RESEARCH-ARTICLE

An End-to-End Trainable Power Line Communication System

MIN-SUNG KOH, Eastern Washington University, Cheney, WA, United States

Published: 10 March 2023

ICMLT 2023: 2023 8th International Conference on Machine Learning Technologies
March 10 - 12, 2023
Stockholm, Sweden

ICMLT '23: Proceedings of the 2023 8th International Conference on Machine Learning Technologies (March 2023)
https://doi.org/10.1145/3589883.3589922
ISBN: 9781450398329

An End-to-End Trainable Power Line Communication System

Min-Sung Koh
Department of Computer Science and Electrical Engineering, Eastern Washington University, Spokane, WA 99202, USA,
TEL : 509-828-1304, Email :
mkoh@ewu.edu

ABSTRACT

This paper introduces an end-to-end trainable power line communication (PLC) system based on a deep neural network (DNN). PLC channels are challenging to be modeled and estimated because of uncertain loads in utility / power grids, heterogeneous network topologies, and many equipment such as switching gears, relays, transformers, etc., which are usually not considered in traditional communication systems. With a *trainable* transmitter and receiver, this paper designs a PLC system *not requiring* any power line channel information. The designed PLC system can achieve similar bit error rate (BER) performances to the cases harnessing perfect power line channel information. Since the proposed design can be end-to-end trained without any PLC channel information in diverse power line environments, it will be a promising PLC communication design even for internet of things (IoT) applications targeting for smart grids, smart homes / factories etc.

CCS CONCEPTS

• **Computing methodologies**; • **Hardware**;

KEYWORDS

Machine Learning, Power Line Communication, End-to-End trainable

ACM Reference Format:
Min-Sung Koh. 2023. An End-to-End Trainable Power Line Communication System. In *2023 8th International Conference on Machine Learning Technologies (ICMLT 2023), March 10–12, 2023, Stockholm, Sweden.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3589883.3589922

1 INTRODUCTION

Power line communication (PLC) is a communication utilizing power distribution lines as a communication medium, which is available in these days at any places even at rural sites. Universal availability of already existing power distribution lines is one of the advantages to consider PLC as an alternate inexpensive and competitive option to wireless communication, which is relatively expensive to establish anew. Utilizing power line medium is a very aged idea, which can be traced back up to the late 1800s [1]. Most of applications up to the early 1900s were simple applications such as telecommunication to remotely sense / control various equipment in utility grid such as street lights, power meters etc. [1-3]. Nowadays, PLC is expanded to high speed data communications from several Mbps and to several hundreds Mbps to support smart grid services, internet access, and in-home multimedia applications etc. [1, 3], and it is a promising communication medium to support IoT as well [2, 4, 15].

To achieve robust and reliable PLC communications, there have been lots of research to model PLC channels and to analyze their characteristics in various power grid network environments. To name a few only, those efforts are shown in [5-8]. However, although PLC has many advantages in various applications, it is a challenging task to model an accurate PLC channel in complex utility / power grids because the channel characteristics are time-varying depending on heterogeneous network topologies and interconnected equipment such as switching gears, transformers, relays, load appliances etc. [5]. Hence, a new PLC system without harnessing channel estimation and modeling will open a door for more diverse PLC applications in utility / power grids.

Since the first end-to-end trainable communication system was proposed in [9] using a DNN for additive white Gaussian noise (AWGN) channels, it has been drawing lots of attention in communication technologies. Ever since the first idea for an end-to-end trainable communication system in [9], it has been successfully extended to optical fiber channels [10], and wireless channels [11-14], where both transmitter and receiver are trained to minimize various special loss functions e.g., a special negative achievable rate [12, 14]. Furthermore, the concept of end-to-end trainable communication system is thoroughly verified by both simulations and the over-the-air test as well in [11, 12].

The same concept of end-to-end trainable communication in [9-14] is extended in this paper to a PLC system. A DNN-based communication system similar to [9, 12, 14] is proposed for PLC channels in this paper without employing any PLC channel information. The proposed DNN-based PLC uses multiple layers of residual network (ResNet) to train both transmitter and receiver, which results in an end-to-end trainable PLC system. This paper will show the proposed DNN-based PLC system can achieve the similar BER performances to the scenarios harnessing perfect PLC channel information. The results in this paper imply that the concept of [9-14] can be well extended to challenging PLC channels.

