Impulsive Noise Mitigation Using Hybrid Nonlinear Preprocessor and Turbo Code in OFDM-PLC System

Fikreselam Gared Mengistu[1](✉) and Girma Gebeyehu Amlaku[2]

[1] Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia
[2] Institute of Technology, University of Gondar, Gondar, Ethiopia

Abstract. Power-line communication (PLC) reuses electric signal carrying power-lines to carry information bearing signals. Since PLC channels are primarily designed to carry low frequency electric signal, they pose harsh conditions for transmission of high frequency communication signals. Impulsive noise (IN) and frequency selective fading, due to multipath characteristic of PLC channels, are the primary challenges in PLC. Orthogonal frequency division multiplexing (OFDM) is successful in reducing the effect of both challenges up to a certain level of noise energy in the system, beyond which it needs to be augmented by other mitigation techniques to reduce error, which is primarily caused by IN. In this paper, Turbo codes and hybrid nonlinear preprocessors are proposed to be applied together to combat IN in OFDM-PLC systems. Error performance of an OFDM-PLC system with the proposed scheme applied is investigated for various levels of PLC channel impulsiveness. It is found that Turbo code plus hybrid preprocessors achieve better error performance than previously applied IN mitigation techniques of Turbo code plus blanking and Turbo code plus clipping nonlinear preprocessors.

Keywords: Blanking · Clipping · Hybrid nonlinear preprocessor · IN · OFDM-PLC · Turbo code

1 Introduction

Power-line communication is a technology which reuses power-lines to carry communication signals. Even though, it enables the reuse of already existing media, power-line channels provide hostile conditions for high frequency communication signals. The primary challenges in PLC are additive noise and multipath induced frequency selective fading [1,2].

Unlike other communication systems, noise in PLC cannot be modeled as additive white Gaussian noise (AWGN), rather it is broadly classified into IN and background noise. Impulsive noise has at least 10–15 dB more power spectral density than the background noise, which can be considered AWGN [3,4].

M. A. Delele et al. (Eds.): ICAST 2020, LNICST 384, pp. 143–156, 2021.
https://doi.org/10.1007/978-3-030-80621-7_11

Middleton class A (MCA) noise model is commonly used to model noise in PLC. MCA noise model gives the probability density function (PDF) of noise in PLC as an infinite series, where each term represents a weighted zero-mean Gaussian distribution [5,6].

IN causes significant error in PLC systems, as a result of which several IN mitigation techniques have been proposed. OFDM, being successful in reducing the effect of multipath induced frequency selective fading and IN, is considered a primary shield against both challenges [2,6,7]. It enables IN mitigation by distributing noise energy among subcarriers, making the noise effect on a single subcarrier minimized. So, it is common practice to use OFDM in PLC, giving rise to OFDM-PLC systems. But, as IN energy in the system exceeds a certain threshold, the noise energy distributing effect of OFDM becomes a disadvantage, as all subcarriers will be affected by significantly high levels of IN energy. In such cases, other IN mitigation techniques are needed alongside OFDM. In General, IN mitigation techniques can be categorized into classes of nonlinear preprocessing, error correcting codes and iterative methods. It is a common practice to combine two or more IN mitigation schemes to achieve a required level of error performance [1].

In [2], blanking/clipping preprocessors, with optimized blanking and clipping thresholds, are proposed to combat IN in OFDM-PLC systems. The proposed nonlinearity, named adaptive hybrid preprocessor (AHP), is shown to perform better than blanking, clipping and conventional blanking/clipping hybrid nonlinear preprocessors. It is found that, with a slight increment in level of complexity, AHP provides the best error performance, considering different levels of channel impulsiveness, compared to the other preprocessors. When it comes to error correcting codes, iteratively decoded codes are of primary choice to combat IN in PLC [1]. In [8], Turbo codes are proposed to combat IN in an impulsive environment. It is found that Turbo codes can achieve a significant improvement in error performance compared to convolutional codes selected for comparison. Low-density parity-check codes are also applied to mitigate IN in OFDM-PLC systems [9,10]. Iterative methods are proposed to reduce the effect of IN in [11,12].

