Ensemble extreme learning machine based equalizers for OFDM systems

Michel Saideh, Eric Pierre Simon, Joumana Farah, Jonathan Villain, Anthony Fleury, Virginie Deniau, Christophe Gransart

Ensemble Extreme Learning Machine Based Equalizers for OFDM Systems

Michel Saideh*, Eric Pierre Simon*, Joumana Farah†, Jonathan Villain‡, Anthony Fleury§, Virginie Deniau‡, Christophe Gransart‡

*Univ. Lille, IEMN lab, France, {firstname.lastname@univ-lille.fr}
†Faculty of Engineering, Lebanese Univ., Roumieh, Lebanon {joumanafarah@ul.edu.lb}
§IMT Lille Douai, CERI SN and Univ. Lille, F-59650 Villeneuve d'Ascq, France. {anthony.fleury@imt-lille-douai.fr}
‡COSYS-LEOST, Univ Gustave Eiffel, IFSTTAR, Univ Lille, F-59650 Villeneuve d'Ascq, France, {firstname.lastname@univ-eiffel.fr}

***Abstract*—Extreme Learning Machine (ELM) technology has started gaining interest in the channel estimation and equalization aspects of wireless communications systems. This is due to its fast training and global optimization capabilities that might allow the ELM-based receivers to be deployed in an online mode while facing the channel scenario at hand. However, ELM still needs a relatively large amount of training samples, thus causing important losses in spectral resources. In this work, we make use of the ensemble learning theory to propose different ensemble learning-based ELM equalizers that need much less spectral resources, while achieving better performance accuracy. Also, we verify the robustness of our proposed equalizers within different communication settings and channel scenarios by conducting different Monte Carlo simulations.**

***Index Terms*—Extreme Learning Machine, Ensemble Learning, OFDM, Equalization.**

I. Introduction

Recently, many Machine Learning (ML) techniques have been proposed and investigated for performing channel estimation and equalization within an Orthogonal Frequency Division Multiplexing (OFDM) communication system. Compared to traditional estimation techniques, ML-based receivers [1], [2] have shown better performance in frequency selective and non-linear channel cases without necessitating the knowledge of channel statistics. However, the majority of these ML techniques conduct an offline training where stochastic gradient descent based iterative algorithms are deployed. These training algorithms require a significant amount of time resources so that they might reach a global optimum. In addition, the offline training concept suffers from a performance degradation when the channel scenario (channel statistics) faced in the deployment phase differs from the ones used during the training process.

Here comes the interest of the extreme learning machine network that was first proposed in [3]. In fact, ELM is a single-layer feed-forward network that generates its hidden layer parameters randomly and uses a one-shot Least Square (LS) technique to calculate the output weights of (*i.e.,* to train) the network. This structure resulted in an extremely fast training and in global optimization capabilities that allowed its online deployment so that it can face the channel scenario at hand. In that regard, different online receivers have been proposed in the literature [4]–[7]. Unlike traditional ML-based communication receivers where the ML network input is split into its real and imaginary parts, ELM could be easily extended to its complex counterpart. This allowed [4] to apply complex received signal directly as input to the ELM; however, it considered a time-domain approach that posed limitations when extended to the OFDM waveform. This has been shown by [5] that proposed a frequency domain ELM system and employed it to jointly equalize and detect the transmitted bits. Using ELM as an equalizer followed by a detection block resulted in a better performance in [6], where the concept was used on a QAM symbol-per-symbol basis. In [7], one ELM network was used to estimate all OFDM subcarriers symbols at once, which yielded better performance compared to the Deep Neural Network (DNN) used in [1], and to the traditional Linear Minimum Mean Squared Error (LMMSE) and LS based channel estimators. However, all these previous studies necessitated a relatively large number of training samples in order to achieve a good performance accuracy.