2 DESIGN OF AN END-TO-END TRAINABLE SYSTEM

Overall structure of the proposed trainable PLC system is shown in Fig. 1, which is similar to the structures in [9, 12, 14]. Specially, the similar structure for orthogonal frequency-division multiplexing (OFDM) in [14] is applied in this paper because PLC channels also show undesirable multipath effects [1]. First of all, Channel Encoder

ICMLT 2023, March 10–12, 2023, Stockholm, Sweden

ACM ISBN 978-1-4503-9832-9/23/03…$15.00
https://doi.org/10.1145/3589883.3589922

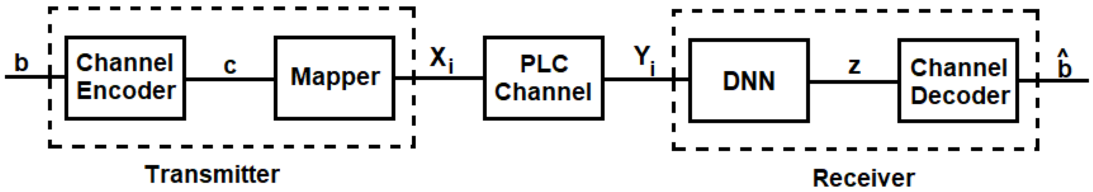

Figure 1: Overall structure of the proposed end-to-end trainable PLC system.

and Decoder in Fig. 1 are not included in the design stage of the proposed system (i.e., Channel Encoder and Decoder are excluded in training but included in testing only). In Fig. 1, therefore, the PLC transmitter is designed by training the mapper expressed by a constellation diagram. A trainable constellation diagram is composed of $M = 2^m$ constellation points, and each constellation point is expressed by both real and imaginary parts of a complex number. It means the trainable mapper changes the given m-bit binary data to one of M complex-value symbols, which can be identified as a constellation point. The mapped complex-value symbols are transmitted to the receiver in Fig. 1 through PLC channels.

In this paper, a DNN is considered to design a trainable receiver because many DNNs working for a "discriminative classifier" have good capability to fit complicated conditional probabilities (or likelihoods), $p(z|b)$, where "b" and "z" are given in Fig. 1 and b=c in training. The complicated conditional probabilities are due to the non-linear function blocks of Mapper, PLC channels, and DNN in Fig.1. Therefore, the DNN in the receiver is simultaneously trained with Mapper in the transmitter to generate proper likelihoods, $p(z|b)$, for diverse PLC channel environments. To accomplish the goal to generate proper likelihoods, a ResNet is chosen as a building block to make the DNN.

In practical design for the DNN, however, it must be considered to embed all non-linear functionalities in Fig. 1 to the DNN using the channel output, Y_i , in Fig. 1 because the Y_i is a practically available input to the DNN. Therefore, the DNN is designed to generate proper log-likelihood ratio (LLR) values denoted by "z" in Fig. 1 for the given Y_i so that those LLR values can contribute to any conventional channel decoder determining the estimated bit-sources denoted by $\hat{b}$.

As shown in Fig. 1, the proposed trainable PLC system is much different from traditional PLC systems because: 1) there is no conventional demapper, which is replaced by a DNN, and 2) the Mapper is also a trainable mapper (i.e., not a fixed mapper in conventional PLC systems).

Designing the DNN for an arbitrary constellation diagram in Mapper is completed by jointly minimizing a special loss (i.e., bit cross-entropy loss which is given in section 2.4). Once the training process is completed, the obtained transmitter and receiver are combined with a conventional channel encoder and decoder to detect and correct bit errors in testing process. Through the testing with a channel encoder / decoder, BER performances of the proposed system are checked for a few challenging PLC channels

In this paper, a low-density parity check (LDPC) encoder and decoder is used for channel encoding and decoding. Many conventional channel decoders such as LDPC decoder show better BER performances with soft decision using LLR values than those with hard decision using signs. However, in general, calculating LLR information requires high computational complexity. In this paper, instead of a traditional demapper, a DNN is used to provide proper LLR values at the beginning of the channel decoding.

2.1 PLC channel models

In traditional PLCs, the channel modeling and estimation is an important issue to achieve a robust and reliable PLC system. Lots of different approaches to model PLC channels have been researched, for instance, as shown in [5-8] to name a few only. Indoor PLC channel is modeled as one-parameter deterministic models in [6]. In [5], indoor PLC channel is modeled by simplified structural parameters and those simplified structural parameters are randomly generated by a statistical distribution to get the ensembles of random channels. Whereas, to model indoor / in-home PLC channels, transmission line theory is applied in [7, 8] with a statistical concept for network parameters such as line length and load distribution [7].

