a function of impulsive index $A$. ($\Gamma = 0.01$ and SNR=20dB) 55

Figure 4.8  BER of an OFDM-PLC system with preprocessing for varying $\Gamma$. ($A=0.1$, SNR=20dB) 57

Figure 4.9  BER performance of Turbo coded and uncoded OFDM-PLC system for different values of SNR as a function of $\Gamma$. ($A=0.1$) 59

Figure 4.10  Error performance of Turbo coded OFDM-PLC system for different values of channel impulsiveness 'A' as a function of input signal SNR. ($\Gamma = 0.01$) 61

Figure 4.11  BER versus input signal SNR of an OFDM-PLC system with different preprocessor plus Turbo coding applied for $\Gamma = 0.1$ and $A = 0.1$. 63

Figure 4.12  BER versus input signal SNR of an OFDM-PLC system with different preprocessor plus Turbo coding applied for $\Gamma = 0.1$ and $A = 0.05$. 63

Figure 4.13  BER versus input signal SNR of an OFDM-PLC system with different preprocessing plus Turbo coding applied for $\Gamma = A = 0.05$. 65

Figure 4.14  Performance of different IN mitigation schemes for OFDM-PLC systems with and without multipath. ($\Gamma = 0.1$ and $A = 0.1$) 66
List of Tables

Table 4.1  Simulation parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
Table 4.2  Parameters of four-path model of a PLC channel. . . . . . . . . . . . . . . . . . . . . 66
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHP</td>
<td>Adaptive Hybrid Preprocessing</td>
</tr>
<tr>
<td>APP</td>
<td>a posteriori Probability</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BBPLC</td>
<td>Broadband Powerline Communication</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>BG</td>
<td>Bernoulli-Gaussian</td>
</tr>
<tr>
<td>CHP</td>
<td>Conventional Hybrid Preprocessing</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>ECC</td>
<td>Error Correcting Code</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>IDFT</td>
<td>Inverse Discrete Fourier Transform</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>IN</td>
<td>Impulsive Noise</td>
</tr>
<tr>
<td>ISI</td>
<td>Intersymbol Interference</td>
</tr>
<tr>
<td>LLR</td>
<td>Log Likelihood Ratio</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a posteriori</td>
</tr>
<tr>
<td>MCA</td>
<td>Middleton Class A</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PLC</td>
<td>Power Line Communication</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal-to-Impulsive Noise Ratio</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
</tbody>
</table>
1.1 Overview of Power Line Communication (PLC)

Power Line Communication (PLC) reuses existing infrastructure (i.e., power lines whose primary purpose is the delivery of AC (50 Hz or 60 Hz) or DC electric power, for the purpose of data communication. It was started in the late 1800s and early 1900s, and the first application was remote meter reading. Frequency ranges of 15 kHz to 500 kHz with signal bandwidths of few kHz were used. Two-way PLC was developed in the 1980s for automatic meter reading and automation in the distribution grid, all applications being narrowband. In the late 1990s broadband PLC (BBPLC) services were started using frequency bands between 1.8 MHz to 25 MHz and providing data rates ranging from several Mbps to several hundred Mbps. The beginning of BBPLC was accompanied by a surge in research activities. Nowadays, PLC provides natural communications for numerous applications such as home automation, video surveillance, health care monitoring and even Internet access [1, 9–11].

Basically there are three distribution systems over which PLC is employed. The indoor voltage refers to indoor transmission systems within a customers premises. Medium voltage refers to three phase 3.3/6.6 to 11/33 kV transmission systems far from distribution transformer by few kilometers. The third group, low voltage transmission systems, involve three phase lines at few hundreds of volts. Typical values of voltage for low voltage distribution systems are 400V and 230V. Figure 1.1 shows a simplified indoor broadband PLC networking that connects home equipments to low voltage access network using data switches [1].
Even though PLC systems avoid the need to lay new cables for new services, using the power line for communicating data is not so easy due to the hostile channel environment it provides when high frequency signals are applied to it. Multipath propagation induced frequency-selective fading and additive noise (especially IN) are the main challenges. The multipath effect is caused by a number of concentrated (distributed) branches and different connected load impedances, including line length (both direct and branched lengths). The connected loads and branches result in delayed signals which interfere with the direct waves and cause inter-symbol interference (ISI) [1],[9],[2, 5, 11]. The two primary challenges in PLC and their modeling are explained in the following sections.

### 1.2 Impulsive Noise (IN)

The noise which usually exist in PLC is non-Gaussian which is relatively strong, colored and time-varying [9]. Noise in PLC is not only the common thermal noise which is present in other communication systems. Different classifications of PLC noise do exist. One way is to group the noise into two: background noise and IN. The background noise is the superposition of multiple noise types, whereas
IN, which is usually very strong compared to the background noise is the primary cause of error in PLC, and its power spectral density is at least 10-15dB higher than the background noise [5],[12]. Due to this reason, IN mitigation is an active area of research and it is the primary task of this thesis work.

IN are abrupt noise with large amplitudes and short (typically a few micro-seconds to milli-seconds) duration and random interarrival characteristic. They can be synchronous or asynchronous to the AC mains or isolated. A silicon-controlled rectifier is a typical example of an electric appliance which causes synchronous to AC mains IN. The diodes which make up the rectifier switch on and off with the direction change of electric power flow. This zero crossings at fixed intervals or fixed frequency cause the device to produce strong IN regularly. If the IN consists of a series of regular pulses but at a higher frequency than the AC mains, it is referred as asynchronous, and a switching regulator is a typical source of such a noise [9],[11].

A typical switch will result in the creation of strong impulses during switching on and off operations. This is due to the reactive effect, which tries to maintain current flow in the same direction. The switching on and off operations are not related to the frequency of the AC mains. They are controlled by the will of individuals and as a result can be pressed at any arbitrary time. So, the IN produced during manual switching is an example of isolated IN. A thermostat is another cause of isolated types of IN. Figure 1.2 shows synchronous to AC mains IN waveforms [9]. IN can occur in isolation or in burst. When they happen in

![Noise waveform by a CRT TV](image)

**Figure 1.2:** Noise waveform by a CRT TV [1].
burst they provide a channel with memory for which Markov chains are used to model them [9],[2],[13]. In this thesis, the channel studied is memoryless, so that IN models for memoryless channels only are briefly explained.

1.3 Noise modeling in PLC

To tackle noise in PLC, proper modeling is required. Commonly used noise models cannot be applied directly to PLC systems, as PLCs are different from most commonly studied communication systems, because PLCs are characterized by non-Gaussian noise in general. As a result, different theoretical and empirical models have been proposed, which are appropriate to the colored, non-Gaussian noise inherent in PLC. The following are some of the commonly applied models of noise in PLC.

1.3.1 Poisson noise model

Middleton was the first to describe the phenomenon of IN [2]. He described IN as a sequence of nonoverlapping pulses of varying duration and intensity. In time, the pulses were considered to be occurring randomly. A model for such pulse time modulation like IN is given as [2]

\[ n(t) = \sum_{i=1}^{L} a_i \delta(t - t_i), \]  

where

- \( \delta(t - t_i) \) - is the \( i^{th} \) unit (ideal) impulse, described as a delta function,
- \( a_i \) - are statistically independent random variables with identical PDFs,
- \( t_i \) - are independent random variables uniformly distributed in the time period \( T_0 \), which is the observation period of the impulses,
- \( L \) - is number of impulses in any observation period \( T_0 \), assumed to obey a Poisson distribution given in Equation 1.2.

\[ P_{T_0}(L) = \frac{(nT_0)^L e^{-(nT_0)}}{L!}, \]  

where

- \( n \) - is average number of impulses per second,
• $e$ - irrational constant approximately equal to 2.718.

The Poisson model was widely studied and applied, even if it is ideal due to the assumption that the impulses are delta functions. In addition to that this model was developed by Middleton to describe man-made INs. So this model doesn’t consider the naturally occurring IN, such as that caused by thunder storms. Later on, Middleton developed a model for both man-made and natural phenomena caused noise and classified into three groups [2].

• Class A: noise has narrower bandwidth than receiver.
• Class B: noise has larger bandwidth than receiver.
• Class C: the sum of Class A and Class B noise.

1.3.2 Middleton Class A noise model

Middleton Class A (MCA) noise model is the most widely accepted and applied model to study IN. In Poisson noise model, the pulses are represented by delta functions which are ideal. In MCA noise model, which is still a form of Poisson noise model, the pulses are no more represented by ideal delta functions. Their width is considered to be nonzero, which is more realistic. The width of the pulses is taken into consideration, and the PDF for a noise sample is given as [2],[14–16]

\[
p(X) = \sum_{m=0}^{\infty} \frac{e^{-A}A^m}{m!} \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{|X|^2}{2\sigma_m^2}\right),
\]

(1.3)

Where the variance $\sigma_m^2$ is given by

\[
\sigma_m^2 = \sigma_u^2 \left( \frac{m}{A + \Gamma} \right),
\]

(1.4)

and

\[
\sigma_u^2 = \sigma_G^2 + \sigma_I^2, \quad \Gamma = \frac{\sigma_G^2}{\sigma_I^2},
\]

(1.5)

The parameters $\sigma_G^2$ and $\sigma_I^2$ are the variances of Gaussian noise and IN, respectively. $\Gamma$ is the background to IN average power ratio, and $A$ is the impulsive index which increase the impulsive behavior as it becomes smaller and conversely the noise becomes Gaussian when $A$ is large [2].
Chapter 1. Introduction

A represents the density of a certain width pulses in an observation period and it is given as

$$ A = \frac{n \tau}{T_0} $$

(1.6)

where

- $n$ - average number of pulses in a second,
- $\tau$ - average duration of each pulse,
- $T_0$ - observation period, which is commonly set to 1 ($T_0 = 1$).

The impulsive index $A$ is always less than or equal to 1, because it is defined as a fraction of time occupied by pulses in an observation time. Even if $n \tau$ is bigger than $T_0$, the value of $A$ cannot exceed one. Figure 1.3 [2], shows a simple diagram depicting an impulsive index value which is three time the average duration of impulses.

![Figure 1.3: Impulsive index ($A$) of three impulses [2].](image)

1.3.3 Bernoulli-Gaussian noise model

Bernoulli-Gaussian (BG) is one of the most popular noise models widely adopted in the literature. Similar to MCA model, it is also a Gaussian mixture; meaning the PDF of a noise sample is a sum of weighted Gaussian PDFs. But unlike MCA model, which has infinite terms, BG is the sum of two weighted zero mean Gaussian PDFs.

The PDF of a noise sample is given as [5],[17],[18]

$$ P_{n_k}(n_k) = p_0 G(n_k, 0, \sigma_0^2) + p_1 G(n_k, 0, \sigma_1^2) $$

(1.7)

where

- $n_k$ - total noise component,
- $p$ - IN probability of occurrence,
• \( p_1 = p \),
• \( p_0 = (1-p) \),
• \( G(.) \) - Gaussian PDF,
• \( \sigma_0^2 \) - variance of Gaussian component of noise,
• \( \sigma_1^2 \) - sum of Gaussian and IN variances (total variance),

Using the BG model a sample of the total noise can be given as

\[
n_k = w_k + i_k \quad k = 0, 1, 2, ..., N - 1
\]  

(1.8)

where \( w_k \) and \( i_k = b_k g_k \) are the AWGN and IN components of the total noise, respectively. \( g_k \) is complex white Gaussian noise with mean zero and \( b_k \) is the Bernoulli process with probability \( P(r(b_k = 1)) = p \) [5].

In terms of adoption in the literature, the BG model has the upper hand. The primary reason for many researchers choosing BG over MCA is because it is more tractable, since MCA has infinite terms, where as BG has only two simple terms. So comparing the two models BG is easier to handle mathematically and simulate using computational softwares as simulating infinite terms is impossible.

But, MCA’s parameters can directly be related to the physical channel. In addition to that, MCA model’s PDF can be simplified by taking few first terms of the infinite series and still be fairly accurate. By doing so, the BG model can be approximated by MCA, which has parameters relateable to the physical channel. In [2], it was shown that the MCA PDF can be approximated by truncating the infinite series into the first few terms of the summation and still maintain sufficient accuracy. The approximate PDF for MCA noise model becomes

\[
P(X) = \sum_{m=0}^{K-1} \frac{P_m}{\sum_{m=0}^{K-1} P_m} N(X; 0, \sigma_m^2)
\]

(1.9)

where

\[
P_m = \frac{e^{-A} A^m}{m!}
\]

(1.10)

and \( N(X; 0, \sigma_m^2) \) represents a Gaussian distribution of zero mean and variance \( \sigma_m^2 \), \( k \) is the number of terms of the MCA PDF series. BER was plotted versus SNR for different values of \( K \) and different levels of impulsiveness in [2].
1.4 IN mitigation techniques

Different techniques have been proposed to combat IN on different literatures. A brief summary of these methods, in light of OFDM, can be found in [2]. IN combating methods can be grouped into three classes,

- **Nonlinear preprocessing**: basic preprocessing methods are clipping and blanking, which involve clipping and blanking or nulling of a received noise affected sample to reduce IN energy. Other preprocessors called hybrid are made by the joined application of these two. Adaptive hybrid preprocessing (AHP) and conventional hybrid preprocessing (CHP) fall under this group. Nonlinear preprocessors are used to mitigate IN in [15],[19],[20].

- **Error Correcting Coding (ECC)**: Iteratively decoded ECCs are of primary choice to combat IN. Most researches use convolutional coding, Turbo coding and others. In this thesis work, Turbo codes are applied alongside AHP and CHP. Iteratively decoded error correcting codes are applied to combat IN in [14],[21–25].