Nonbinary Turbo codes and simple nonlinear preprocessors are applied jointly in [6] to combat IN in OFDM-PLC systems. The nonlinear preprocessors considered are blanking and clipping separately. The performance of the dual IN mitigation schemes is studied for a very impulsive channel condition, and nonbinary Turbo codes are found to perform better than their binary counterparts, both being combined with blanking and clipping. By replacing blanking and clipping by a relatively more complex hybrid blanking/clipping preprocessor, error performance can be improved. In this paper, binary Turbo codes and hybrid nonlinear preprocessors are proposed to be applied together to reduce the effect of IN in OFDM-PLC systems. The relative error performance of the proposed IN mitigation scheme is studied under different levels of channel impulsiveness.

The rest of the paper is organized as follows. In Sect. 2, noise model in PLC is discussed. In Sect. 3, proposed system model is given and components are described. Results and discussions are given in Sect. 4. Finally, in Sect. 5 conclusions are drawn.

2 PLC Noise Model

MCA noise model is commonly used to model noise in PLC systems, and its PDF is defined as [1]

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

Where the variance σ_m^2 is given by

$$\sigma_m^2 = \sigma_u^2 \left(\frac{\frac{m}{A} + \Gamma}{1 + \Gamma}\right), \tag{2}$$

and

$$\sigma_u^2 = \sigma_G^2 + \sigma_I^2, \qquad \Gamma = \frac{\sigma_G^2}{\sigma_I^2}, \tag{3}$$

The parameters σ_G^2 and σ_I^2 are variances of Gaussian noise and IN, respectively. Γ is the background (Gaussian) to IN average power ratio, and A, called impulsive index, represents the density of a certain width pulses in an observation period. It increases the impulsive behavior as it becomes smaller and conversely the noise becomes Gaussian when it is larger [1,6].

The impulsive index A is given as

$$A = \frac{n\tau}{T_0} \tag{4}$$

where n is the number of pulses, τ is the average duration of each pulse and T_0 is the observation period, which is commonly set to unity. A is always less than or equal to one, because it is defined as a fraction of time occupied by pulses in an observation time. Even if $n\tau$ is greater than T_0, the value of A cannot exceed one. Figure 1 [1], shows a simple diagram depicting an impulsive index value which is three times the average duration of impulses.

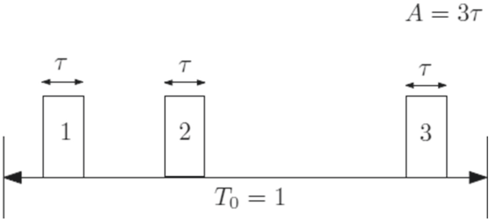


Fig. 1. Impulsive index (A) of three impulses.

3 Noise and System Model

Figure 2 shows the proposed system to mitigate IN. Input bits, arranged in blocks, are Turbo encoded and mapped into baseband symbols S_k using binary phase shift keying (BPSK) modulator. The output symbols of the modulator are then passed to an OFDM modulator, which is commonly executed using Inverse Fast Fourier Transform (IFFT) to result a complex baseband OFDM signal given as [2,6]

$$s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k e^{\frac{j2\pi kt}{T_s}}, \quad 0 < t < T_s \tag{5}$$

where N is the number of sub-carriers and T_s is the active symbol interval. The resulting complex OFDM signal is then converted into analog form and passed to a PLC channel after proper filtering is performed.

At the receiver side, after filtering and conversion to discrete form by sampling, the received signal is processed by a hybrid nonlinear preprocessor to reduce the effect of IN. Hybrid nonlinear preprocessors combine the effects of blanking and clipping to get enhanced performance in reducing the noise energy, which primarily comes from IN. After preprocessing by hybrid nonlinearities, conversion to digital form is performed followed by OFDM demodulation, which is commonly executed using Fast Fourier Transform (FFT). Next comes BPSK demodulator, which passes soft outputs to the Turbo decoder, which intern detects and corrects as many errors as possible and outputs bits.

The proposed system is basically a PLC system which uses OFDM, to make an OFDM-PLC system, to which hybrid nonlinear preprocessor and Turbo code are applied to reduce the effect of IN.

3.1 Nonlinear Preprocessors

Simple processors which replace received noise affected samples with zero and clip amplitudes of samples when the amplitudes exceed certain thresholds, which are called blanking and clipping respectively, are commonly applied to OFDM-PLC systems to mitigate IN. It is also found that, combined application of the two methods generally results in a better performance. Such combined blanking/clipping nonlinearities are called hybrid nonlinearities, in which two thresholds are used to clip, blank or pass unchanged samples. Two types of hybrid nonlinearities, called conventional and adaptive, can be identified [2]. Below are described nonlinearities commonly used in OFDM-PLC systems.