Even with those active research to model PLC channels, it is still challenging to have an universal accurate PLC model to fit diverse PLC environments. Therefore, if a robust and reliable PLC can be achieved *without* any power line channel information, it can be expanded to many PLC applications having different channel environments. In this paper, a PLC system *without* any power line channel information is proposed by a trainable transmitter and receiver utilizing a DNN. The proposed trainable PLC is compared with scenarios having perfect PLC channel information generated by the PLC channel modeling in [5]. As an exemplary case, a network topology is chosen as proposed in [5] and the best, medium, worst PLC channels are generated for the chosen network topology, where those generated channels are for up to 30 MHz considering broad-band indoor PLCs.

The magnitude and phase responses of the generated PLC channels are shown in Fig. 2. The validation of the generated channels in Fig. 2 for indoor PLCs are also given in [5] as well. As shown in Fig. 2, the PLC channels do not have flat frequency response, specially, when it is considered for broad-band PLCs. Those non-flat magnitude responses in broad-band PLC applications imply that transmitting signals in PLC are distorted differently depending on frequencies (i.e., frequency selective channels). Interestingly, the PLC channels show linear phase responses roughly as shown in right plot of Fig. 2, which implies constant group delays. Notice that the generated channels in Fig. 2 are to compare the ideal scenarios having the perfect PLC channel information with the proposed trainable PLC system having no channel information.

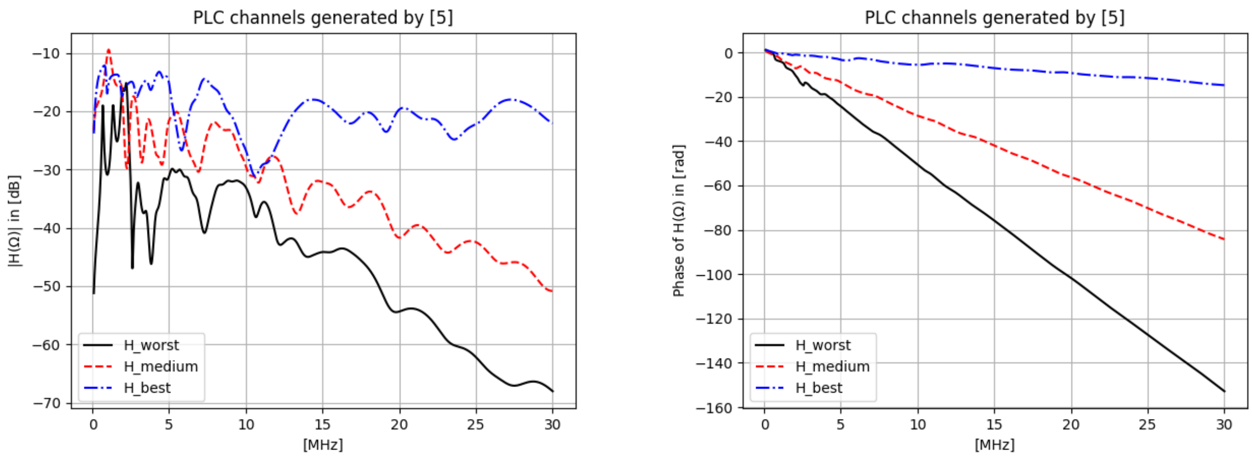

Figure 2: PLC channels generated by [5]. Left and right figures are respectively for magnitude and phase responses of the generated channels, H_best, H_medium, and H_worst.

Furthermore, the PLC channels corrupt the transmitting signals with random noises. In this paper, AWGN is also considered in PLC channel, which means the PLC channels distort the transmitting signals according to the given magnitude / phase responses in Fig. 2 and corrupt the transmitting signals at the same time by adding noises. Consequently, the channel output signal denoted by Y_i in Fig. 1 implies a distorted and corrupted signal by the PLC channels.

2.2 Trainable transmitter for PLC

In this paper, the transmitter in Fig. 1 is designed as a trainable transmitter by utilizing trainable parameters in the mapper so that they can be determined by training process. The mapper in Fig. 1 generates a complex-value symbol denoted by X_i, where $X_i = f_{Mapper}(b)$ (or $X_i = f_{Mapper}(c)$ in testing stage) and $f_{mapper} : b, c \in \{0, 1\}^m \longrightarrow X_i \in \mathbb{C}^1$. It is to map a m-bit binary data, "b", (or m-bit channel coded binary data, "c", in testing stage) to a complex-value symbol out of $M = 2^m$ symbols. Therefore, the mapper has a mapping table, X, where $X = [X_0, X_1, ..., X_{M-1}]^T \in \mathbb{C}^M$ and X_i is a complex-value. The mapped complex-value symbol, X_i, corresponding to a constellation point is transmitted through the PLC channel in Fig. 1. In traditional PLC communications (i.e., without any training capacity), the mapper is used with various constellation diagrams such as binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), M-ary phase shift keying (MPSK), M-ary quadrature amplitude modulation (MQAM) etc.