- **Iterative techniques**: Methods under this group try to estimate the IN and subtract it from the received vector. The noise estimation could be done in time or frequency domain. They refine the estimation for more iterations. So, the higher the number of iterations, the better the estimation. But, the number of iterations need to be limited as it can cause unacceptable levels of delay. Iterative techniques are used to suppress IN in [26],[27].

From the above mentioned three classes of IN mitigation techniques, AHP and CHP which are grouped under nonlinear preprocessors and binary Turbo coding from ECCs are chosen to be applied jointly to combat IN. They are chosen due to the fact that they are at the top of the list in their respective groups in terms of error reduction in communication systems. Another reason for the choice is that, as far as we know, they have never been applied together before towards this goal.

1.5 Multipath effect in PLC

PLC is similar to wireless communication when it comes to the effect of multipath [28]. Multipath effect exists because of multiple paths followed by copies of the
same signal from source to sink, which arrive at the receiver at different times. Some of them arriving very late can interfere with other signals that are arriving at the receiver even though they were transmitted at a later time. The reason for this is the length of the paths followed are different, which makes the arrival times different. This overlapping of different component signals causes fading. The signals are affected differently at different frequency values, which can be observed from the frequency response of multipath channels.

The difference with wireless channels in light of multipath is, in PLC analytical determination of the frequency response of the channel is possible due to the fact that PLC topology is stable, while in wireless channels, change is a continuous. Mobile objects in the wireless channel can cause the frequency response to be continuously changing. Where as in PLC the power line network components and the power line channel itself are fixed, which makes it possible to determine the characteristic of the network. Some devices connected and disconnected causes a slight change from time to time, but this change is not significant compared to the change in wireless channels.

In PLC the primary cause of multipath propagation is reflection of signals due to impedance mismatches. These mismatches happen at branching points and where loads unmatched to the power line channel are connected. The plugging on and off of devices also causes these impedance mismatches. The profile of reflections is dependent on the power line network topology and the loads connected to the network. Despite the fact that the type of cable used is one, multiple reflections happen at branching and termination points [9],[14],[28].

The most widely accepted and used PLC channel model is the Zimmerman and Doster model. A generalized transfer function combining multipath propagation and frequency- and length- depending attenuation is defined as [29]

\[
H(f) = \sum_{i=1}^{N} |g_i(f)|e^{j\phi g_i(f)}e^{-(a_0 + a_1 f^k) d_i}e^{-j2\pi f\tau_i}
\]  

(1.11)

where

- N is number of multipath components,
- \( g_i \) summarizes the reflection and transmission factors along the propagation path. It is in general complex and frequency dependent, because reflection points may exhibit complex and frequency dependent values,
- \( a_0 \) and \( a_1 \) are attenuation parameters,
• $k$ is an exponent of the attenuation factor (typical values are between 0.5 and 1),

• $d_i$ is length of path $i$,

• $\tau_i$ is time delay of path $i$,

• $f$ is frequency of operation,

• $\phi$ is the phase of the complex valued weighting factor $g_i$.

Equation 1.11 shows that the frequency response consists of three parts. The first part is the complex and frequency dependent weighting factor, the second part is the attenuation portion, whose parameters are determined experimentally and the last part is the delay portion of the frequency response.

It was found that it is possible to simplify the weighting factor $g_i$. $g_i$ can be assumed as frequency independent and real-valued in many cases. This was revealed by conducting extended measurements. The delay portion of the function can be written as ratio of $d_i$ and the signal propagation speed in the conductor, $v_p$. Thus, the simplified form of the transfer function of PLC multipath channel is given by [29]

$$H(f) = \sum_{i=1}^{L} g_ie^{-(a_0+a_1f^k)d_i}e^{2\pi f \frac{d_i}{v_p}}$$  \hspace{1cm} (1.12)

1.6 Statement of the problem

As power lines are designed to carry electric power which is of low frequency, high frequency signals experience a harsh channel condition, and as such a high level of channel induced error. IN and frequency selective fading are the primary challenges in PLC. OFDM modulation reduces the effect of both, but it is not sufficient when the IN energy level is beyond a certain threshold. So, the OFDM-PLC system needs to be augmented with other IN mitigation methods.

In this thesis work, two of the three basic mitigation methods, ECCs and nonlinear preprocessing, are adopted. The specific ECC chosen is Turbo code, since it is capable of achieving near Shannon limit performance, with relatively lower levels of complexity. With regard to nonlinear preprocessing techniques, hybrid preprocessors (i.e CHP and AHP which employ both clipping and blanking) are chosen, because they have better error performance than blanking and clipping applied separately. These two methods, Turbo codes and hybrid nonlinear preprocessors
are applied to OFDM-PLC systems to reduce error, which is primarily caused by IN.

1.7 Objective

1.7.1 General objective

The primary target of this thesis work is to apply a dual (i.e Turbo codes with Hybrid nonlinear preprocessors) IN mitigation technique to OFDM-PLC systems and interpret the resulting error performance for different levels of channel impulsiveness.

1.7.2 Specific objectives

The specific objectives of the thesis work include,

- To understand the effect of multipath and IN in PLC.
- To model IN and multipath nature of PLC channels.
- To study the effect of OFDM on multipath and IN in PLC in comparison to single carrier modulation techniques.
- To study different IN mitigation techniques available in literature.
- To evaluate performance of different nonlinear preprocessing techniques.
- To design an OFDM-PLC system with the proposed dual IN mitigation technique applied.
- To simulate the designed system and determine its error performance under various IN conditions and analyze the results obtained.
- To study error performance of the proposed IN mitigation technique considering multipath effect.

1.8 Scope of the Study

Since the powerline channel considered in this thesis work is memoryless, bursty IN occurrence is not considered. In this work, showing the error performance improvement achieved by the combined application of Turbo codes and AHP/CHP
mitigation techniques to OFDM-PLC systems by simulations is the main concern. MCA noise model is used exclusively to model noise in the channel and different levels of channel impulsiveness are considered.

As the primary focus is on IN, not multipath characteristic of the channel, the number of channel paths is kept fixed while the impulsive index and the Gaussian to IN average power ratio are varied, and channel equalization or other multipath mitigation methods are not employed. The error performance gain achieved by the application of the proposed dual mitigation technique in comparison to other methods chosen for comparison is determined.

Binary Phase Shift Keying (BPSK) modulation is used in all simulations. With regard to the Turbo codes applied alongside hybrid nonlinear preprocessors, binary Turbo codes are exclusively considered.

1.9 Significance of the Study

The contribution of this thesis work lies in proposing an untried combination of high performance ECC and hybrid nonlinear preprocessors, which are both of relatively low complexity while achieving significantly reduced BER in communication systems.

The dual mitigation methods proposed in this work can be incorporated in modern OFDM-PLC devices and ensure relatively lower error rate data communication. Even though CHP and AHP methods are relatively more complex compared to blanking and clipping nonlinear preprocessing methods, with the continually improving digital signal processing techniques and processing speed, their better performance can outweigh their complexity and they can applied to achieve better error performance.

The proposed mitigation technique is not limited to PLC systems as it can also be applied to any system which is prone to significant effect of IN. Thus, it can be employed to both wired and wireless communication systems.

To make the ground for justification of final results, separate treatment of Turbo coding and nonlinear preprocessors for different channel conditions using MCA parameters are also made. These results and explanations can give an insight about the performance of Turbo coding and hybrid nonlinear preprocessors in
OFDM-PLC systems, and also strengthen the argument to choose the model over other noise models.

1.10 Methodology

The main target of this thesis work is to achieve an improved error performance by applying Turbo coding and CHP/AHP techniques together to an OFDM-PLC system affected by different levels of IN. Towards the accomplishment of the task the following methods and materials are used.

As shown in Figure 1.4, the first step of the thesis work is literature review. Basically the problem at hand can be assumed to have three main parts: PLC, OFDM and IN mitigation. Materials related to these topics are gathered and studied throughly. The range of materials to be studied includes, but not limited to, the references used in preparing this thesis document. Books, standard journals, research publications and other online materials related to an OFDM-PLC system, IN, multipath effect, Turbo codes, nonlinear preprocessing and other related concepts are carefully studied to build the background knowledge required to successfully accomplish the work.

The next task is modeling of an OFDM-PLC system. The channel basically has two important characteristics of interest: IN and multipath. Both characteristics are modelled by using the mathematical models described by Equation 1.3 and Equation 1.12, respectively. MCA noise model, which is in a form of an infinite sum, but can be approximated by the first few terms of the sum and still be fairly accurate, is used to model IN. MCA model is chosen over the BG noise model, which has been widely adopted in the literature, because MCA model parameters can directly be related to the physical channel as described in [2]. To model the multipath characteristic of the PLC channel, Zimmermann and Dostert multipath model [29] is chosen because it is relatively simple, relatively more accurate and the most widely used model.

As of the nonlinear preprocessing part at the receiver end, hybrid preprocessors require optimizations of the thresholds and scaling parameters (for AHP only), which are performed by using the output SNR of the preprocessed signal as a parameter of optimization. Threshold and scaling factor parameters chosen are the ones which provide the maximum output SNR value. Such optimization tasks
are performed using MATLAB by searching for values that provide maximum output SNR.

The third major step of the thesis work consists of simulations of the proposed
system using MATLAB. Before simulating the OFDM-PLC system, to which the dual IN mitigation technique is applied, the characteristic of the system with different preprocessors applied is simulated for different channel conditions. These plots of BER of different preprocessors will enable us to explain the final results we get when the overall system with proposed dual mitigation technique is simulated. Simulations are also done with the application of Turbo coding alone to have insight on the performance of Turbo coding under varying channel conditions.

BER performances of different cases are simulated with respect to different input signal SNR values. As the primary aim is to reduce the effect of IN on data communication, different IN conditions will be considered by changing the impulsiveness of the channel. To do that, the impulsive index, $A$, and AWGN to IN average power ratio, $\Gamma$, are assigned different values. In addition to IN, the proposed IN mitigation methods are studied taking multipath effect into consideration. But, the number of channel paths is kept fixed and no multipath mitigation method is applied, as the primary focus of the work is combating IN, not multipath.

Based on the simulation results obtained, the relative performance of AHP/CHP plus Turbo code is compared to other Turbo code and preprocessing combinations. The performance gain of the proposed dual mitigation schemes over others will be explained for different levels of channel impulsiveness. Finally, conclusions are made, and recommendations for future work are pointed out.

1.11 Organization of the thesis

The thesis work is organized by five chapters. General overview of PLC, challenges in PLC (i.e IN and multipath), their modeling and commonly applied mitigation techniques are discussed in the first chapter. Review of related and relevant works are presented in chapter two. The proposed system model and description of system components is dealt with in chapter three. Results and discussions are presented in the fourth chapter, and finally, chapter five of the thesis work contains drawn conclusions and recommendations for future work.
Chapter 2

Literature Review

PLC avoids the need to deploy dedicated infrastructure for different communication applications. This appealing property does not come without demerits. As power line channels are primarily made to carry electric power at low frequencies, high frequency signals find them inconvenient. The two primary challenges are IN and frequency selective fading caused by the multipath nature of PLC channels, which can introduce significant error to the transmitted data. To alleviate these problems, several research is being done. As the primary target of this thesis work is IN mitigation, some selected papers that are strongly related to IN mitigation in OFDM-PLC systems are reviewed.

Shongwe, Vinck and Ferriera (2014) [2] reviewed studies focused on IN. Different IN models are briefly described. As described in the paper, there are basically three categories of techniques for combating IN. They are nonlinear preprocessors like clipping and nulling (or blanking), iterative and ECCs, and it is also described that it has become a common practice to implement a combination of two or more IN combating techniques in one system for enhanced error performance. When it comes to ECCs, convolutional coding, Turbo coding and low density parity-check coding or in general codes that are iteratively decoded are identified to be of a primary choice.

In the paper, single and multicarrier modulation techniques in relation to PLC are discussed. It is clearly indicated that multicarrier modulation methods, for which OFDM is a typical representative, are generally better in terms of error performance than single carrier modulation schemes for impulsive channel conditions.
Error performance of OFDM for different channel conditions (different $A$ and $\Gamma$ values) are simulated and well explained.

Different IN models are also described in the paper. The description is developed from the simplest to more complex and recent models. It is described that BG noise model is relatively more widely adopted in the literature, while MCA noise model is described as the one whose parameters are more relateable to physical channel characteristics. The impulsive index, $A$, is well explained in the paper with examples. It is also indicated that, the PDF of MCA model can be truncated to approximate BG noise model. With regard to nonlinear preprocessing, only the two basic nonlinear preprocessors, blanking and clipping, are explained in the paper; adaptive methods (conventional and hybrid) are not covered.

Generally the paper is aimed at a brief review of IN related points, not in detail coverage of them. Different IN mitigation techniques are briefly explained, but, no in depth analysis is available on any one of them. As the paper is kind of a review on IN mitigation in PLC, it was found to be important input towards the accomplishment of the thesis work.

In a paper by Faber, Scholand and Jung (2004) [21] turbo codes are shown to achieve considerable improvements over non-coded systems in IN scenarios as well as over systems deploying a rate $1/2$ convolutional code with constraint length 5 and octal generators $(28)_8$ and $(35)_8$. In the paper, the suitability of Turbo codes to communication environments affected by IN is asserted. The IN affected communication channel considered in the paper is not specifically set to be a PLC channel.