Blanking: refers to the task of removing or nulling samples whose amplitudes are greater than a certain threshold, while samples with smaller amplitudes are left unchanged. Blanking preprocessors are given as [13]

$$y_k = \begin{cases} r_k, & |r_k| \le T_b \\ 0, & |r_k| > T_b \end{cases} \tag{6}$$

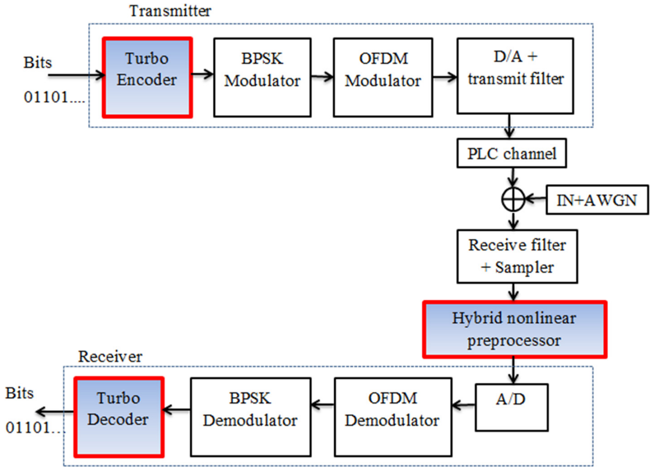


Fig. 2. OFDM-PLC system with the proposed IN mitigation.

Where T_b is blanking threshold, $k = 0, 1, ..., N - 1$, 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 unchanged. Clipping involves changing amplitudes only with phases of samples unaffected. The operation of clipping preprocessors is given as [13]

$$y_k = \begin{cases} r_k, & |r_k| \leq T_c \\ T_c e^{j\ arg(r_k)}, & |r_k| > T_c \end{cases} \tag{7}$$

where T_c is clipping threshold, and $arg(r_k)$ is angle of the input sample r_k to the clipping preprocessor.

Conventional hybrid preprocessing (CHP): is a hybrid (blanking/clipping) 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. Samples with amplitudes less than clipping threshold are passed unchanged. In CHP, blanking threshold value is a constant times clipping threshold, and 1.4 is a commonly used constant. CHP is given as [2]

$$y_k = \begin{cases} r_k, & |r_k| \leq T_c \\ T_c e^{j\ arg(r_k)}, & T_c < |r_k| \leq T_b \\ 0, & |r_k| > T_b \end{cases} \tag{8}$$

Where T_c and T_b are clipping and blanking thresholds, respectively, and usually $T_b = 1.4T_c$.

Adaptive Hybrid Preprocessing (AHP): is similar to CHP in that it employs both clipping and threshold. The difference is, in AHP clipping and blanking thresholds are not related by a fixed value. The constant of proportionality, also called scaling factor, between them is treated as a variable, and it is optimized alongside the clipping threshold to get maximum output signal-to-noise ratio (SRN). AHP is given as [2]

$$y_k = \begin{cases} r_k, & |r_k| \le T \\ Te^{j\ arg(r_k)}, & T < |r_k| \le \alpha T \\ 0, & |r_k| > \alpha T \end{cases} \tag{9}$$

Where T is clipping threshold and $\alpha > 1$ is scaling factor. In this case blanking threshold is αT.

The output signal to noise ratio (SRN) of a preprocessor is given as [13,14]

$$SNR_{out} = \frac{E[|K_0 s_k|^2]}{E[|y_k - K_0 s_k|^2]} = \left(\frac{E_{out}}{2K_0^2} - 1\right)^{-1} \tag{10}$$

where K_0 is an appropriately chosen scaling factor given as $K_0 = E[y_k s_k^*]/E[|s_k|^2]$, $E_{out} = E[|y_k|^2]$ and s_k is the useful OFDM signal.

Since different threshold values of a preprocessor results in different values of SNR_{out}, determining the specific threshold value that maximizes SNR_{out} is a crucial task. In blanking, clipping and CHP, single threshold values are optimized. Whereas, in AHP two optimum threshold values that jointly provide maximum SNR_{out} need to be determined.