In this paper, the constellation diagram for M symbols is corresponding to a *learnable* mapper obtained by training process, which means the mapping function "f_{Mapper}" is a learnable function having codomain of M constellation points, X_i, and those M constellation points are obtained by training. To make the mapper have trainable capacity, the mapper is defined as a table having trainable parameters for both real part and imaginary part on a 2D constellation diagram (i.e., $Re\{X_i\}$ on x-axis and $Im\{X_i\}$ on y-axis). Therefore, there are total 2M parameters to be trained in the mapper of Fig. 1 for both real and imagery parts. At the beginning of training process, both real and imaginary parts of the trainable mapper are respectively initialized as random numbers generated by uniform distribution within the range of $[X_{min}, X_{max}]$ as given in eq. 1), where $X^{(k)} = [X_0^{(k)}, X_1^{(k)}, ..., X_{M-1}^{(k)}]^T \in \mathbb{C}^M$ denotes the mapping table in the mapper at the k$^{\text{th}}$ iteration in training. Notice that the mapping table $X^{(k)}$ can be expressed equivalently by $X^{(k)} = [Re\{X^{(k)}\},\ Im\{X^{(k)}\}]^T \in \Re^{2\times M}$.

$$X^{(0)} \sim U\,(X_{min},\ X_{max}) \tag{1}$$

where $X_{min} = -\sqrt{\frac{3}{M}}$ and $X_{max} = \sqrt{\frac{3}{M}}$

Once the trainable mapper is initialized by eq. 1), then the mapper is continuously updated by training process to minimize a special loss function, which is defined in section 2.4. In the updating process, each new constellation diagram is made by geometric shaping (i.e., adjusting its center and normalizing constellation points) as mentioned in [14] but a little differently as shown in eq. 2) and eq. 3). Centering and normalizing the constellation in each training iteration are completed by applying eq. 2) and then eq. 3) sequentially.

$$\tilde{X}^{(k)} = X^{(k-1)} - \frac{1}{M}\sum_{i=0}^{M-1} X_i^{(k-1)} \tag{2}$$

$$X^{(k)} = \tilde{X}^{(k)} / \sqrt{\frac{1}{M}\sum_{i=0}^{M-1}\left|\tilde{X}_i^{(k)}\right|^2} \tag{3}$$

In testing process after training, the obtained mapper (i.e., obtained constellation diagram equivalently) is used without further updating.

2.3 Trainable receiver for PLC

The distorted and corrupted signal, Y_i, by the PLC channels is decoded at the receiver in Fig. 1, where the receiver is a trainable receiver because the DNN is trained to provide a proper LLR values

Table 1: Detail structures of Mapper and DNN in Fig. 1

Mapper	Input : b (or c for testing stage) Output : X_i,	$b,\ \ c \in \{0,1\}^m$ and the mapper has total 2M trainable parameters X_i is a complex-value, where i = 0,1,. . .,M-1
DNN	Input: Y_i, (or both Y_i and SNR) Conv2D	Y_i is the PLC channel output in Fig.1 Conv2D at this layer has 128 output filters and the input is composed by $[Re\{Y_i\},\ Im\{Y_i\},\ Var]$
	$\left\{\begin{matrix}\text{layer normalization}\\ \text{Relu}\\ \text{Conv2D}\\ \text{Layer normalization}\\ \text{Relu}\\ \text{x = Conv2D}\\ \text{xskip = Conv2D(}Y_\text{i}\text{)}\\ \text{Out = xskip + x}\end{matrix}\right\} \times \text{r}$	1. "×r" means { } is repeated "r" times. 2. Each { } block (i.e., r=1) is a ResNet. 3. Conv2Ds in each ResNet has 128 output filters. 4. Conv2Ds in each ResNet use Stride = [1, 1] 5. Kernel size and dilation rate of Conv2Ds in each ResNet are set according to Table 1 in [14]
	Conv2D	Conv2D at this layer has "m" output filters and the output of this layer is corresponding to "z" in Fig. 1.

to the channel decoder. The detail structure of DNN is given in Table 1, which is a similar structure to the neural network-based receiver in [14] (i.e., r=5 is chosen in this paper). Notice that the block denoted by { } in Table 1 is a ResNet and "$\{\ \} \times r$" is implying the ResNet blocks repeated by "r" times to make the DNN.