It is a common practice to combine two or more IN mitigation techniques. But, in the paper, Turbo codes are applied solely, unlike the IN mitigation method proposed in this thesis work. No nonlinear preprocessing or iterative technique is used in conjunction with Turbo codes to reduce the effect of IN. IN occurrence probability, $p$, is taken to be 0.5, and simulation results are based on this value of $p$. The performance of Turbo coding for impulsive environments is not discussed under various levels of channel impulsiveness. In the paper, the Signal-to-impulsive noise ratio (SINR) value is not indicated, and without the knowledge of SINR value, it is impossible to tell the level of channel impulsiveness for sure. It appears that the authors considered both impulsive and Gaussian components in the noise term, as such used SNR only. Symmetric alpha stable noise model is used for IN modeling.
Abd-Alaziz, Mei and Johnston (2017) [14] showed enhanced performance of non-binary Turbo codes in combating IN in OFDM-PLC systems compared to binary Turbo codes. In conjunction with the non-binary Turbo code, blanking and clipping preprocessing techniques are employed. Simulation results indicate, blanking preprocessor combined with Turbo coding (both binary and nonbinary) is better than clipping with Turbo coding for the extremely high level of channel impulsiveness considered (i.e $\Gamma = A = 0.01$).

Performance comparisons with no coding and with binary Turbo coding are made. The performance of nonbinary Turbo coding is found to be better than binary Turbo combined with nonlinear pre-processors, but the improved error performance is achieved at the expense of higher complexity caused by the adoption of non-binary Turbo codes.

The BER versus SNR simulations performed for binary and nonbinary Turbo codes combined with blanking and clipping are done for single values of $\Gamma$ and $A$, both equal to 0.01, representing an ‘extremely high’ impulsive channel condition. The BER performance is not studied for other values of $\Gamma$ and ‘$A$’. So their work is limited to highly impulsive channel conditions. Two different number of multipath components, 4 and 15, are considered. The case with 15 paths is found to result higher levels of BER compared to the case with 4 paths only. So the effect of different levels of frequency selective fading is well presented.

A different approach to improve error performance of an OFDM-PLC system is to adopt a better nonlinear preprocessing technique without the need to go from binary to nonbinary Turbo codes. This path is followed in this thesis. Even if, hybrid preprocessors are slightly more complex than other conventional preprocessors, the level of complexity that comes with the mitigation techniques is expected to be lower than that brought by the adoption of nonbinary Turbo coding.

Different nonlinear preprocessing techniques in light of their performance in combatting IN are compared by Rabie and Alsusa (2015) [5]. The authors proposed to enhance the performance of a hybrid nonlinear preprocessing technique by jointly optimizing the threshold and scaling factor. They differentiated between two types of hybrid preprocessing, the first one they named CHP and the second one AHP. CHP has a fixed scaling factor relating clipping and blanking threshold values (1.4, usually), and AHP is a modification of CHP. They hypothesized, if the scaling factor is also taken as a variable and optimized just like threshold values, performance improvement could result. Simulation results confirm the validity of
their hypothesis. AHP and CHP perform better than clipping and blanking with AHP achieving least BER in all channel conditions.

They used BG noise model. No satisfactory explanation why blanking and clipping have different relative performances for different levels of channel impulsiveness is made. This could stem from their choice of BG as the noise model. MCA is better suited towards this goal, as channel impulsiveness is understood in terms of two parameters (Γ and $A$) in MCA whereas there is only one parameter in BG describing channel impulsiveness, alongside SINR. This one parameter is not as convenient as the MCA parameters to rationalize the relative performance of blanking and clipping under different levels of channel impulsiveness. The task accomplished in the paper is the performance comparison of different preprocessors, and no other IN mitigation techniques are used alongside the preprocessors considered.

To the best of our knowledge, Turbo coding was not used in conjunction with AHP and CHP, to mitigate IN in PLC. So, a dual mitigation technique, by pairing Turbo coding with AHP and CHP, is proposed in this thesis work towards the goal of reducing error caused by IN in PLC.
Proposed System Model

IN is a major challenge in PLC. In this thesis, combined application of hybrid nonlinear preprocessing and Turbo coding is proposed to mitigate IN. Figure 3.1 shows the overall system model of an OFDM-PLC system with the proposed dual IN mitigation technique. As can be observed from Figure 3.1, the system is basically an OFDM-PLC system to which nonlinear preprocessing and Turbo coding are added to reduce the error caused by IN and multipath. The basic components of the system; OFDM, nonlinear preprocessing and Turbo coding are discussed in the following sections.
3.1 Orthogonal frequency division multiplexing (OFDM) in PLC

OFDM is found to be effective in combating both IN and frequency selective fading in PLC. Because of this well established fact, OFDM and PLC goes hand in hand, making OFDM-PLC the name of such systems [2],[5],[14]. The primary defense against the two challenges in PLC is thus OFDM. Other IN mitigation techniques mentioned in previous chapters are applied to enhance system performance.

In OFDM, data transmission is executed in parallel using multiple narrowband carriers. Each of these carriers (usually called subcarriers) modulate a portion of transmitted data. The serial data is first converted to a parallel configuration and modulate subcarriers. This results in lower data rate for each of these parallel streams. The available channel bandwidth now consists of several narrowbands. The frequency response of the channel over this small band is almost linear, which means that signal within each of this subcarrier bands experience flat fading, which is relatively easy to combat. Simple frequency domain equalizers which undo the effect of the small band almost linear channel responses can be used to reduce the effect of multipath characteristic of the PLC channel. In addition, cyclic prefixes are also added to further reduce the multipath effect [3, 30–33].

OFDM modulation using IDFT distributes noise energy in frequency domain over all the subcarriers. The noise energy is spread over a wide OFDM bandwidth. This makes the effect of IN being shared by multiple carriers, making the effect on each of them smaller than it would have been if single carrier modulation was used. Therefore, OFDM is an effective way of combating IN. OFDM’s success in reducing both IN and multipath has made it a common choice in PLC technology.

OFDM is an efficient digital modulation technique that employs multiple mutually orthogonal carriers to transmit data [3],[4],[34]. Each carrier consists of a pair of sine and cosine waves. Such pair of carriers are usually called subcarriers. For an OFDM scheme with N subcarriers, the available transmission bandwidth $B$ is divided into $N$ equal bands, each of which is allocated to a subcarrier. So a subcarrier will occupy a bandwidth of $\frac{B}{N}$ Hz, with $B$ Hz being a single sided bandwidth available for transmission. The wide transmission bandwidth being divided into N narrow bands, each of the sub-carriers after modulating their share of data can be considered narrowband. The OFDM signal is the superposition of
these narrowband subcarrier modulated signals. As a result, for large values of N, the composite OFDM signal is wideband.

Figure 3.2 shows a general block diagram of OFDM system. Serial stream of bits is first converted to parallel form to be mapped into a complex signal using QAM or PSK. In the simulation of this thesis BPSK is used. If, for example, QPSK is used, the serial bit stream is converted into parallel form giving two bits at a time to the QPSK modulator. The modulator maps the dibits it received into a single complex signal. The modulation scheme being QPSK, two bits will be mapped to one of the four complex signals of the constellation. Now we have N complex signals corresponding to the number of subcarriers in the system. These complex signals modulate the subcarriers and the sum of the modulated signals is the OFDM time domain signal.

![Block diagram of a communication system employing OFDM](image)

**Figure 3.2:** Block diagram of a communication system employing OFDM [3].

OFDM modulation is commonly performed using Inverse Discrete Fourier Transform (IDFT). To understand how modulation corresponds to IDFT, a conceptual diagram of Figure 3.3 is used. Let us assume information bits are arranged into groups of two bits (dibits). The first bit of the dibit modulates a cosine signal, and the second a sine signal of the same frequency. These two pairs, the sine and cosine signal, makes a single subcarrier. Let the cosine carrier be seen as the In-phase component (I-phase) and the sine as the quadrature-phase component (Q-phase). In the diagram $a_i$s are the the I-phase and $b_i$s are the Q-phase modulating signals. All the $a_i$s and $b_i$s after being modulated by their respective cosine and sine carriers are added to give the OFDM signal.
Chapter 3. Proposed System Model

Figure 3.3: OFDM modulation conceptual diagram [4].

If \( X(k) \) contains \( N \) complex modulating signals, the OFDM signal \( x(t) \) is given as [4]

\[
x(t) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kf_0 t}, \quad 0 \leq t \leq T \quad \text{and} \quad k = 0, 1, \ldots, (N - 1) \tag{3.1}
\]

where

- \( N \) is the number of subcarriers,
- \( f_0 \) is the inter subcarrier spacing (\( \frac{1}{T} \) for example),
- \( X(k) \) and \( x(t) \) are the modulating signal and the OFDM signal respectively.

If \( x(t) \) is uniformly sampled with an interval \( \frac{T}{N} \), the \( n^{th} \) sample is given as

\[
x(n\frac{T}{N}) = x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kf_0 n\frac{T}{N}}, \quad 0 \leq n \leq N - 1 \tag{3.2}
\]

If we take the inter subcarrier spacing to be \( f_0 = \frac{1}{T} \), the exponential term becomes \( 2\pi kn\frac{1}{T} = 2\pi kn/N \) From which we can rewrite Equation 3.2 as

\[
x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}, \quad 0 \leq n \leq N - 1 \tag{3.3}
\]

Equation 3.3 is exactly the same as the IDFT formula given by definition. It is the IDFT of the sequence of complex numbers \( X(k) \). This is an interesting result as it means OFDM baseband modulation is equivalent to the evaluation of the IDFT of \( X(k) \). The significance of this result lies on the availability of a very efficient algorithm used to evaluate IDFT, which is called Inverse Fast Fourier Transform
(IFFT). It can be used to evaluate IDFT hundreds, even thousands, time faster than the formula given by Equation 3.3 [35].

Back to OFDM block diagram in Figure 3.1, after serial to parallel conversion of the bit stream, PSK or QAM modulation is executed. Then OFDM modulation is performed with an N point IFFT. Next, Cyclic prefix is added to reduce ISI. The beginning portion of the OFDM symbol is copied and appended to the end of the same symbol. This will be affected by interference with other symbols. But, it doesn’t contain original information, as it is a copy. So, it will be discarded at the receiver side, while it achieves reduced ISI.

The parallel time domain OFDM symbols are converted into a serial form, after which the digital data is converted to an analog form and transmitted after passing through a transmitter filter. The D/A process involves up conversion of the baseband modulated OFDM signal, to make its characteristic match that of the PLC channel. The transmitted signal passing through the channel, experiences noise and multipath effects. It is received with distortion due to the harsh channel conditions.

The first component of the OFDM receiver subsystem is a receiver filter. After filtering, the analog signal is converted to a digital form using an A/D converter. The cyclic prefix added at the transmitter side is removed and the signal gets converted back to a parallel form for processing. As OFDM modulation was done with an N-IFFT, the inverse process of OFDM demodulation is done with an N point Fast Fourier Transform (N-FFT). When the channel is divided into smaller band of N subcarriers, the process has simplified the process of equalization by making the subdivided frequency ranges to have an almost linear frequency response, which corresponds to flat fading. So it is the frequency domain equalizer which comes after the demodulation process. In addition to the subdivision of the channel, the channel characteristic is easier to determine than other dynamic channels like wireless channel.

OFDM-demodulated signal is then QAM or PSK demodulated to convert the complex signal into their corresponding bits, which are then converted into a serial form and delivered to the user.

Equation 3.3 indicates a time domain form of a baseband OFDM signal. OFDM modulation and demodulation using IFFT and FFT respectively is performed with baseband signals. But, the OFDM signal transmitted over the PLC channel is a
passband signal given by the following formula [4],

\[
x(t) = \text{Re} \left[ \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi(f_c+kf_0)t} \right], \quad 0 \leq t \leq T \quad \text{and} \quad k = 0, 1, \ldots, (N-1)
\]

(3.4)

Where \( \text{Re}[.\] denotes an operation which gives real part of the expression in bracket and \( f_c \) is a carrier frequency.

Subcarriers in OFDM are orthogonal to each other. In frequency domain they do overlap, but crosstalk is avoided due to their orthogonality nature. This is the reason behind OFDM’s higher spectral efficiency than conventional frequency division multiplexing. Figure 3.4 shows three subcarriers orthogonal to each other over one symbol duration. As can be noted, their frequencies are 1, 2 and 3 times the inverse of the symbol duration, or their periods are the symbol period divided by 1, 2 and 3 for the green, red and blue waveforms, respectively. In this typical case, subcarrier frequencies are \( 1/T, 2/T \) and \( 3/T \), where \( T \) is the symbol duration in second. These three frequencies are not the only combinations that can be assigned to the subcarriers to ensure their orthogonality. Other possibilities are \( 1/T, 2/T, 4/T \) or \( 1/T, 5/T, 9/T \).

Several options of frequency choices are available which ensures orthogonality. But, all combinations are not equally bandwidth efficient. As bandwidth is a scarce resource in communication in general, a combination which will take the least overall bandwidth should be chosen as much as possible. Figure 3.5 shows
the bandwidth saving achieved by OFDM over conventional frequency division multiplexing (FDM).

Figure 3.5: Bandwidth saving achieved by OFDM modulation: (a) Conventional FDM and (b) OFDM. [4]

Figure 3.6 shows the frequency domain plot of a single and six subcarriers [4]. It is observed that where one subcarrier achieves its maximum value, the others have nulls, assuming perfect synchronization. So under perfect synchronization, the receiver’s task is it to evaluate spectral values at discrete frequency values corresponding to maxima points of individual subcarriers. At points where one is maximum, the others are zero, so that each subcarrier can be demodulated independent of others.