In [13], mathematical analysis of different nonlinearities in terms of the maximum SNR_{out} in relation to different threshold values is presented. In the paper, it can be noticed that the mathematical descriptions are well supported by simulation results.

3.2 Turbo Codes

Turbo codes are powerful codes that are known to achieve near Shannon limit performance [15]. They are among the error correcting codes used to mitigate IN in OFDM-PLC systems, due to their enhanced capability to detect and correct errors caused by transmission over the harsh PLC channel.

Turbo encoders are parallel concatenated encoders separated by an interleaver. Recursive systematic convolutional (RSC) encoders are commonly used as component encoders and pseudo-random interleavers are usually chosen over other interleaver types [16]. Figure 3 shows a typical Turbo encoder, which is used in this work.

Just before a block of input bit sequence X_i is encoded, both RSC encoders are brought to the all-zero state by moving the switches downwards, and operating the encoders until the shift registers contain all zeros. Once they are in an initial all-zero state, they are ready to perform Turbo encoding. The upper RSC

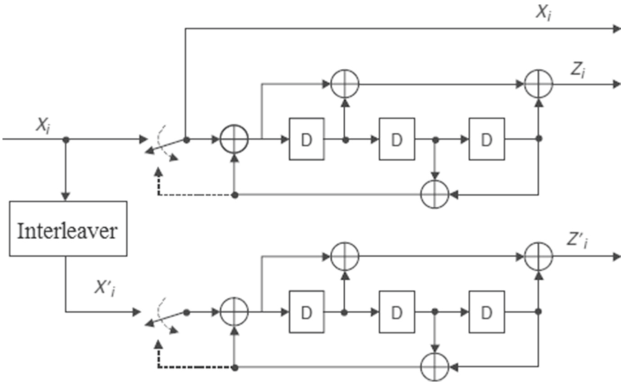


Fig. 3. Turbo encoder

encoder acts on X_i as it is, where as the lower one will encode $X_i^{'}$, which is an interleaved version of input bit sequence X_i. Both encoders act on the same set of bits, but in different order. This difference in sequence reduces the chance of getting low weight encoded outputs from both encoders at the same time. This is one of the key features that make Turbo codes powerful.

The first output of the upper encoder, which is called systematic output, is equivalent to the input sequence X_i. Another output of the same encoder is Z_i, which is a parity sequence. The lower RSC encoder receives $X_i^{'}$, which is X_i interleaved, and outputs parity sequence $Z_i^{'}$. The systematic output of the lower encoder is suppressed. Thus, the final output of the Turbo encoder is an interleaved version of the three sequences X_i, Z_i and $Z_i^{'}$, giving a coding rate of 1/3.

Turbo decoding is an iterative process to estimate transmitted bits from the received bits that are affected by additive noise in the transmission process. 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 is also called Maximum a Posteriori (MAP) or forward-backward algorithm. Detailed explanation of MAP algorithm can be found in [16] and [17].

4 Results and Discussion

In this section simulation results depicting performance of the proposed dual IN mitigation technique (i.e. Turbo code plus AHP/CHP) are presented, accompanied by respective discussions. To show performance improvement attained by the proposed scheme, error performance of Turbo code paired with blanking and clipping preprocessors separately are also included. Three different channel conditions are considered to show relative performance of the proposed scheme compared to Turbo code combined with blanking and clipping separately under varying channel conditions. As the primary aim of this work is IN mitigation, multipath characteristic of PLC is not considered. Input sequence of bits are divided into Turbo blocks of 1020 bit size, which are encoded by the Turbo encoder shown in Fig. 3. BPSK modulation is used to perform mapping of coded bits into symbols which are then passed to OFDM system with 3072 subcarriers. At the receiver side, after preprocessing, Turbo decoding is performed using MAP algorithm with 5 iterations. Bit error rate (BER) as a function of bit energy per noise power spectral density (Eb/No) are plotted for all Turbo code plus preprocessor combinations for the considered channel conditions, and comparisons are made.