In Fig. 1, the ultimate goal of training is to fit the DNN to have proper *"posteriori likelihood"* denoted by $p(b|z)$ so that any conventional channel decoder can estimate the original transmitted sources, "b", correctly using the available LLR values, "z". However, instead of posteriori likelihood, $p(b|z)$, the ultimate goal of training can be accomplished by employing the *"likelihood"*, $p(z|b)$, with training binary data sources, "b", because the posteriori likelihood, $p(b|z)$, can be expressed by eq. 4). As shown in eq. 4), the posteriori likelihood, $p(b|z)$, is proportional to likelihood value, $p(z|b)$, where the prior likelihood, $p(b)$, for the binary source "b" is generated by a uniformly distributed function, i.e., $p(b=0) = p(b=1)$. Therefore, it is valid to train the DNN to have proper likelihood, $p(z|b)$, instead of posteriori likelihood, $p(b|z)$. In this paper, the DNN is trained to fit the likelihood, $p(z|b)$, using the available channel output, Y_i, and to generate proper LLR values, z, in Fig. 1 at the end, which means non-linear log functionality to make LLR is also implicitly included on top of the likelihood in the DNN.

$$p\,(b|z) = \frac{p(z|b)p\,(b)}{p\,(z)} \propto p\,(z|b)\,p\,(b) \tag{4}$$

where $z = f_{DNN}(Y_i)$, $\ Y_i = f_{PLCchanel}(X_i)$, and $X_i = f_{Mapper}(b)$,

Notice that LLR values, z, are generated by the DNN in this paper, whereas conventional communication systems generate LLR values through a demapper, which is not trainable. The DNN is to fit well overall non-liner function, $f_{overall}(b)$, at the output of the DNN, z, where $z = f_{DNN}(f_{PLCchannel}(f_{Mapper}(b))) \overset{\Delta}{=} f_{overall}(b)$. It means the DNN must be designed to accommodate all non-linear functions through the mapper, $X_i = f_{Mapper}(b)$, PLC channel, $Y_i = f_{PLCchanel}(X_i)$, and the DNN, $z = f_{DNN}(Y_i)$. It can be accomplished by jointly training the DNN with the mapper using the channel output, Y_i, so that the designed DNN can generate proper LLR values helping the channel decoder at the receiver. The proposed trainable PLC system is distinctly different from conventional PLC systems because it has capacity to accommodate non-linear functionalities caused by the mapper and PLC channels in training process so that reliable PLC communication can be achieved even for various challenging PLC channel environments, which are usually unknown.

In training the proposed PLC system, it must be stressed that if a fixed mapper (i.e., non-trainable mapper) having conventional constellations such as MQAM, MPSK, etc. is required for an application, then the trainable mapper can be replaced by a fixed mapper because a well-trained DNN itself can accordingly absorb all non-linear functionalities given in Fig. 1 regardless of a fixed or trainable mapper. In [12], SNR is used for both transmitter and receiver in training to apply the observation that the posterior distribution is dependent on SNR. Since the DNN uses the noisy channel output, Y_i, in training, it means the training data, Y_i, itself includes implicitly SNR information in training process [12]. In this paper, SNR is given as an input to the *receiver only*, whereas notice that [12] needs SNR information both for transmitter and receiver. Hence, the feedback of SNR in [12] is not required in this paper.

The outputs of DNN, $z \in \Re^m$, are LLR values corresponding to the m-bit binary sources and those LLR values are fed to a conventional channel decoder (e.g., a LDPC decoder in this paper). Like for conventional channel decoders, the LLR values, z, contribute to detection and correction of error bits caused by PLC channels.

2.4 Training for Mapper and DNN

All trainable parameters in the proposed PLC system must be updated by training process. In [9-14], various loss functions are used for training. In this paper, the eventual objective of training is to make the proposed system of Fig. 1 generate proper LLR values, "z", for given sources, "b", so that any traditional channel decoder can recover the transmitted sources using the LLR values. Therefore, a little simplified binary cross-entropy loss function using "b" and "z" is adopted for training as defined in eq. 5), where "B" is the batch