Another important concept in OFDM is cyclic prefix. Cyclic prefix extension refers to adding a trailing portion of an OFDM symbol to the head of itself. This added component doesn’t contain a new information, as it is a copy. It contains a redundant information, but it serves an important purpose by helping to reduce intersymbol interference (ISI) [3].

Due to the multipath characteristic of a PLC channel, several copies of a signal arrive at the receiver at different times. Some of the copies arrive at a time when another symbol is being received, which causes ISI. Without cyclic prefix, the
head of a symbol arriving now will overlap with a tail of a symbol transmitted earlier. In the presence of cyclic prefix, the head of a symbol is a copy of its own trailing portion. It contains information that it has at the end of itself. So even if information contained in its head is lost due to interference with a previously transmitted symbol, it contains the same information at its own trailing portion. The interference affected prefix can be removed without any lose and the symbols can be recovered without the damaging effect of ISI. Figure 3.7 shows the addition of cyclic prefix and how it acts as a guard band.

Looking at it in the frequency domain, adding a cyclic prefix makes a signal appear periodic to the channel. In such cases, circular convolution is equivalent to linear convolution of the signal with the channel, which is the output of linear systems.
in general. The DFT of circular convolution of a signal with a channel impulse response is the product of the DFT of the impulse response, which is the frequency response of the channel, and the DFT of the signal. These two frequency domain signals can efficiently be evaluated using FFT [36]. The product of the two is equivalent to the DFT of the circular convolution, which is the same as linear convolution due to the cyclic prefix addition. The convolution is then obtained by IFFT of the product. So, adding cyclic prefix has double advantage: eliminating ISI and making computation of output signal faster by using the FFT algorithm.

OFDM distributes the IN energy among the subcarriers, which results in reduced error rate, until a certain level of IN energy in the system. If the IN energy increases beyond a certain threshold, all the subcarriers will get affected significantly. So the success of OFDM is up to a certain IN energy. Beyond that, other IN mitigation techniques need to be applied along side OFDM to achieve acceptable error performance. The two common methods applied with OFDM are nonlinear preprocessing and ECCs. Hybrid methods from nonlinear preprocessing and Turbo coding from ECCs are chosen to make the proposed dual IN mitigation schemes. The next two sections discuss these two components of the proposed IN mitigation techniques.

3.2 Nonlinear preprocessing

Nonlinear preprocessors are commonly applied alongside OFDM to reduce BER in PLC [15],[37–39]. Preceding an OFDM receiver with a nonlinear preprocessor is a simple IN mitigation technique. Its simplicity and ease of implementation has resulted in wide application of such preprocessors to reduce IN effect in PLC. The term preprocessor indicates the fact that the nonlinear device is placed before the OFDM receiver. It preprocesses the received samples before the main processing by the main receiver.

The following nonlinear preprocessors are commonly applied in PLC [5],[15],[39].

- **Blanking**: refers to the task of removing or nulling samples whose amplitudes are greater than a certain threshold, while samples with smaller
Chapter 3. Proposed System Model

Amplitudes are not changed. In equation form it is given as,

\[ y_k = \begin{cases} 
  r_k, & |r_k| \leq T \\
  0, & |r_k| > T 
\end{cases} \]  

(3.5)

Where \( T \) is blanking threshold and \( k = 0, 1, ..., N - 1 \). \( r_k \) and \( y_k \) are input and output of the blanking nonlinear device, respectively.

- **Clipping**: involves replacing samples whose amplitudes are greater than a certain threshold with a sample of fixed amplitude. In other words samples greater than threshold value are clipped and those with smaller amplitude values are left untouched. In equation form it is given as,

\[ y_k = \begin{cases} 
  r_k, & |r_k| \leq T \\
  T e^{j \text{arg}(r_k)}, & |r_k| > T 
\end{cases} \]  

(3.6)

Where \( T \) is the clipping threshold and \( k = 0, 1, ..., N - 1 \), and \( \text{arg}(r_k) \) is the angle of the complex input sample to the clipping preprocessor.

- **Clipping/blanking (conventional hybrid)**: is a hybrid preprocessing scheme in which samples with amplitudes that lie between clipping and blanking thresholds are clipped, and those with sample amplitudes greater than blanking threshold are nulled. Similar to the blanking and clipping cases, samples with amplitudes less than the clipping threshold are passed unaffected.

Blanking threshold value is a constant times clipping threshold, and 1.4 is a commonly used constant [5]. In equation form it is given as,

\[ y_k = \begin{cases} 
  r_k, & |r_k| \leq T_1 \\
  T_1 e^{j \text{arg}(r_k)}, & T_1 < |r_k| \leq T_2 \\
  0, & |r_k| > T_2 
\end{cases} \]  

(3.7)

Where \( T_1 \) and \( T_2 \) are the clipping and blanking thresholds respectively, and usually \( T_2 = 1.4 T_1 \).

- **Optimized Clipping/blanking (Adaptive hybrid)**: is similar to conventional hybrid in that it employs both clipping and threshold. The only difference is, in adaptive hybrid case, clipping and blanking thresholds are...
Chapter 3. Proposed System Model

not related by a fixed value. The constant of proportionality, also called scaling factor, between them is treated as a variable, and it needs to be optimized alongside an a threshold. Equation 3.8 expresses AHP in equation form.

\[
y_k = \begin{cases} 
    r_k, & |r_k| \leq T \\
    T e^{j \arg(r_k)}, & T < |r_k| \leq \alpha T \\
    0, & |r_k| > \alpha T 
\end{cases}
\]  

(3.8)

Where \( \alpha \geq 1 \) is the scaling factor.

Figure 3.8 shows nonlinearity applied before an OFDM demodulator [5]. It can be observed from the figure that all preprocessors act only on the amplitude of samples, which is why the block representing preprocessors comes after the \(|.|\) block which denotes determination of amplitudes of samples. The angle of samples, which is determined by the block represented by \( \text{ang}(.) \) in the figure, is left unchanged. Comparing the amplitude with a certain preset threshold, any of the above mentioned preprocessors act. The phases of the samples remain unchanged. When the sample amplitude is too large, all preprocessors replace it with zero, due to the fact that, too large amplitudes are more likely to carry more IN energy, so that to reduce IN effect, such samples are nulled. For smaller samples, all preprocessors perform differently according to their definitions.

3.2.1 Hybrid nonlinear preprocessing

AHP and CHP perform both clipping and blanking using two thresholds (i.e blanking and clipping thresholds). They make use of the strengths of both blanking and clipping. Comparing the performance of clipping and blanking, one is better than
the other depending on the channel impulsiveness and input information bearing signal condition (SNR). For highly impulsive channels, blanking preprocessors perform better than clipping. Where as, in better channel conditions and higher SNR values, clipping is better.

In [15], it is indicated that clipping has a slightly better performance when the signal-to-impulsive noise energy (SINR) is close to 0dB. But as SINR decreases, typically below -6 dB, blanking rapidly overtakes clipping in terms of performance and asymptotically, as SNR approaches $-\infty$, blanking performance approaches ideal blanking scheme (i.e blanking with ideal impulse detection). The reason is, in highly impulsive channels large samples contain a significant amount of noise energy. Those samples are predominantly made of noise energy, and the information bearing signal in them is small. In such cases, the overall IN energy in the system can significantly be reducing by nulling those large amplitude samples, which causes minimal information loss. So in such cases, blanking generally performs better.

On the other hand, when the impulsiveness is not that strong and the signal energy is higher, most received samples contain predominantly the information bearing signal. The noise component in them is smaller than the useful signal energy. If such samples are blanked, significant information loss occurs. So, the more convenient preprocessing in such cases is clipping. By clipping large samples, the noise energy can significantly be reduced while maintaining the information bearing signal affected only slightly.

The difference between CHP and AHP is, in CHP clipping and blanking thresholds are related by a fixed constant. Only one variable can be varied to optimize preprocessor performance. A commonly used constant of proportionality between the two thresholds is 1.4 [5], in which blanking threshold is 1.4 times clipping threshold. Where as, in AHP, the relating factor (called scaling factor, $\alpha$) is not fixed. It is considered as a variable. In [5], it is asserted that performance of a hybrid preprocessor doesn’t only depend on the threshold, but on the scaling factor too. In the same paper it is shown that AHP provides the smallest BER comparing to other preprocessors for the same SNR value and channel conditions.

AHP’s better performance doesn’t come without a cost. Because two parameters need to be optimized, optimization is relatively more complex and takes longer time. For the other preprocessors, only one parameter needs to be optimized. For different SNR values, different threshold value achieve maximum output SNR
which corresponds to the least BER. Choosing a threshold that provides maximum output SNR is the task for clipping, blanking and CHP. For AHP, both threshold value, $T$, and scaling factor, $\alpha$, determine the output SNR value of preprocessed signal. In other words two threshold values, clipping and blanking thresholds need to be optimized. But, since processing speed is rising fast and optimization need to be done once for fixed channel conditions and transmitted signal SNR, the fact it requires more complex and longer time taking optimization can be outweighed by the reduced BER it achieves.

### 3.2.2 Signal-to-Noise ratio at the output of nonlinear preprocessor

The output of a nonlinear device can be expressed as [15]

$$ y_k = K_0 s_k + d_k, \quad k = 0, 1, ..., N - 1, $$

(3.9)

where

- $y_k$ is output of preprocessor,
- $K_0$ is an appropriately chosen scaling factor.
- $K_0 s_k$ represents a scaled replica of the information-bearing signal,
- $d_k$ represents the cumulative noise and distortion term,

Usually the noise process and the useful signal are required to be uncorrelated. If we set this as requirement, then $E[d_k s_k^*] = 0$. Multiplying both sides of Equation 3.9 by $s_k^*$ and taking the expectation of both sides and rearranging terms gives an expression for $K_o$ as [15],

$$ K_o = \frac{E[y_k s_k^*]}{E[|s_k|^2]} $$

(3.10)

and the SNR at the output of a nonlinear preprocessor can be expressed as [15],

$$ SNR_{out} = \frac{E[|K_0 s_k|^2]}{E[|y_k - K_0 s_k|^2]} = \frac{1}{\frac{E_{out}}{2K_o^2} - 1} $$

(3.11)

where $E_{out} = E[|y_k|^2]$ is the total signal power at the output of a nonlinear preprocessor.
The DFT module used to demodulate the OFDM signal, doesn’t alter the SNR value that it receives. So the SNR value at the output of the preprocessor is the value that the demodulator gets. Maximizing this SNR value ensures improvement of BER.

In this thesis, the maximum SNR value that can be achieved with different threshold values is evaluated using MATLAB. As this value depends on $K_o$ and $y_k$, these two values are evaluated by varying the blanking or clipping threshold values according to the type of preprocessing used, at a fixed input SNR value, while the energy of the useful signal, $s_k$, is normalized to one. So, for a fixed input signal SNR value, different values of $K_o$ and $y_k$, as a result $E_{out}$, are obtained for different threshold values. Output SNR is then evaluated using Equation 3.11. The threshold that gives maximum $SNR_{out}$ value is then used. Similarly, all optimum threshold values are evaluated for a range of input signal SNR values. Maximizing the SNR will result in minimum BER value in general. So optimizing threshold values and performing preprocessing using those values will give us a minimum error rate for a given preprocessing method.

For blanking, clipping and conventional hybrid, only one threshold value needs to be optimized. For CHP, there are actually two thresholds: blanking and clipping, but, as they are related by a constant, in effect, there is only one threshold which needs optimization. The case is different for AHP as the two thresholds are not related by a fixed value. Changing the relating constant is found to have a significant effect on $SNR_{out}$ of the preprocessor. That is why it needs to be considered as a variable just like the threshold value. So, in AHP, two parameters need optimization; threshold, $T$ and scaling factor, $\alpha$. The pair that provides maximum $SNR_{out}$, will be used for that specific channel condition and specified input signal SNR value.

Turbo codes which are commonly used iteratively decoded IN mitigation ECCs are discussed in the next section. They are among the ECCs standardized to be used in PLC systems [2],[14] [40–43].

### 3.3 Turbo codes

Turbo codes are powerful codes discovered in 1993 by a group of researchers in France. Parallel concatenation of codes is exploited by Turbo codes, in which two
component codes are connected in parallel. An interleaver is used to interleave the input to one of the two constituent encoders. The constituent encoders are systematic. But, the information bits from the systematic line of the second encoder are not transmitted, which increases the coding rate of the system [7].

To decode Turbo encoded data, two decoders exchanging information are used. Generally the decoders are based on a maximum a posteriori (MAP) probability algorithm or a soft output Viterbi algorithm (SOVA). Soft estimate of the input information is generated by the component decoders. The decoding process is iterative; it will be performed for a number of iterations while the component decoders exchange information [7],[44].

### 3.3.1 Turbo Encoding

A Turbo encoder is made of two parallel concatenated recursive systematic convolutional (RSC) encoders and an interleaver separating them. Figure 3.9 shows a generic block diagram of a Turbo encoder. The constituent encoders may be any type of Forward Error Correcting codes, but it is a common practice to use two identical RSC codes. As can be noted from the figure, the systematic and parity outputs from the constituent encoders are converted into a serial form to make a single code word.

Alongside the constituent encoders, the interleaver is an essential component of the Turbo code. It rearranges the order of bits of the input data in a prescribed, but irregular way. The primary role of the interleaver is to generate a long block code from small memory convolutional codes. The secondary purpose is to decorrelate inputs to the constituent encoders. This decorrelation increases the probability of correcting remaining errors by the second decoder after error correction by the first decoder. If the inputs to the two encoders were identical, such advantage would not exist [7].