At the receiver side, as processing by a nonlinearity is performed before Turbo decoding, performance of a Turbo code is dependent on performance of a preceding nonlinearity. If a nonlinearity results outputs of higher SNR, a Turbo decoder can detect and correct more errors than when it receives a lower quality signal form the preceding preprocessor unit. Due to this dependence, error performance of preprocessors applied to OFDM-PLC systems is studied first, the results of which are shown in Fig. 4 and Fig. 5, under varying channel conditions. After error performance of nonlinearities is explained, performance analysis of OFDM-PLC systems to which nonlinear preprocessor plus Turbo code IN mitigation techniques are applied is undertaken.

4.1 Error Performance of OFDM-PLC Systems with Preprocessors Applied

Figure 4 shows error performance of blanking, clipping, CHP and AHP applied to an OFDM-PLC system for varying A with fixed values of Γ and SNR. For a fixed value of Γ, channel impulsiveness depends on A, and impulsiveness decreases as A increases. Without any nonlinearity, it is shown that BER remains the same for varying A, because in OFDM error performance depends on the total noise energy in the system, not on how it is distributed [1]. Due to this reason, BER is insensitive to change of A for a fixed Γ and input signal SNR.

Looking at the general trend of error performance of all nonlinearities, it can be noted that all result in lower BER for lower A values which correspond to highly impulsive PLC channels. As channel impulsiveness decreases with increasing A, performance of all preprocessors deteriorates and becomes closer and closer to the case where no nonlinearity is applied to the OFDM-PLC system. The reason can be explained remembering the definition of A. If the observation period is set to one, A is then the product of number of pulses and average duration of pulses. So, A decreasing can result from number of pulses decreasing for a fixed average duration of pulses, pulses becoming narrower with the same number of pulses, or both number and average duration of pulses decreasing. If either or both number and/or average duration of pulses decreases, samples affected by IN become easily identifiable. So for smaller A, nonlinear preprocessors can easily identify and process IN affected samples, resulting in better performance of preprocessors in such channel conditions. On the other hand, as A increases for a fixed Γ, number and/or average duration of pulses increases,

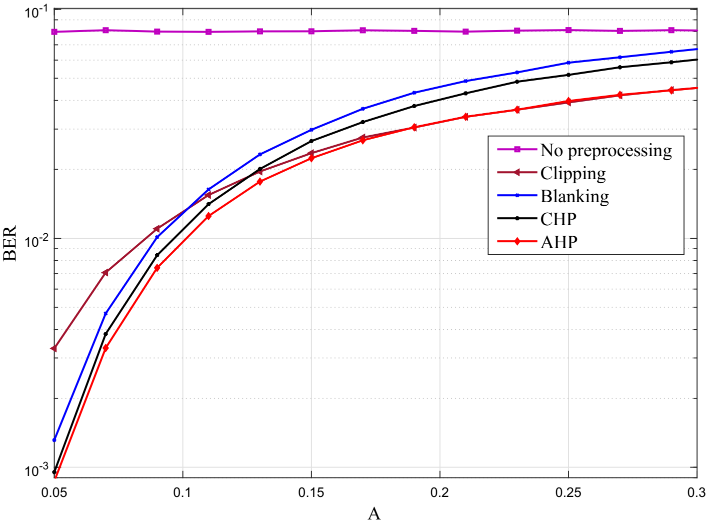


Fig. 4. BER as a function of A of an OFDM-PLC system to which different preprocessors are applied when $\Gamma = 0.1$ and $SNR = 20\,\text{dB}$.