Table 2: Parameter values used in training

Parameter	Value	Note
m	4	i.e., $M = 2^m = 16$
L	1794*CR	L is for the size of $\boldsymbol{b} \in \{0,1\}^{B\times mL}$
B	128	Batch Size
LR	0.001	Learning Rate for Adam optimizer
beta1	0.9	Beta1 for Adam optimizer
Beta2	0.999	Beta2 for Adam optimizer
Iteration	20,000	Number of training iteration

size used for training data and notice that "b" and "z" are respectively binary source and the output of DNN as indicated in Fig. 1. In eq. 5), "$b_{k,l}$" denotes the lth bit in the kth training data of one batch, where $\boldsymbol{b} \in \{0,1\}^{B\times mL}$ and $\boldsymbol{z} \in \Re^{B\times mL}$ are respectively denoting a batch for multiple m-bit binary sources and LLR values. In addition, $\sigma(\cdot)$ means the sigmoid function defined by $\sigma(x) \triangleq 1/(1+e^{-x})$. Mapper and DNN explained in previous sections are jointly trained to minimize the loss function defined in eq. 5).

$$L(\boldsymbol{b},\ \boldsymbol{z}) \triangleq -\frac{1}{mLB}\sum_{k=1}^{B}\sum_{l=1}^{mL} b_{k,l}\ln\left(\sigma\left(z_{k,l}\right)\right)+\left(1-b_{k,l}\right)\ln\left(1-\sigma\left(z_{k,l}\right)\right) \tag{5}$$

Notice that the special loss function using a negative achievable rate in [12, 14] also can be applied for the proposed PLC system to get an optimized mapper and a DNN maximizing information rate. However, because a discriminative classifier making proper likelihoods is required for the training as explained in section 2.3, this paper applies the simple loss function in eq. 5) to minimize bit cross-entropy between LLR values and the given binary sources so that the channel decoder in Fig. 1 can take advantage of the likelihoods in its decoding.

3 EXPERIMENTAL RESULTS

For the three PLC channel environments (i.e.,H_best, H_medium, and H_best in Fig. 2), the proposed end-to-end trainable PLC system is trained and tested by randomly generated binary data sources, $\boldsymbol{b}$, where the loss function in eq. 5) is minimized by the Adam optimizer. All non-trainable parameters used for training are shown in Table 2, where the code rate (CR) is the parameter for a conventional channel encoder and decoder.

As explained in previous sections, the channel encoder and decoder are excluded in training process. The SNR input for DNN is uniformly sampled in the range of [-5, 10], [5, 20], and [12, 27] in [dB] for the best, medium, and worst channel scenarios respectively at each training iteration. Notice the training is also available for a fixed SNR input due to the implicit SNR information included in noisy, Y_i, but with a price getting a reduced BER performance, like [12]. A fixed SNR value can be decided depending on the given environment in an application.

In training process, the trainable mapper in Fig. 1 is continuously updated. Updated constellation diagrams by the mapper are given in Fig. 3 for the H_mediium PLC channel. Left/top plot of Fig. 3 is the constellation diagram initialized randomly using eq. 1). After 1000 and 5000 training iterations, the updated constellations are respectively given in right/top and left/bottom plots of Fig. 3. The final constellation at the end of training is also shown in right/bottom plot of Fig. 3. Interestingly, the final constellation is a skewed MQAM constellation for M=16 compared with the traditional MQAM constellation. And the final constellation shows that a gray coding is also achieved by training because each constellation point has only one bit difference compared with the bit-labels in neighbor constellation points. Those similar phenomena were also observed in [9, 12, 14] for AWGN and wireless communication channels.

After the training is completed, the proposed PLC system is tested by newly generated random binary data with a LDPC channel encoder and decoder to have BER performances. Code rate, CR, for the channel encoding/decoding is set with 0.5. In testing process, BERs for a traditional MQAM (i.e., without any training for a fixed square constellation) are also obtained by utilizing the given perfect channel information in Fig. 2 so that they can be compared with the proposed trainable system. It is shown in Fig. 4 that the proposed trainable PLC system even without any channel information can have very close BER performances to those of the cases harnessing the perfect channel information for all 3 PLC channel scenarios in Fig. 2. The BER curves in Fig. 4 are encouraging results implying the proposed trainable PLC system can be excellently applied to given unknown PLC channels (or challenging PLC channels to be modeled and estimated).