There are different types of interleavers, of which two common interleaver types are discussed below.

- **Block interleavers**: Types of interleavers which arrange the input sequence into a matrix of \(M\) rows and \(N\) columns, as a result of which they require memory size of \(M \times N\). Inputs are written row-wise and read out column-wise. Figure 3.10 shows a block interleaver [7].
Block interleaving and deinterleaving are relatively easy to implement, their weakness being their poor performance with low weight input sequences. When certain low weight input patterns are written row-wise and read out column-wise, they would give similarly low-weight sequences. This would deteriorate the performance of Turbo codes. The solution to this problem is to read bits in a random fashion, which is how pseudo-random interleavers operate.

- **Random interleavers**: A block of N input bits is read into the interleaver and read out pseudo-randomly. The interleaving pattern should be known by the decoders. Reading of bits pseudo-randomly reduces the chance of low weight input sequences resulting in the same or different low weight patterns, compared to block interleavers. Due to this reason, random interleavers are standard interleavers in Turbo coding [6], as a result, they are used in this work.
In Figure 3.9, the input data is a bit sequence denoted by $X_i$, which is directly passed to the output making the systematic output component. The upper encoder encodes the input data $X_i$ as it is and produces sequence $Z_i$. The second encoder, which is usually identical to the first one, acts on interleaved data $X'_i$ resulting in sequence $Z'_i$. $X'_i$ contains the same information as the original input, but the order of bits is rearranged in a pseudo-random fashion. Because the sequences to the two encoders are different, their outputs are also different. The three parallel inputs $X_i, Z_i$ and $Z'_i$ are serialized to make the output of the Turbo encoder.

Constituent encoders in Turbo code are commonly RSC encoders. RSC codes are made from conventional convolutional codes in such a way that one of the outputs is fed back to the input. This feedback makes the encoder recursive. Figure 3.11 shows a typical RSC encoder used in UMTS Turbo code. One of the two parity lines being folded back and connected to the input, a systematic output can be taken while maintaining the same 1/2 code rate. This encoder’s code words contain bit sequences that can be separated into systematic and parity portions, which is required by Turbo decoders.

\[ G(D) = \begin{bmatrix} 1 & \frac{g_1(D)}{g_0(D)} \end{bmatrix} \]  \hspace{1cm} (3.12)

where $g_0(D)$ and $g_1(D)$ are feedback and feedforward polynomials, respectively. Thus, for the RSC encoder shown in Figure 3.11, the generator matrix $G(D)$ can be written as,

\[ G(D) = \begin{bmatrix} 1, 1 + D + D^3 \end{bmatrix} \frac{1}{1 + D^2 + D^3} \]  \hspace{1cm} (3.13)
3.3.2 Trellis Termination

Trellis termination means driving an encoder to all-zero state [7]. A nonsystematic convolutional encoder can be returned back to the all zero state by transmitting sequence of zeros as many as the memory order. This method of trellis termination does not work for RSC encoders.

Since the component encoders of a Turbo encoder are recursive, it is impossible to perform trellis termination by transmitting zeros with number equal to the memory order of the RSC encoders. Figure 3.12 [7] shows an example of trellis termination for an RSC encoder. From the figure, it can be noted that making sure \( a_t \) is zero for four (memory order of the encoder) clock cycles will ensure that all stages of the shift register have zero state. This is performed by shifting the switch position to "B", when trellis termination is required. This switching will be performed after a block of input data is encoded with the switch position at "A". This method will terminate the trellis of one of the encoders, but it is unlikely that it will do so for both encoders simultaneously. So after trellis termination, usually, the upper encoder will have all-zero state, while the lower encoder will have an unknown final state. This unknown final state will cause performance degradation. But, this degradation is negligible for large interleaver size \( N \). So a 1/3 rate Turbo encoder is equivalent to a linear systematic block code of length \((3(N + v), N)\) after termination, where \( N \) and \( v \) are interleaver size and memory order respectively [7],[45].

![Trellis termination of RSC encoder](image)

**Figure 3.12:** Trellis termination of RSC encoder [6].

3.3.3 UMTS Turbo code

UMTS which stands for Universal Mobile Telecommunications System, is a widely accepted 3G cellular standard, whose standards are prepared and published by the
Third Generation Partnership Project (3GPP) [44],[46],[47]. Turbo codes and convolutional codes are the alternatives to be used with UMTS. The choice depends on the application and technology availability. Figure 3.13 shows the Turbo encoder of UMTS. The constituent encoders are identical RSC encoders and each having a constraint length of four. As it is shown in the figure, the systematic output of the first encoder $X_i$, the parity output of the first and the second encoder, $Z_i$ and $Z'_i$ respectively, are serialized to make the final UMTS Turbo code output. The systematic output of the second encoder is suppressed. The coding rate is approximately equal to $1/3$; to be exact coding rate $r = k/(3k + 12)$. The reason for the coding rate $r$ to be less than $1/3$ is, along the parity bits, trellis termination bits are also transmitted.

Trellis termination for UMTS Turbo encoder is done using a similar technique discussed in Section 3.3.2, by switching to position 'B', after transmitting $k$-bits. The feedback line connected by an XOR gate with itself will make the input to the shift register zero. Continuing the same process, the states of component encoders are brought back to the initial all-zero state after three clock cycles. This termination bits are also transmitted with the systematic and parity bits of the two encoders to result in a coding rate slightly less than $1/3$.

UMTS Turbo code is used in [21] to show performance of Turbo code in IN affected communication systems. It is also the default code in MATLAB Turbo encoder function and encoder block. Being typical and popular Turbo code, it is used in
this work alongside hybrid nonlinear preprocessors, to mitigate IN in OFDM-PLC systems.

### 3.3.4 Turbo Decoding

Demodulators in communication receivers can produce soft or hard decisions. The decisions are transferred to decoders. Soft decision systems result in a better performance than their hard decision counterparts. If the final output of the decoder is bits, then such a decoder is referred as soft input/hard output decoder [46], [48], [49].

If a decoder outputs a soft decision and passed to another decoder, we have soft input/soft output decoder, which has a better performance than soft input/hard output decoders. Turbo decoders make use of such kind of decoders.

The commonly used decoding algorithm in Turbo codes is called BCJR, which was discovered by Bahl, Cocke, Jelinek and Raviv in 1974. The BCJR algorithm, which is also called Maximum a Posteriori (MAP) or forward-backward algorithm, is a general decoding algorithm that can be applied to not only Turbo codes, but to block and convolutional codes also. Because it is more complex than Viterbi algorithm, it was not extensively used, until the discovery of Turbo codes in 1993 [8, 45, 48–50].

To describe the BCJR algorithm, let us assume a convolutional encoder described by a trellis, with output codeword \( x = x_1 x_2 x_3 \ldots x_N \). Each of the elements of the sequence \( x \) are \( n \) bit symbols, so the codeword consists of \( N \times n \) bits, where \( x_k \) is the output symbol generated at time \( k \). In binary codes, input message bit \( u_k \) can be either 0 or 1, which can be represented by -1 and +1, respectively, for convenience, with a priori probability of \( P(u_k) \). From a priori probability of input bits, log-likelihood ratio (LLR) is defined as [8],

\[
L(u_k) = \ln \frac{P(u_k = +1)}{P(u_k = -1)} \quad (3.14)
\]

If the codeword is transmitted over a memoryless AWGN channel and received by a decoder as \( y = y_1 y_2 y_3 \ldots y_N \), which is a sequence of \( n \times N \) real numbers, it will be used by the BCJR algorithm to estimate the transmitted input message bit \( u_k \). The algorithm evaluates a posteriori probability (APP), \( P(u_k|y) \), to estimate the
input message sequence. The LLR is defined from the APP as follows,

\[
L(u_k|y) = \ln \frac{P(u_k = +1|y)}{P(u_k = -1|y)}
\]  

(3.15)

The LLR given by Equation 3.15 is a real number, based on which the transmitted bit at time \( k \), \( u_k \) is estimated. The sign of the the LLR decides the estimation. If it is positive \( u_k = 1 \) is decided and if it a negative number \( u_k = 0 \) is decided as the bit transmitted. The magnitude tells the level of confidence about the decision. The further it is from zero, the more confident the decision is.

The convolutional encoder considered is assumed to have \( M \) states and coding rate of \( \frac{1}{n} \). To estimate \( u_k \), the current state of the encoder is \( S_k = s \) and the previous state is \( S_{k-1} = s' \), and the received sequence \( y \) is divided into three parts, representing the past, current and future subsequences, as follows

\[
y = y_1y_2...y_{k-1}y_ky_{k+1}y_{k+2}...y_N
\]  

(3.16)

Using Equation 3.16, it is shown in that the LLR in Equation 3.15 can be written as \([8],[50]\),

\[
L(u_k|y) = \ln \frac{\sum_{R_1} P(s, s', y)}{\sum_{R_0} P(s, s', y)} = \ln \frac{\sum_{R_1} \alpha_k(s') \gamma_k(s, s') \beta_k(s)}{\sum_{R_0} \alpha_k(s') \gamma_k(s, s') \beta_k(s)}
\]  

(3.17)

where,

- \( P(s, s', y) \) is the joint probability of receiving sequence \( y \) and being in state \( s \) and \( s' \) at times \( k \) and \( k - 1 \), respectively,
- \( R_1 \) and \( R_0 \) indicate the summation is done over all state transitions brought by \( u_k = +1 \) and \( u_k = -1 \), respectively,
- \( \alpha_{k-1}(s') = P(s', y_1, y_2...y_{k-1}) \)
- \( \gamma_k(s, s') = P(y_k, s|s') \)
- \( \beta_k(s) = P(y_{k+1}y_{k+2}...y_N|s) \)

Probabilities \( \alpha \) and \( \beta \) are associated with the past and future of sequence \( y \), and they are calculated recursively as follows,

\[
\alpha_k(s) = \sum_{s'} \alpha_{k-1}(s') \gamma_k(s, s') \quad \text{Initial conditions : } \alpha_0(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0 \end{cases}
\]  

(3.18)
\[
\beta_{k-1}(s') = \sum_s \beta_k(s) \gamma_k(s, s') \quad \text{Initial conditions: } \beta_N(s) = \begin{cases} 
1 & s = 0 \\
0 & s \neq 0 
\end{cases} \tag{3.19}
\]

To find \(\alpha_k(s)\), the summation is done over all previous states \(S_{k-1} = s'\) that are linked to the current state \(S_k = s\) on the trellis. Since current value of \(\alpha\) is determined from previous values of \(\alpha\), it can be evaluated as the sequence \(y\) is being received, going forward from beginning to end of trellis. So forward recursion is used to calculate the values of \(\alpha\) for all state nodes. On the other hand, to evaluate \(\beta_{k-1}(s')\), the summation is done over all next states \(S_k = s\), which correspond to edges on the trellis which start from \(s'\). So, the recursion to determine \(\beta\) is going backwards, from end to beginning of trellis, which necessitates the calculation of \(\beta\) to be done after the whole sequence \(y\) is received. These two directions of recursion (i.e forward for \(\alpha\) and backward for \(\beta\)), have given the BCJR algorithm another name; forward-backward algorithm.

\(\gamma\) values, which are required in the recursive calculations of both \(\alpha\) and \(\beta\) values, are determined as follows for a memoryless AWGN channel [8],

\[
\gamma_k(s, s') = C_k e^{u_k L(u_k)/2} \exp \left[ \frac{L_c}{2} \sum_{l=1}^{n} x_{k_l} y_{k_l} \right] \tag{3.20}
\]

Where \(C_k\) is a quantity which cancels out when \(L(u_k|y_k)\) is evaluated. \(x_{k_l}\) and \(y_{k_l}\) are the \(l^{th}\) bits of the \(k^{th}\) symbol of the encoder output and decoder input, respectively, and \(L_c\), which is called channel reliability value, is given as,

\[
L_c = 4aR_cE_b/N_0 \tag{3.21}
\]

where \(a\) is the fading amplitude of the channel, \(R_c\) is the coding rate, \(E_b\) is transmitted energy per message bit and \(N_0/2\) is the bilateral noise spectral density of the channel.

Using an expanded trellis of the encoder, the estimates of the transmitted bit \(u_k\) are determined with the following steps,

1. Compute \(\gamma(s, s')\) using Equation 3.20 for each trellis branch.

2. Calculate \(\alpha_k(s)\) values for all state nodes using the forward recursion formula given in Equation 3.18 using initial condition \(\alpha_0(s)\), going from beginning to end of trellis.
Chapter 3. Proposed System Model

3. Similarly calculate $\beta_k(s)$ values with the backward recursion formula given in Equation 3.18 from initial condition $\beta_N(s)$, going from end to beginning of trellis.

4. Evaluate the conditional LLR, $L(u_k|y)$ using Equation 3.17 by using the probabilities calculated above.

5. Compare $L(u_k|y)$ with zero, and decide received bit as 1 if it is positive and 0 if it is negative.

In practical applications of BCJR algorithm, all the probabilities described above, need to be normalized to achieve numerical stability.

To see the application of BCJR algorithm for iterative decoding of Turbo codes, let us consider a $1/n$ coding rate systematic RSC encoder, for which the first bit of the encoder symbol at time $k$, $x_{k_1}$ is equal to the transmitted bit $u_k$. In [8], it is shown that, the LLR, $L(u_k|y)$, can be expressed as a sum of three terms as,

$$L(u_k|y) = L(u_k) + L_c y_{k_1} + L_e(u_k) \quad (3.22)$$

The third term depends on the parity bits, while the first two are related to the input information bit at time $k$, $u_k$. As a result $L_e(u_k)$ is called extrinsic information, and it gives a more accurate estimate of $L(u_k)$.