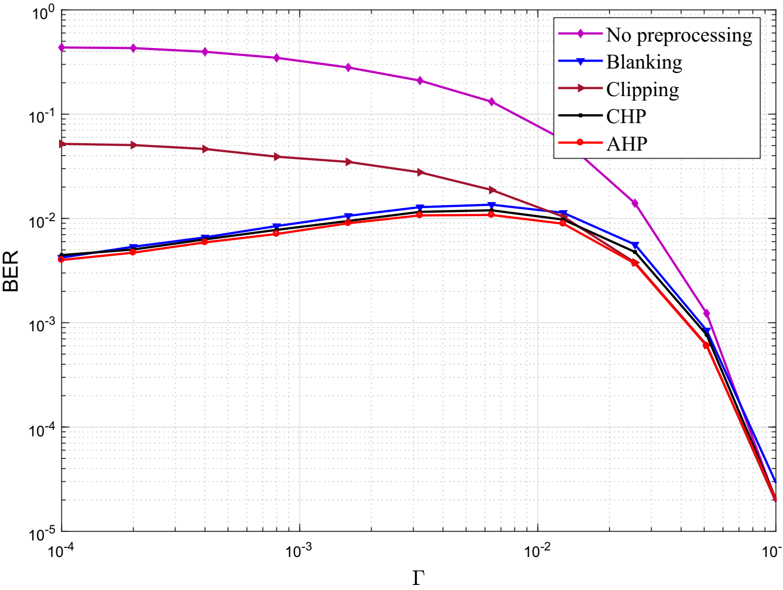


Fig. 5. BER as a function of Γ of an OFDM-PLC system to which different preprocessors are applied when $A = 0.1$ and $SNR = 20\,\text{dB}$.

making it more difficult for preprocessors to identify samples which are affected by IN, as a result of which success of all preprocessors declines.

Figure 5 shows error performance of the four preprocessors considered applied to an OFDM-PLC system as a function of $\varGamma$ for fixed values of A and input signal SNR. Unlike the constant BER curve displayed in Fig.4, there is change in BER for varying $\varGamma$, because for a fixed A, change in $\varGamma$ means change in amplitude of impulses with the distribution of pulses being similar. As $\varGamma$ increases, IN power in the system decreases assuming a fixed AWGN noise power. This decrement in IN noise energy results in smaller BER, as it is shown in the figure. Considering performance of the nonlinearities, their performance compared to the OFDM-PLC system without nonlinearity decreases as $\varGamma$ increases, which is shown by a narrowing of the gap between the curves corresponding to nonlinearities and the curve corresponding to the OFDM-PLC system with no preprocessing. The reason for the decline in performance of preprocessors as $\varGamma$ increases is, IN affected samples become more difficult to distinguish from their IN unaffected neighbors, since higher $\varGamma$ channels are characterized by pulses with smaller amplitudes.

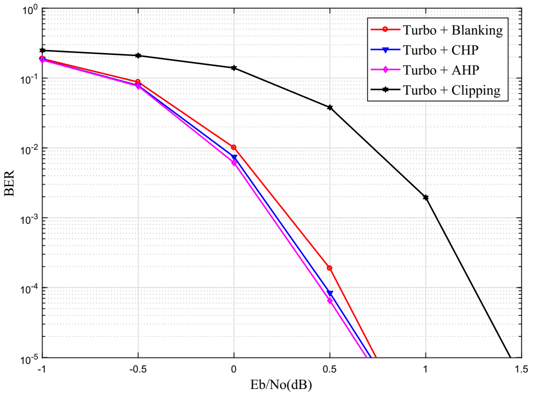
Comparing the four nonlinearities considered, it can be noted form both Fig. 4 and Fig. 5 that hybrid methods (CHP and AHP) have better performance than blanking and clipping for most channel conditions. In both figures, it can be noted that clipping has the worst performance for highly impulsive channels and it shows improvement as channel impulsiveness decreases. Its performance becomes better than blanking and then CHP for higher values of A and $\varGamma$. AHP marks the lowest BER performance for all channel conditions.

Error performance gain by hybrid methods compared to blanking is not significant for very small values of A and $\varGamma$. For such highly impulsive channel conditions, samples significantly affected by IN stand out, and get processed by blanking effectively. Adding clipping effect alongside blanking in such cases does not improve performance that much. But, as both A and $\varGamma$ increase, applying clipping with blanking starts to pay off, as a result of which performance gain by hybrid methods is higher than blanking for higher values of A and $\varGamma$.

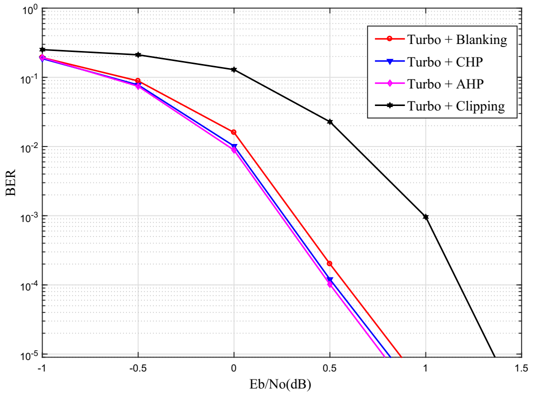
4.2 Error Performance of OFDM-PLC Systems with Turbo Code Plus Preprocessors Applied

Figure 6 displays error performance of an OFDM-PLC system to which all four preprocessors combined with Turbo code are applied in turn to mitigate IN for three channel conditions, with channel impulsiveness decreasing in order from Fig. 6(a) to Fig. 6(c). Of the three channel conditions considered, $A = \varGamma = 0.05$ is the most impulsive one, and $A = \varGamma = 0.1$ is the least impulsive of all, while $A = 0.1$ and $\varGamma = 0.05$ is used to simulate a channel with impulsiveness that lies between the other two.