4 CONCLUSIONS

As an extension of recently proposed end-to-end communication concept, an end-to-end trainable PLC system was proposed in this paper by exploiting a DNN and a mapper having trainable parameters. Many PLC systems are challenged by signal attenuation / distortion problems because of uncertain channel environments. To tackle those problems, diverse research to model accurately PLC channels have been conducted so far in PLC technologies. However, this paper has considered a different approach not requiring any PLC channel information with a trainable PLC system. Without any PLC channel information, this paper has shown that an end-to-end trainable PLC system can have a very close BER performance to the cases having perfect PLC channel information. It has been achieved by jointly training a DNN with a mapper in the proposed PLC system. Since the proposed trainable PLC system does not need any challenging PLC channel information, it is expected to be

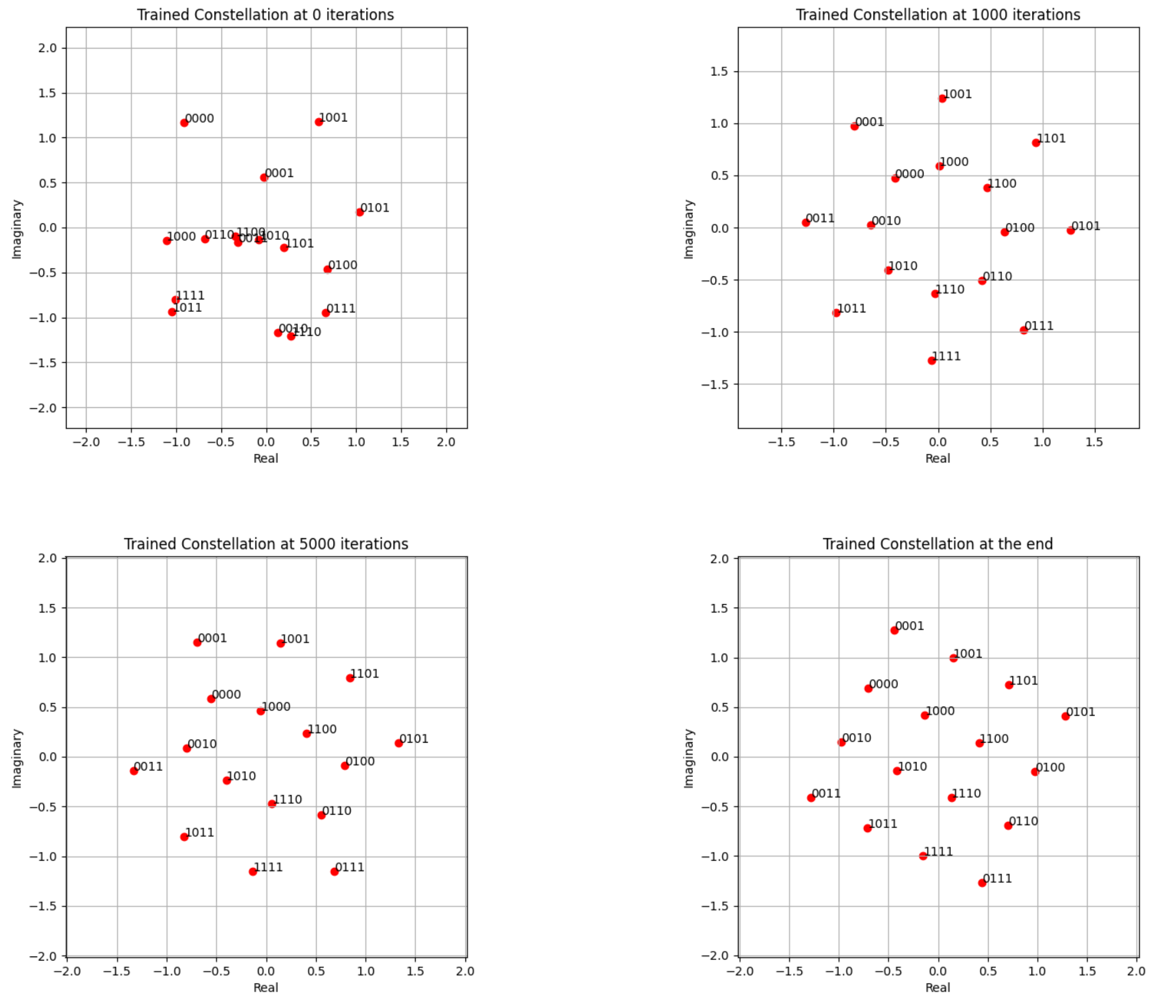

Figure 3: Constellation diagrams obtained by the trainable mapper for H_medium channel in Fig.2. Top/Left: at the beginning of iteration (i.e., randomly initialized). Top/right: at training iteration =1000. Bottom/left: at training iteration=5000. Bottom/right: at the end of training iteration