In Turbo decoding, once the a priori probability $L(u_k)$ is used to find $L_e(u_k)$, it will be dropped and the more correct estimate of a priori probability, which is $L_e(u_k)$, is passed to the next decoder, and used with $L_c y_{k_1}$ to get a more accurate $L(u_k|y)$, as a result a more accurate estimate of $u_k$. Figure 3.14 shows a simplified block diagram of a Turbo decoder employing BCJR decoding algorithm, in which $y_{k_p}^{(1)}$ and $y_{k_p}^{(2)}$ are the first and second parity bits of the symbol received at time $k$, and blocks labeled $P$ and $P^{-1}$ perform interleaving and de-interleaving operations [8].

The following decoding steps are used to estimate transmitted bits by the Turbo decoder using BCJR algorithm [8],[50],

- Initially, $L(u_k)$ is set to zero, assuming equally likely 0s and 1s encoded. Using $L_c y_{k_1}$, which depends on the systematic bit, and $L_c y_{k_p}^{(1)}$, which depends
on the first parity bit of the symbol, extrinsic information $L_{e1}(u_k|y)$ is determined and passed to the next decoder, after appropriate interleaving, to be used as a priori probability, while $L_1(u_k|y)$ is dropped.

- Decoder 2 outputs its own extrinsic information, $L_{e2}(u_k|y)$, using properly interleaved $L_{e1}(u_k|y)$ as an educated guess of $L(u_k)$. Alongside $L_{e1}(u_k|y)$, the two parity bits $y_{kp}^{(1)}$, after interleaving, and $y_{kp}^{(2)}$, both multiplied by $L_c$ are also inputs to Decoder 2.

- Next iteration begins with de-interleaved $L_{e2}(u_k|y)$ as an a priori probability for Decoder 1. In such a way more accurate estimates of $L(u_k)$ are obtained, until a stop criteria is fulfilled, at which time Decoder 2 outputs $L_2(u_k|y)$ de-interleaved to a decision device that estimates message bit $u_k$ based on the sign of the LLR. Based on de-interleaved $L_2(u_k|y)$, information bit is estimated as 1 or 0, based on whether it is positive or negative, respectively.
Results and Discussion

In this chapter, simulation results of the proposed dual mitigation techniques and discussions are presented. Turbo coding plus AHP/CHP are the IN mitigation techniques proposed in this thesis. MATLAB simulation results are used to show performance improvements brought by the proposed mitigation techniques compared to other commonly applied dual IN mitigation schemes formed by pairing Turbo coding with blanking and clipping separately.

Before presenting final results, results which are related to the work and which pave the way to justify final results are presented. After common simulation parameters are presented, the first results presented are random IN samples generated using MATLAB for different levels of channel impulsiveness followed by the plot of PDFs for different levels of channel impulsiveness.

Next, error performance of OFDM-PLC systems for different channel conditions are presented. These results show the dependence of BER performance in OFDM-PLC on channel parameters without the application of IN mitigation schemes, after which, optimization of nonlinear preprocessing threshold values is discussed. Results are presented accompanied by discussions to show how optimization of parameters is performed.

Before applying the proposed IN mitigation schemes to OFDM-PLC systems, performance improvement brought by hybrid preprocessors compared to clipping and blanking is presented for different channel conditions. MCA model, the noise model used exclusively in all the simulations in this thesis, is convenient to do that. The values of $A$ and $\Gamma$ are varied to simulate different levels of PLC channel impulsiveness.
Finally, Turbo coding error performance results are presented and discussed. The Turbo code chosen in this thesis is the one applied in UMTS technology, which was explained in section 3.3.3. Similar to the case with preprocessors, performance of Turbo coding is also examined under different levels of channel impulsiveness.

Once the characteristic and performance of the two component IN mitigation methods is discussed separately, performance improvements achieved by the combined application of Turbo coding and AHP/CHP compared to Turbo coding plus clipping and Turbo coding plus blanking preprocessor combinations are presented. Even if, the primary target of the thesis is to study IN mitigation, error performance of one of the proposed schemes (AHP plus Turbo coding) considering multipath effect is also presented at the end of the chapter.

4.1 Simulation parameters

Different parameters are used in undertaking plots and simulations. The values of Γ and A are chosen depending on the channel condition needed to be simulated. Generally typical values of Γ and A are in the ranges [0.0001,0.1] and [0.001,0.5] respectively [38]. Values of Γ and A both equal to 0.01 are also used in [14] to simulate extremely impulsive channel conditions. Based on these two observations about the two parameters, different Γ and A values are used to simulate different levels of channel impulsiveness.

Number of subcarriers is set to 3072 according to IEEE 1901 standards. Not all of the subcarriers are used in practical applications. But, in this thesis, we assumed all subcarriers are used to carry data and refrain from dealing with practical issues of masking some of the subcarriers to avoid interference with other nearby radio frequency transmissions.

The first 10 terms of the infinite series of the PDF of a noise sample in MCA are used in all simulations. It is shown in [2] that taking the first few terms ensures fairly accurate results. But, to be achieve high level of accuracy 10 terms are used in this thesis. All the other parameters are also set with some logic behind the selection of the specific values. Simulation parameters are summarized in Table 4.1.

SNR is the ratio of signal power to Gaussian noise power and SINR is the ratio of signal to IN average powers. Different SNR values are used based on their
### Table 4.1: Simulation parameters

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of subcarriers</td>
<td>3072</td>
</tr>
<tr>
<td>2</td>
<td>PSK Modulation order</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Turbo block size</td>
<td>1020</td>
</tr>
<tr>
<td>4</td>
<td>Number of multipath components</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>BER value used for comparison</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>6</td>
<td>Turbo decoding algorithm</td>
<td>APP</td>
</tr>
<tr>
<td>7</td>
<td>Number of Turbo decoding iterations</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>Noise Model</td>
<td>MCA</td>
</tr>
<tr>
<td>9</td>
<td>Number of terms of MCA PDF</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>Range of $A$ values</td>
<td>[0.001,0.5]</td>
</tr>
<tr>
<td>11</td>
<td>Range of $\Gamma$ values</td>
<td>[0.0001,0.1]</td>
</tr>
</tbody>
</table>

ability to clearly convey representative characteristics of the simulated system for the specific channel conditions considered.

### 4.2 Impulsive noise and its model

As the primary aim of this work is to show the performance of hybrid preprocessing combined with Turbo coding to combat IN effect in PLC using simulations, IN needs to be generated using MATLAB. Unlike AWGN, MATLAB doesn’t provide a built in function to add IN, so it was dealt with by using one appealing feature of the noise PDF in MCA model. The PDF is a Gaussian-mixture, which is a sum of weighted zero mean Gaussian terms of different variance, and MATLAB do have an object which can be used to define a Gaussian-mixture. Once the object is created, it can be used to generate IN according to MCA model.

There is an object called gmdistribution in matlab, where ‘gm’ stands for Gaussian-mixture. It is a multivariate distribution which consists of a mixture of one or more multivariate Gaussian distribution components. The number of components for a given gmdistribution object is fixed. Each multivariate Gaussian component is defined by its mean and covariance, and the mixture is defined by a vector of mixing proportions, which corresponds to the weighting factor of the PDF terms.

Another method of the same gmdistribution class can be used to plot the PDF of the generated IN. The IN generated in time domain and the PDF plotted for the
generated IN using gmdistribution, for different $A$ and $\Gamma$ are shown in Figure 4.1 and Figure 4.2, respectively.

Figure 4.1 shows the time domain IN for different $A$ and $\Gamma$. Figure 4.1(a) and (c) correspond to impulsive channels having the same $A$ and different $\Gamma$. It can easily be noted that Figure 4.1(c) has higher amplitudes, with peak value of about 50mV. Whereas, Figure 4.1(a) shows samples with smaller amplitudes, with peak value of around 20mV. From this observation, it can be noted that channel condition of Figure 4.1(c) corresponds to a relatively higher impulsive channels than the channel condition of Figure 4.1(a). Similar pattern is observed comparing Figure 4.1(b) and Figure 4.1(d). The justification for these observations is, the $\Gamma$ value of the lower one is smaller than the other. This value tells the ratio of Gaussian noise power to Impulsive noise power. The smaller the value, the higher the IN power compared to the Gaussian. So smaller values of $\Gamma$ corresponds to more impulsive channels; or the smaller the $\Gamma$, the higher the impulsiveness of the channel. This is taking the same value of $A$, as it also has its say on channel impulsiveness level.

When we compare the waveforms in row, Figure 4.1(a) and Figure 4.1(b) have the same $\Gamma$, but different $A$. Similarly Figure 4.1(c) and Figure 4.1(d) differ in their $A$ values only. Higher level of channel impulsiveness with smaller $A$ is apparent in both cases. As it was described in the MCA noise model section of this document, $A$ is the number of impulses of a certain width within a certain observation period. The ratio of Gaussian to IN average power being fixed by the $\Gamma$ values, to keep the same amount of noise power ratio, the IN amplitudes will be larger for smaller $A$ and vice versa, assuming the average power of the Gaussian noise component remains unchanged. As can be noted from the two rightmost plots, their density is smaller compared to their neighbors with the same $\Gamma$, but their amplitudes are higher. This tells that, for the same $\Gamma$, the smaller the impulsive index $A$, the higher the level of channel impulsiveness. So impulsiveness of the channel, described by the IN amplitude in the plots, increases as both $A$ and $\Gamma$ decreases and vice versa.

Similar facts can be deduced from the PDF plots of Figure 4.2. The horizontal axis represents the amplitudes of IN samples. As the values of $A$ and $\Gamma$ decrease, the PDF plots become more flattened (i.e the tails become fatter and fatter), which means that the probability of observing higher amplitude noise samples increases as $A$ and $\Gamma$ decreases. The likelihood of stronger impulses increases as the two
Figure 4.1: IN samples generated using MATLAB according to MCA noise model for different levels of channel impulsiveness.
parameter values decrease. So the PDF plots agree with our previous observation that channel impulsiveness increases for smaller $A$ and $\Gamma$, and vice versa.

### 4.3 Error performance of PLC systems with single and multicarrier modulation

Figure 4.3 shows error performance of PLC systems with single carrier BPSK modulation and multicarrier modulation scheme of OFDM. As can be observed from the figure, as long as the $\Gamma$ value is fixed, error performance of OFDM-PLC systems remains almost the same for different values of $A$. The encircled three curves correspond to the same $\Gamma$ with different $A$ values, and it can be noticed that BER performance of OFDM-PLC systems depend on $\Gamma$, but not on $A$ [2],[51]. This dependence on $\Gamma$ only is due to the IN energy spreading characteristic of OFDM, in the frequency domain.

Unlike that of PLC systems with OFDM, error performance of PLC systems to which single carrier (SC) modulation is applied is dependent on values of both $\Gamma$ and $A$. For the same $\Gamma$ value of 0.1, two curves indicating error performance for impulsive index values of 0.1 and 0.3 are included in the figure, and they show...
significantly different error performances. Error performance is better for $A = 0.3$ case, as it is less impulsive than the $A = 0.1$ case.

The reason for this differences in performance is, OFDM distributes IN to all the subcarriers in the frequency domain as IN pulses cover wide range of frequency. So, what matters is the total IN energy not the specific distribution of the pulses in time domain. This is one of the appealing features of OFDM. There is no such IN effect sharing in single carrier modulation.

There are three points of intersection of interest in the figure. As we move from left to right on the figure, the first two intersections are between the BER curve with OFDM and SC curves of two different channel conditions, with the same $\Gamma$ and different $A$. The first intersection between the BER curve with OFDM happens at a lower SNR value for the relatively less impulsive channel condition of the two cases. From this, it can be observed that, OFDM requires higher SNR to overcome the error performance with SC applied as the level of the channel impulsiveness increases. Comparing error performance of the two channel conditions with two
different channel conditions and SC modulation applied, intersection happens at the third (right most) intersection point on the figure. Error performance for the relatively higher impulsive channel condition (\(\Gamma = 0.1\) and \(A = 0.1\)) is better for lower SNR value and beyond the intersection point, performance of SC modulation for the relatively less impulsive channel condition (\(\Gamma = 0.1\) and \(A = 0.3\)) is better.

### 4.4 Optimization of preprocessor thresholds

The output SNR of all preprocessors depend on threshold values selected for preprocessing. Selecting a threshold which provides maximum \(SNR_{out}\), is an essential task before applying preprocessing. Unfortunately, there are no closed form expressions to determine optimum threshold and scaling values for preprocessors. But, they can be determined numerically with numerical softwares [15],[39]. MATLAB is handy for such kind of problems, as a result it is used in this work to determine optimum parameters.

In MCA noise model the channel characteristic is determined by \(A\) and \(\Gamma\). For a fixed \(A\) and \(\Gamma\), not all thresholds give the same \(SNR_{out}\) value. The optimum threshold needs to be chosen wisely. Different threshold values are tried and the corresponding \(SNR_{out}\) values evaluated, and the one threshold that gives maximum \(SNR_{out}\) is decided as the optimum threshold value and is used for preprocessing under similar channel and signal conditions as it was evaluated.

Figures 4.4 and 4.5 show how \(SNR_{out}\) values vary for a fixed channel condition as the threshold values are varied. PLC channel impulsiveness is fixed for all at \(A=\Gamma=0.1\), and SNR values are 5dB and 15 dB for the figures, respectively.