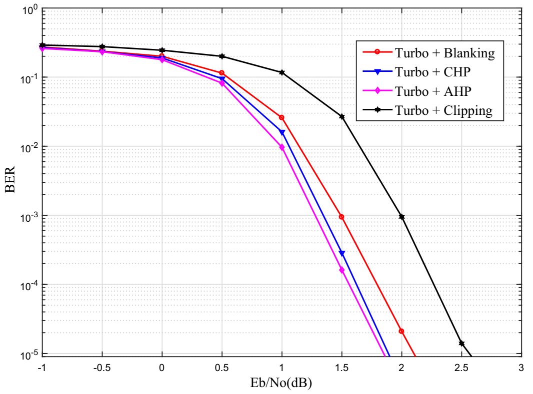
From the figure, it can be noted that hybrid preprocessors combined with Turbo code achieve better error performance than previously proposed IN mitigation techniques of Turbo code plus blanking and Turbo code plus clipping for the channel conditions considered. In all three channel conditions considered, AHP plus Turbo code has the least BER followed by CHP plus Turbo code for a given SNR value. Hybrid methods combined with Turbo code are followed by



(a)$\Gamma = 0.05,\ A = 0.05$



(b)$A = 0.1,\ \Gamma = 0.05$



(c)$A = 0.1,\ \Gamma = 0.1$

Fig. 6. BER versus input signal SNR of an OFDM-PLC system with different preprocessors plus Turbo code applied for different channel conditions.

blanking paired with Turbo code, and clipping plus Turbo code has the worst performance.

Comparing SNR values required by Turbo code plus preprocessor combinations to achieve a BER value of 10^{-5}, energy saved by the proposed IN mitigation scheme (i.e. Turbo code plus AHP/CHP) compared to Turbo code plus blanking is displayed in Table 1.

Table 1. Energy saved by Turbo code plus hybrid preprocessing compared to Turbo code plus blanking to achieve a BER value of 10^{-5} in an OFDM-PLC system.

PLC channel condition		Energy saved by Turbo code plus AHP	Energy saved by Turbo code plus CHP
A	Γ		
0.05	0.05	0.049	0.027
0.1	0.05	0.086	0.047
0.1	0.1	0.255	0.212

From the table it can be noted that, energy saved by Turbo code plus AHP/CHP increases as channel impulsiveness decreases, and vice versa. As a result, it can be said that performance improvement brought by the proposed scheme over Turbo code plus blanking nonlinearity becomes more significant as channel impulsiveness decreases. This results due to the fact that for very impulsive cases, performance of blanking and hybrid nonlinearities are close to each other, and significant performance difference starts to appear as channel impulsiveness decreases, as can be noted from Fig. 4 and Fig. 5. In those figures, BER curves corresponding to blanking, CHP and AHP start being close to each other for small A and Γ and diverge as A and Γ increase, which correspond to lowering of levels of PLC channel impulsiveness.

Another important point is, clipping plus Turbo code error performance narrows the gap with the performance of Turbo code plus other preprocessors, as channel impulsiveness decreases, which is expected remembering that clipping is suited to channels of lower impulsiveness, as displayed in Fig. 4 and Fig. 5.

The energy saved by the proposed IN mitigation technique will increase as the system employing it operates longer and longer. The reason is, the values presented in a Table 1 are only for relatively small number of bits transmitted, as simulation is run until BER reaches a value of 10^{-5}. As an OFDM-PLC system with the proposed IN mitigation technique operates longer, transmitting more and more bits, the energy saved increases accordingly.