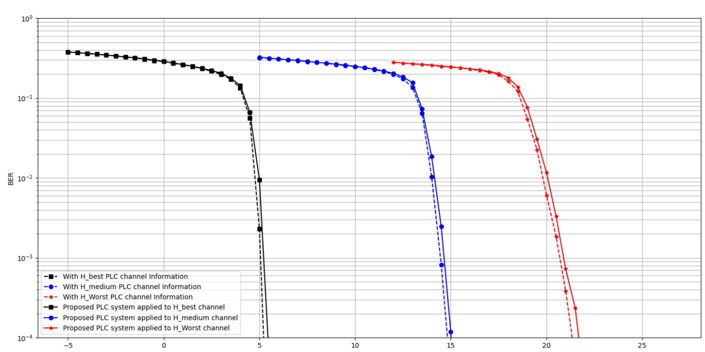

Figure 4: BER performance comparisons of the proposed trainable PLC system with a traditional MQAM system having perfect channel information for 3 PLC channels of H_best, H_medium, and H_worst in Fig. 2, where M = 16.

a promising communication for future PLC including IoT applications targeting for smart homes / factories, smart grids, controller area networks etc.

REFERENCES

[1] L. Lampe, A. M. Tonello, and T. G. Swart. 2016. Power Line communications Principles, Standards and Applications from Multimedia to Smart Grid. 2^{nd} ed., Wiley.

[2] A. O. Aderibole, E. K. Saathoff, K. J. Kircher, S. B. Leeb, and L. K. Norford. 2022. Power Line Communication for Low-Bandwidth Control and Sensing. vol. 37, no. 3, 2172-2181.

[3] S. Galli, A. Scaglione, and Z. Wang. 2011. For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid, Proc. IEEE, vol. 99, no. 6, 998-1027.

[4] A. D. dos Santos, R. M. Bacurau, A. V. Martins, A. Dante, and E. C. Ferreira. 2022. Simple and Low-Cost PLC Modem for IoT Applications. IEEE Latin America Transactions, vol. 20, no. 12, 2455-2462.

[5] F. J. Canete, J. A. Cortes, L. Diez, and J. T. Entrambasaguas, U. d. Malaga. 2011. A Channel Model Proposal for Indoor Power Line Communications. IEEE Comm. Magazine, vol. 49, no. 12, 166-174.

[6] M. Tlich, A. Zeddam, F. Moulin, and F. Gauthier. 2008. Indoor Power-Line Communications Channel Characterization Up to 100 MHz-Part I: One-Parameter Deterministic Model. IEEE Trans. on Power Delivery, vol. 23, no.3, 1392-1401.

[7] A. M. Tonello and F. Versolatto. 2011. Bottom-Up Statistical PLC Channel Modeling-Part I: Random Topology Model and Efficient Transfer Function Computation. IEEE Trans. on Power Delivery, vol. 26, no. 2, 891-898

[8] F. Aalamifar, A. Schlogl, D. harris, and L. Lampe. 2013. Modelling Power Line Communication Using Network Simulator-3. IEEE Global Comm. Conference (GLOBECOM), IEEE. Atlanta, GA. https://doi.org/10.1109/GLOCOM.2013.6831526

[9] T. O'Shea and J. Hoydis. 2017. An Introduction to Deep Learning for the Physical Layer. IEEE Trans. on Cognitive Comm. and Networking. Vol. 3, no. 4. 563-575

[10] B. Karnanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bulow, D. Lavery, P. Bayvel, and L. Schmalen. 2018. End-to-End Deep Learning of Optical Fiber Communications. Journal of Lightwave Tech. vol. 36. no. 20. 4843-4855

[11] S. Dorner, S. Cammerer, J. Hoydis, and S. ten Brink. 2018. Deep Learning Based Communication Over the Air. IEEE Journal of Selected Topics in Sig. Proc. vol. 12. No. 1. 132-143

[12] S. Cammerer, F. A. Aoudia, S. Corner, M. Stark, J. Hoydis, and S. ten Brink. 2020. Trainable Communication Systems: Concepts and Prototype. IEEE Trans. on Comm. vol. 68, no.9. 5489-5503.

[13] H. Ye, Y. Li, and B-H. Juang. 2021. Deep Learning Based End-to-End Wireless Communication Systems Without Pilots. IEEE Trans. on Cognitive Comm. and Networking, vol. 7. No. 3. 702-714

[14] F. A. Aoudia and J. Hoydis. 2022. End-to-End Learning for OFDM: From Neural Receivers to Pilotless Communication. IEEE Trans. on Wireless Comm. vol. 21, no. 2, 1049-1063

[15] D. P. F. Moller and H. Vakilzadian. 2014. Ubiquitous Networks: Power Line Communication and Internet of Things in Smart Home Environments. IEEE Int. Conf. on Electro/Info Technology. 596-601. Milwaukee, WI. https://doi.org/10.1109/EIT.2014.6871832