The general trend of \(SNR_{out}\) as a function of threshold values is similar in that as the threshold value increases, \(SNR_{out}\) increases, achieves its maximum value and then falls. The threshold that corresponds to the peak output SNR is the optimum threshold value, and it is the value that gives the best result for that specific channel condition and input signal SNR value. Likewise, optimum thresholds are evaluated for specific \(A, \Gamma\) and input signal SNR combinations.

In addition to the maximum \(SNR_{out}\) trend as a function of threshold values, we can also observe the relative performance of the three preprocessing methods considered. In Figure 4.4, which corresponds to SNR value of 5dB, the maximum \(SNR_{out}\) value for CHP is greater than blanking and clipping. The maximum
SNR\textsubscript{out} achieved by blanking is in turn greater than clipping. This peak values indicate the relative performance of the three preprocessors for this specific channel condition and input signal SNR value. If any of the channel parameters or SNR are changed, we will get different curves with different relative performances. This fact is indicated in Figure 4.5, in which clipping has the best performance followed by conventional hybrid and blanking preprocessors, for the same channel conditions and input signal SNR value of 15dB.

Unlike blanking, clipping and CHP, which requires single parameter optimization, AHP’s performance cannot be shown with the rest in two dimensional figures. It has two parameters to be optimized, as a result of which a three dimensional figure is required to convey similar information. Figure 4.6 shows how SNR\textsubscript{out} varies with the blanking and clipping threshold values. The threshold values corresponding to the peak SNR\textsubscript{out} value, are the optimum values. The smaller value of the two thresholds is the clipping threshold.

From Figure 4.6 the scaling factor value can be evaluated by dividing the blanking threshold value with the clipping threshold. The scaling factor in this case is $2.7/1.6 = 1.6875$, different from a value of 1.4 in CHP, which is used commonly [5]. This difference has made the SNR values achieved by the two hybrid methods different. From Figure 4.4, the maximum SNR\textsubscript{out} value of CHP is 1.104dB, and in
AHP case it is about 1.14dB. This difference in output SNR is the result of letting the scaling factor be a variable to be optimized rather than keeping it fixed.

### 4.5 Performance analysis of different nonlinear preprocessors for IN mitigation

Figure 4.7 shows error performance of an OFDM-PLC channel, to which different nonlinear preprocessors are applied, with $\Gamma = 0.01$ and a range of $A$ values. The SNR of the input signal is fixed at 20dB. As the value of $A$ decreases, channel impulsiveness increases and vice versa. So this analysis is to show the relative performance of the preprocessors considering different levels of channel impulsiveness. The figure shows BER performance of an OFDM-PLC with different preprocessors applied to mitigate IN. As can be seen from the figure, as the value of $A$ decreases (i.e. channel impulsiveness increases), the improvement brought by all preprocessors increases. This is indicated by the widening gap between the preprocessors’ BER curves and the BER curve of the OFDM-PLC system with no preprocessing.
applied (‘no preprocessing’ curve in Figure 4.7). From this it can be noted that for more impulsive channels, all preprocessors are generally more successful than the case with lower levels of impulsiveness.

Comparing blanking and clipping, blanking is better for high levels of channel impulsiveness. This results from the fact that, the smaller the channel impulsiveness $A$, the smaller is the impulse density. In a certain observation period, the number of impulses of a certain width decreases as $A$ decreases. As the average power of the Gaussian noise to IN is fixed by $\Gamma$, smaller $A$ will result in fewer pulses with larger amplitude. The pulses stand out for smaller $A$, and such large amplitude samples are primarily made of IN. The information signal they consist of is very small compared to the energy that such samples acquire form the IN. Blanking such samples will significantly reduce the IN energy in the system, without hurting the useful signal significantly. Due to this reason, blanking is preferable for such cases than clipping. Clipping will reduce the energy of the sample, but it will still leave significant amount of IN energy in the output, preprocessed, sample. The relative performance of blanking and clipping for different levels of channel impulsiveness is also indicated in [5], [15].
From the same figure, we can see that, ultimately, clipping outperforms blanking for large $A$. This is due to the reason that, larger $A$ values correspond to lower levels of impulsiveness, in which the received samples’ energy is no more predominantly obtained from IN. Such samples consist of significant amount of useful signal energy. Blanking them would primarily affect the useful signal. So, clipping, which reduces the energy of the samples results in better $SNR_{out}$, as a result lower BER. For the particular case simulated, the intersection point where the relative performance of blanking and clipping changes occurs at about $A = 0.1$.

The BER performance of CHP is also displayed in the figure. As can be noted, it somehow inherits the performance characteristics of blanking for low $A$ and that of clipping for high $A$. For $A$ values near zero, its performance is close to that of blanking, and as $A$ increases, performance of CHP shows improvement compared to blanking. This indicates that, samples under such channel conditions are better preprocessed by combining blanking and clipping, rather than blanking or clipping alone. Above $A$ values of about 0.125, performance of CHP is between blaking and clipping, still better than blanking. As can be noted from the trend of the blanking, clipping and CHP curves, they approach the BER values of the case with no preprocessing as $A$ increases. So in such cases of high $A$ values (lower
levels of channel impulsiveness), preprocessing BER performance gain is not as significant as it is for small values of $A$ (i.e. highly impulsive channel conditions).

From the figure, we can see that AHP has better performance than both clipping and blanking for small values of $A$, and its BER performance is almost the same as CHP. As $A$ increases, AHP’s performance starts to show significant gain over other preprocessors, including CHP. So for intermediate values of $A$, AHP has the best BER performance relative to other preprocessors. Unlike CHP, AHP’s performance is not overcome by clipping for larger $A$ values. This is indicated by almost overlapping curves corresponding to AHP and clipping. So, AHP marks the lowest BER performance for the overall range of $A$ values.

Another important point observed is, AHP’s performance becomes better than CHP’s as $A$ increases. This is due to the fact that AHP combines both blanking and clipping flexibly, whereas in CHP the two thresholds of clipping and blanking are related by a constant.

If different values of $\Gamma$ and input signal SNR were used, the relative performances would exhibit similar pattern. The range of $A$ values one outperforms the other, the relative improvement of one over the other, intersection between blanking and clipping curves would change. But, the general pattern in terms of relative performance for different levels of channel impulsiveness remains similar.

Figure 4.8 shows the BER performance of an OFDM-PLC system with fixed $A$ and input signal SNR for varying $\Gamma$. Impulsive index $A$ is fixed at 0.1 and SNR of input signal is set at 20 dB.

In the figure, the relative performance of all preprocessors relative to no preprocessing is displayed. As the value of $\Gamma$ decreases (i.e. channel impulsiveness increases), moving to the left on the horizontal axis, the gap between the 'no preprocessing' curve and all preprocessor curves widens. This shows the improved performance of all preprocessors compared to the absence of preprocessing as channel impulsiveness increases. These gaps between the 'no preprocessing' curve and others increase as $\Gamma$ decreases, which means, the gain of preprocessing increases as channel impulsiveness increases. In other words, the gain attained by preprocessing is more significant when the channel is characterized by higher level of IN energy compared to the background AWGN noise for a fixed impulsive index value.

As channel impulsiveness decreases ($\Gamma$ increases), the gaps between the preprocessor curves and no preprocessing become narrower. For higher values of $\Gamma$ (close to
Figure 4.8: BER of an OFDM-PLC system with preprocessing for varying $\Gamma$. ($A=0.1$, SNR=20dB)

0.1 in the figure), the performance improvement by preprocessors will eventually become negligible. The gain may not be sufficient enough to outweigh the higher level of complexity introduced by the presence of a nonlinear preprocessing device.

Another important point that can be deduced from the same figure is, except clipping, other preprocessors are characterized by a slight increment of BER as $\Gamma$ increases for the fixed value of $A$. They have maximum BER around a $\Gamma$ value of 0.05, beyond which all BERs start to drop. But, even the BER values show slight increment as one moves from left to right, the gaps between the preprocessors’ and ‘no preprocessing’ curves show that preprocessing has significant advantage over the system without preprocessing. Clipping is different from the others in that it has a monotonically decreasing BER for increasing $\Gamma$.

The relative performance of hybrid preprocessors against blanking and clipping can be noted from the figure. For all values of $\Gamma$ considered, both hybrid preprocessors have smaller BER than blanking. The relative gain of the hybrid methods is dependent on the values of $\Gamma$. For small values of $\Gamma$, the performance gain by the hybrid methods over blanking is not very significant. This region corresponds to extremely impulsive channel conditions. Blanking is known to perform better than clipping in such channel conditions [5], [15]. So hybrid methods do not get much
improvement by using clipping alongside blanking in such cases. As $\Gamma$ increases, 
the gain by hybrid methods over blanking increases.

Comparing hybrid methods with clipping, two regions can be identified. For $\Gamma \leq 0.015$, both hybrid methods have better performance than clipping. Beyond the 
specified point, clipping outperforms CHP and starts to close the gap with AHP. 
It achieves almost the same BER performance with AHP for $\Gamma \geq 0.025$. In the 
former region, both CHP and AHP can be used over clipping, and for the other 
region, it is better to use clipping rather than the hybrid methods. One important 
point to note here is that AHP’s performance is not worse than clipping, it is just 
slightly more complex.

The values of $\Gamma$ in which AHP and clipping has better BER performance than 
blanking and CHP correspond to low levels of channel impulsiveness. In such 
conditions, clipping is better than blanking, and the performance improvement 
the hybrid methods attain by including blanking is not significant in this region.

The fact that AHP has always the smallest BER can be attributed to its flexibility 
in terms of threshold and scaling factor values. It encompasses all other prepro-
cessors. Blanking, clipping and CHP can be considered as special cases of AHP. 
For example, blanking alone can be thought of as AHP with the same clipping 
and blanking thresholds. The gap between the two thresholds is zero. So there is 
no clipping, but blanking. Similarly, clipping can be understood as AHP with a 
clipping threshold and very large scaling factor. Very large scaling factor $\alpha$ results 
in the blanking threshold to be much higher than the maximum amplitude of re-
ceived samples affected by IN, so all IN affected samples will be clipped. CHP is 
the special case of AHP when the scaling factor is set to a constant value (1.4 is a 
common value in the literature and it is used in this thesis work too). So whatever 
performance blanking, clipping and CHP can attain, AHP can do the same due 
to its flexibility and ability to appear in the form of the other preprocessors. Its 
flexibility to combine blanking and clipping in different forms enables it to perform 
better than the other three under all channel conditions.

From Figure 4.7 and Figure 4.8, it can be concluded that, hybrid preprocessors 
(CHP and AHP) are capable of achieving smaller BER than blanking and clipping 
in most channel conditions. The improvement brought by them depends on the 
values of $A$, $\Gamma$ and input signal SNR. Depending on the error performance gain, 
either CHP or AHP can be used.
4.6 Performance analysis of Turbo coding for IN mitigation

Figure 4.9 shows BER performance of Turbo coded and uncoded OFDM-PLC systems for varying $\Gamma$, three SNR values and impulsive index of $A = 0.1$. From the figure, two important points can be observed about performance of Turbo coding for impulsive channel conditions, the first of which is, Turbo performance for a specific channel condition (fixed $\Gamma$ and $A$) improves with increasing SNR.

Error performance improvement for a specific SNR, increases as the value of $\Gamma$ increases (i.e as channel impulsiveness decreases). For small values of $\Gamma$, near zero, the performance improvement, for various SNR values, is not very significant. Rapid performance improvement happens at a larger $\Gamma$ value. For SNR=10 dB, the ‘water fall’ region starts at around $\Gamma = 0.032$ whereas, for SNR=15dB, it starts at a smaller value of $\Gamma = 0.01$. This is due to the reason that as $\Gamma$ increases total noise power in the system decreases. By SNR, only the signal to Gaussian noise power ratio is fixed. So, as one moves to the right on the horizontal axis, IN becomes weaker, and this causes significant error performance improvement in the Turbo coded OFDM-PLC system.

![Figure 4.9: BER performance of Turbo coded and uncoded OFDM-PLC system for different values of SNR as a function of $\Gamma$. ($A=0.1$)]
The second point to note is performance improvement brought by Turbo coding over an uncoded system is dependent on SNR and level of channel impulsiveness (both $\Gamma$ and $A$). For SNR value of 10dB, Turbo coding improves performance for $\Gamma$ values above 0.032. The value required by Turbo coding to dominate the performance of uncoded system for SNR value of 15 dB is a smaller value of 0.01. These values correspond to the starting edges of the waterfall regions. This observation can also be explained in terms of signal to Gaussian plus IN ratio. Larger values of $\Gamma$ correspond to larger value of signal to Gaussian plus IN average power ratio, so that coded system for smaller SNR value attains better error performance over uncoded one for less impulsive channels (larger $\Gamma$), and coded systems with larger SNR values attain improves error performance over their uncoded counterparts for relatively more impulsive channels (i.e smaller $\Gamma$).

Figure 4.10 shows BER performance by varying the input signal SNR, for a fixed value of $\Gamma$, but three different values of $A$. The three $A$ values correspond to three different cases. $A = 0.01$ corresponds to the most impulsive channel of all the three cases. Since the $\Gamma$ values are fixed, impulsiveness for the considered three cases depend entirely on impulsive index $A$. $A = 0.1$ can be considered as mildly impulsive and $A = 0.5$ is the least impulsive of all. From the figure, it is apparent that, the higher the impulsiveness of the channel (i.e smaller $A$), the worse the BER performance with the application of Turbo coding. The best performance is for the $A = 0.5$ case, as it is the least impulsive of all. So, what is observed here agrees with the observation made on the plot of BER for various $\Gamma$ in Figure 4.9, which is as channel becomes more impulsive the success of Turbo coding decreases, relative to its performance in less impulsive cases.