5 Conclusion

In this paper, joint application of Turbo code and a hybrid nonlinear preprocessor (i.e. AHP/CHP) is proposed to combat IN in OFDM-PLC systems. BER performance of the proposed scheme is compared to previously applied Turbo code plus blanking and Turbo code plus clipping IN mitigation techniques for various levels of PLC channel impulsiveness, and it is found that the proposed technique enables performance with reduced BER. It is also found that, comparing the two hybrid preprocessors types, AHP achieves better results than CHP when applied to Turbo-coded OFDM-PLC system, due to its flexibility in combining blanking and clipping to reduce the effect of IN. Obtained simulation results are justified by studying BER performance of different preprocessors applied to OFDM-PLC systems for varying A and Γ values.

References

1. Shongwey, T., Vinck, A.H., Ferreira, H.C.: On impulse noise and its models. In: 18th IEEE International Symposium on Power Line Communications and Its Applications, pp. 12–17. IEEE (2014).https://doi.org/10.1109/ISPLC.2014.6812360
2. Rabie, K.M., Alsusa, E.: Performance analysis of adaptive hybrid nonlinear preprocessors for impulsive noise mitigation over power-line channels. In: 2015 IEEE International Conference on Communications (ICC), pp. 728–733. IEEE (2015). https://doi.org/10.1109/ICC.2015.7248408
3. Anatory, J., Theethayi, N.: Broadband Power-line Communications Systems: Theory & Applications. Wit Press (2010)
4. Zimmermann, M., Dostert, K.: A multipath model for the powerline channel. IEEE Trans. Commun. **50**(4), 553–559 (2002). https://doi.org/10.1109/26.996069
5. Middleton, D.: Statistical-physical models of electromagnetic interference. IEEE Trans. Electromagn. Compat. **3**, 106–127 (1977). https://doi.org/10.1109/TEMC.1977.303527
6. Abd-Alaziz, W., Mei, Z., Johnston, M., Le Goff, S.: Non-binary turbo-coded OFDM-PLC system in the presence of impulsive noise. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 2576–2580. IEEE (2017). https://doi.org/10.23919/EUSIPCO.2017.8081676
7. Ghosh, M.: Analysis of the effect of impulse noise on multicarrier and single carrier QAM systems. IEEE Trans. Commun. **44**(2), 145–147 (1996). https://doi.org/10.1109/26.486604
8. Faber, T., Scholand, T., Jung, P.: On turbo codes for environments impaired by impulsive noise. In: 2004 IEEE 60th Vehicular Technology Conference, VTC2004-Fall, vol. 3, pp. 2268–2272. IEEE (2004). https://doi.org/10.1109/VETECF.2004.1400445
9. Nakagawa, H., Umehara, D., Denno, S., Morihiro, Y.: A decoding for low density parity check codes over impulsive noise channels. In: 2005 International Symposium on Power Line Communications and Its Applications, pp. 85–89. IEEE (2005). https://doi.org/10.1109/ISPLC.2005.1430471
10. Hsu, C., Wang, N., Chan, W.-Y., Jain, P.: Improving a power line communications standard with LDPC codes. EURASIP J. Adv. Signal Process. **2007**(1), 1–9 (2007). https://doi.org/10.1155/2007/60839
11. Häring, J., Vinck, A.H.: OFDM transmission corrupted by impulsive noise. In: Proceedings of the 2000 International Symposium on Power-Line Communications and its Applications, Limerick, Ireland, pp. 5–7 (2000)
12. Windyga, P.S.: Fast impulsive noise removal. IEEE Trans. Image Process. **10**(1), 173–179 (2001). https://doi.org/10.1109/83.892455

13. Zhidkov, S.V.: Analysis and comparison of several simple impulsive noise mitigation schemes for OFDM receivers. IEEE Trans. Commun. **56**(1), 5–9 (2008). https://doi.org/10.1109/TCOMM.2008.050391
14. Zhidkov, S.V.: Performance analysis and optimization of OFDM receiver with blanking nonlinearity in impulsive noise environment. IEEE Trans. Veh. Technol. **55**(1), 234–242 (2006). https://doi.org/10.1109/TVT.2005.858191
15. Vucetic, B., Yuan, J.: Turbo Codes: Principles and Applications, vol. 559. Springer, Heidelberg (2012)
16. Andersen, J.D.: A turbo tutorial. TELE (ISSN 1396–1535) (15) (1999)
17. Abrantes, S.A.: From BCJR to turbo decoding: MAP algorithms made easier. ResearchGate (2004)