From the same figure, it is apparent that, even for different levels of error performance corresponding to different $A$, all curves corresponding to Turbo coding exhibit a far reduced BER compared to the uncoded case. As an example, to achieve a BER value of $10^{-5}$, the uncoded system requires SNR value of around 29dB. To achieve the same level of BER, the coded cases require 18.75dB, 17.5dB and 17.25dB for $A = 0.01$, $A = 0.1$ and $A = 0.5$, respectively. So an energy saving of more than 10dB is achieved by the application of Turbo coding with OFDM-PLC systems. This success in energy saving makes Turbo coding a favorable choice for IN mitigation applications.

Another important point to note from the figure is, for a fixed value of $\Gamma$ and varying $A$, different levels of error performance are obtained using Turbo coding.
This was not the case when OFDM was applied to PLC systems without Turbo coding. OFDM is characterized by an error performance which depend only on IN energy in the system. It was dependent only on $\Gamma$, and for a fixed $\Gamma$, varying $A$ does not change error performance. This is changed with the application of Turbo coding. Even for fixed $\Gamma$, different BER values can be obtained corresponding to different $A$, with the application of Turbo coding.

### 4.7 Performance analysis of Turbo coding plus nonlinear preprocessing for IN mitigation

Figures 4.11, 4.12 and 4.13 show error performance of OFDM-PLC systems with the combined application of four preprocessors each paired with Turbo coding, for different $\Gamma$ and $A$ values. The first observation that can be made from all the figures is, AHP/CHP combined with Turbo coding, which is the proposed
IN mitigation technique in this thesis work, has better performance compared to clipping or blanking preprocessing plus Turbo coding IN mitigation techniques.

The performance improvement brought by Turbo coding depends on the performance of the preprocessor that it is paired with. The reason is Turbo decoding comes after preprocessing, and the SNR output of the preprocessor determines the error detection and correction performance of the Turbo decoder. As it was shown in previous sections, the performance of a preprocessor depends on the level of channel impulsiveness.

For different values of $\Gamma$ and $A$, different levels of error performance enhancement by Turbo coding plus AHP/CHP over the other preprocessors plus Turbo coding are obtained. So, the overall error performance of the dual mitigation technique by pairing Turbo coding with each preprocessor depends on the values of $\Gamma$ and $A$.

Figures 4.11 and 4.12 show performances of different combinations between the four preprocessors considered and Turbo coding. The value of $\Gamma$ is the same for all cases, which is 0.1, and the $A$ values considered for Figures 4.11 and 4.12 are 0.1 and 0.05, respectively. Both values of $A$ considered are taken from a range of typical $A$ found in [38], to simulate two channel conditions having different levels of impulsiveness. From the figures, it can be noted that Turbo coding plus AHP/CHP has a significant error performance improvement over other combinations for the first case of $A = 0.1$ compared to the other. This result is expected remembering the characteristic of different preprocessors under varying impulsive index $A$ shown in Figure 4.7. It was shown that as $A$ becomes closer and closer to zero (i.e as channel impulsiveness increases), the gap between the BER curves of hybrid preprocessors and blanking decreases. In other words, as channel impulsiveness increases due to a decreasing impulsive index $A$ value, the improvement brought by using AHP/CHP with Turbo coding rather than blanking with Turbo coding decreases. This is confirmed by the reduced gap between AHP/CHP plus Turbo coding curve with the other two as $A$ decreases from 0.1 in Figure 4.11 to 0.05 in Figure 4.12. AHP paired with Turbo coding has the best performance and clipping with Turbo coding has the worst performance in both cases.

For the relatively low impulsive case of Figure 4.11, in order to achieve a BER value of $10^{-5}$, Turbo coding plus blanking needs 0.255dB and 0.212dB more energy compared to AHP and CHP each combined with Turbo coding, respectively. These values correspond to the energy saving by using Turbo coding plus AHP/CHP over
Figure 4.11: BER versus input signal SNR of an OFDM-PLC system with different preprocessor plus Turbo coding applied for $\Gamma = 0.1$ and $A = 0.1$.

Figure 4.12: BER versus input signal SNR of an OFDM-PLC system with different preprocessor plus Turbo coding applied for $\Gamma = 0.1$ and $A = 0.05$. 
Turbo coding plus blanking applied to combat IN in OFDM-PLC systems. It can be observed from Figure 4.12 that the energy saving by AHP and CHP plus Turbo coding over blanking plus Turbo coding are 0.086dB and 0.057dB, respectively, which are relatively smaller compared to the energy saving for $A = 0.1$ channel condition. These differences in energy saving indicate that Turbo plus AHP/CHP is very suited to the less impulsive case of Figure 4.11 than the relatively more impulsive channel condition depicted in Figure 4.12.

Another observation from the two figures is, SNR values at which all the considered dual IN mitigation schemes achieve a BER value of $10^{-5}$, is generally less for the second case ($A = 0.05$) than the first ($A = 0.1$). The reason can be attributed to the characteristic displayed in Figure 4.7, which shows that for a fixed $\Gamma$, BER increases as $A$ increases. This characteristic stays the same when preprocessors are combined with Turbo coding.

In Figure 4.13 is displayed the BER performance of the four IN mitigation schemes considered above with the values of $\Gamma$ and $A$ both set to 0.05. These values of MCA parameters can be considered to represent highly impulsive channel condition comparing to the values of $A = \Gamma = 0.01$ taken to represent extremely impulsive channel condition in [14].

Similar to what is observed form the previous two figures, hybrid preprocessors combined with Turbo coding provide improved error performance than blanking and clipping paired with Turbo coding, separately. Clipping has the worst performance for the highly impulsive channel condition considered. AHP and CHP plus Turbo coding require 0.049 dB and 0.027 dB less energy than blanking plus Turbo coding, respectively, to achieve a BER value of $10^{-5}$. Again AHP plus Turbo has the best performance followed by CHP combined with Turbo coding. This indicates that hybrid preprocessors paired with Turbo coding provide improved error performance over previously studied IN mitigation schemes of blanking paired with Turbo codes and clipping paired with Turbo codes.

Based on all the figures displayed in this section and explanations made to justify the performances, we can draw one important empirical conclusion: AHP plus Turbo coding has the best performance under all channel conditions, compared to other preprocessor plus Turbo code combinations considered, even if its relative gain in performance varies depending on level of channel impulsiveness. Another important observation is performance gain achieved by Turbo coding plus AHP over Turbo coding plus CHP increases as PLC channel impulsiveness decreases.
Figure 4.13: BER versus input signal SNR of an OFDM-PLC system with different preprocessing plus Turbo coding applied for $\Gamma = \Delta = 0.05$.

Turbo coding plus AHP requires 0.043dB, 0.029dB and 0.022dB less energy than Turbo coding paired with CHP to achieve the above specified BER level, for the cases displayed in Figures 4.11, 4.12 and 4.13, respectively.

### 4.8 Performance analysis of Turbo coding plus AHP considering multipath effect

Alongside IN, multipath induced frequency selective fading is a major challenge in PLC. Since the main focus of this thesis work is IN mitigation, in previously displayed simulation results, multipath was not considered. In this section, the effect of multipath on error performance in relation to the application of proposed Turbo coding plus AHP for IN reduction is briefly presented.

An OFDM-PLC system with channel having four multipath components is considered, and the multipath channel is modeled using parameters given in [29], which are used to determine the frequency response.

Attenuation parameters: $k=1$, $a_0=0$, $a_1 = 7.8 \times 10^{-10}$s/m.

Path parameters are specified in Table 4.2 [29].
Table 4.2: Parameters of four-path model of a PLC channel.

<table>
<thead>
<tr>
<th>i</th>
<th>$g_i$</th>
<th>$d_i/m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.64</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>0.38</td>
<td>222.4</td>
</tr>
<tr>
<td>3</td>
<td>-0.15</td>
<td>244.8</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>267.5</td>
</tr>
</tbody>
</table>

Figure 4.14: Performance of different IN mitigation schemes for OFDM-PLC systems with and without multipath. ($\Gamma = 0.1$ and $A = 0.1$)

Error performance of an OFDM-PLC system with four multipath components, using Turbo coding plus AHP and Turbo coding plus blanking IN mitigations is displayed in Figure 4.14. Error performance of the same system using Turbo coding alone, and error performance of a system under the same channel conditions, neglecting multipath characteristic, as it was done in previous error performance evaluations, is also shown in the figure.

The first observation that can be made from the figure is, Turbo coding plus AHP significantly improves error performance of the system, compared to Turbo coding plus blanking, Turbo coding alone and AHP applied alone. An energy saving of about 1dB is achieved by Turbo coding plus AHP compared to Turbo coding plus blanking, and compared to Turbo coding alone, an energy saving of about 5dB is achieved by the proposed dual IN mitigating scheme, whereas compared to
AHP applied alone an energy of about 14dB is saved, taking a BER value of $10^{-5}$ for comparison.

To achieve the specified BER level with Turbo coding plus AHP, the realistic channel with multipath needs an SNR value of 5.5 dB, whereas without multipath an SNR value of 1.75dB is required. The difference between these two SNR values shows, there is a wide room for error performance improvement, if multipath mitigation techniques are used alongside the proposed IN mitigation schemes, which can be pursued as a future work.
Conclusion and Recommendations

In the previous chapter, simulation results are displayed and discussed, based on which some concluding remarks follow. The scope of the thesis work is limited, as clearly indicated in section 1.8, so as to complete it within the allocated time frame. Using this work as a starting point, other related works can be done. Some recommendations for future work are pointed out next to the conclusion part.

5.1 Conclusion

IN is the primary challenge in PLC, so it needs to be mitigated. The proposed mitigation scheme in this work is hybrid nonlinear preprocessing plus Turbo coding. The first and most important conclusion that can be drawn is, compared to blanking or clipping paired with Turbo coding, hybrid methods (CHP and AHP) paired with Turbo coding provides better error performance in most PLC channel conditions. With a little more complexity AHP plus Turbo coding provides the lowest BER for all channel conditions compared to other commonly applied preprocessor plus Turbo coding IN mitigation techniques.

The relative gain in error performance by using the proposed IN mitigation techniques over previously applied Turbo coding plus clipping and Turbo coding plus blanking methods depends on channel condition, as it was described in the results and discussions part. The choice of the specific mitigation scheme thus depends on
channel condition, BER level required and energy constraint issues. Complexity is also a factor in choosing which specific scheme to apply to an OFDM-PLC system as hybrid preprocessing plus Turbo coding is slightly more complex than the other pairs mentioned above.

Generally, hybrid methods are slightly more complex than clipping and blanking. So, Turbo coding plus hybrid methods is accompanied by a relatively higher level of complexity. But, the rise in complexity is not too much, as both blanking and clipping are simple processing methods. As a result, hybrid methods made from them are not too complex, and as processing capability of devices is continuously improving and becoming more energy efficient, the complexity can be outweighed by the energy saving it provides to achieve a certain level of BER.

AHP plus Turbo coding is the scheme accompanied with highest level of complexity, due to AHP’s need to optimize two parameters. But, this task of optimization does not need to be done every time there is transmission of data. Once optimum thresholds are determined for a specific PLC channel condition, they are applied as long as the channel and input signal SNR conditions does not change significantly. In this respect, once optimization is done, there is not much difference between AHP and other preprocessors.

Another important issue with regard to increased level of complexity brought by optimization of two parameters in AHP is, optimization does not need to be part of the OFDM-PLC system. Optimization is more conveniently done on a separate platform. As the parameters were determined using a laptop with MATLAB, the same thing can be done and the resulting optimum parameters can be fed to the preprocessor unit.

In general, the proposed IN mitigation schemes can save significant amount of energy to attain a specific BER level compared to other previously proposed and applied Turbo coding plus preprocessing schemes, so that the performance of OFDM-PLC systems can be enhanced accordingly.

5.2 Recommendations for future work

Below are some recommendations for future works.
• **Computational complexity:** Using hybrid preprocessors with Turbo code ensures better performance accompanied by a rise in complexity. The computational complexity brought by the proposed technique compared to previous techniques can be pursued as a future work.

• **Experimental Validation:** Experimental validation of the results in this thesis work can be pursued to have better insight into practical aspects. Experiments can be conducted to determine BER performance, processing delay and other issues.

• **IN mitigation for PLC channels with memory:** As it was clearly stated in the scope part, the IN mitigation performance study is conducted for memoryless PLC channels. But, PLC channels with memory are common, and they are characterized by burst IN pulses [2], [12]. The performance of the proposed dual IN mitigation schemes can be studied for PLC channels with memory.

• **Highly impulsive channels other than PLC:** The proposed application of AH-P/CHP plus Turbo coding for PLC channels can be extended to other communication systems characterized by channel impulsiveness. A typical example of such a system is space communication. Solar flare, thunder storm and other natural causes result in highly impulsive conditions. In such cases, using the proposed schemes can be considered as a viable option.

• **Multipath mitigation combined with the proposed IN mitigation schemes:** Multipath is an equally important source of error in OFDM-PLC. As the primary objective of this thesis work is IN mitigation, the performance of the proposed dual mitigation techniques is studied with varying levels of channel impulsiveness, fixing the multipath characteristic of the channel. Performance study with different multipath induced frequency selective channel conditions, performance of the proposed IN mitigation scheme with multipath mitigation methods can be pursued.
References