In our work, we use the ensemble learning theory to propose an Ensemble ELM (EELM) based equalizer that aims at reducing the number of training pilots, thus rendering the online deployment more feasible. Ensemble learning aims to generate different ML models and to combine their decisions in order to obtain an enhanced final accuracy. In contrast to the ELM-based receivers from the literature, we highlight the computational and training efficiencies and the performance superiority of the proposed EELM-based receivers. Different combination and pruning strategies are tested for the design of the EELM equalizer. We also prove the robustness of the proposed EELM for different channel scenarios, different Quadrature Amplitude Modulation (QAM) constellation maps, and for different numbers of subcarriers.

The remainder of this paper is organized as follows. Section II presents the OFDM system model. Section III presents the extreme learning machine based equalizer. Section IV details our proposed Ensemble Extreme Learning Machine

Equalizers. Section V shows the Monte Carlo simulations conducted to evaluate the system performance, while Section VI concludes the work. In this paper, we consider bold lowercase variables $\boldsymbol{x}$ to represent vectors, and bold uppercase variables $\boldsymbol{X}$ to represent matrices. $\boldsymbol{X}^T$, $\boldsymbol{X}^H$ and $\boldsymbol{X}^\dagger$ are the transposition, Hermitian conjugate, and Pseudo-inverse of $\boldsymbol{X}$, respectively. $\otimes$ is the Kronecker product. $diag(.)$ is the Matlab function that transforms a vector into a diagonal matrix and a matrix into a vector of its diagonal elements. $\mathbb{C}$ is the set of complex numbers.

II. OFDM System Model

Considering a block of K OFDM symbols to transmit with M subcarriers each, the transmitted vector $\boldsymbol{x} \in \mathbb{C}^{MK\times 1}$ of the whole transmission block is written as follows:

$$\boldsymbol{x} = [\boldsymbol{x}_1^T\, \boldsymbol{x}_2^T \,...\, \boldsymbol{x}_K^T]^T, \tag{1}$$

where

$$\boldsymbol{x}_k = [x_{1,k}\, x_{2,k}\, \ldots\, x_{M,k}]^T. \tag{2}$$

$x_{m,k}$ is the $\mathcal{M}$-QAM based symbol transmitted at the mth subcarrier and the kth OFDM symbol. We denote by N the number of samples of the whole transmission block, N_u the length of the OFDM symbol and N_{CP} the Cyclic Prefix (CP) length: $N = N_{\text{CP-OFDM}}K$, where $N_{\text{CP-OFDM}} = N_u + N_{\text{CP}}$ is the length of CP-OFDM symbol. The demodulated vector $\boldsymbol{y} \in \mathbb{C}^{MK\times 1}$ of the whole transmitted block can be written as:

$$\boldsymbol{y} = \underbrace{\boldsymbol{F}_{rx}^H \boldsymbol{C} \boldsymbol{F}_{tx}}_{\mathcal{D}} \boldsymbol{x} + \boldsymbol{F}_{rx}^H \boldsymbol{\eta}, \tag{3}$$

where $\boldsymbol{\eta} \in \mathbb{C}^{N\times 1}$ is the additive white Gaussian noise vector, and

$$\boldsymbol{F}_i = diag(\mathbf{1}_K) \otimes \boldsymbol{W}_i. \tag{4}$$

with $i \in \{tx, rx\}$. $\boldsymbol{W}_i \in \mathbb{C}^{N_{\text{CP-OFDM}}\times M}$ represents the OFDM modulation ($i = tx$) or the demodulation ($i = rx$) matrix. $\mathbf{1}_K$ is a vector of K ones.

- **At the transmitter**:

$$\boldsymbol{W}_{tx}(n,m) = \frac{1}{\sqrt{N_u}} e^{j\frac{2\pi}{N_u}m(n-N_{\text{CP}})}$$

- **At the receiver**:

$$\boldsymbol{W}_{rx}(n,m) = 0 \qquad \text{if} \qquad 0 < n \leq N_{\text{CP}}$$
$$\boldsymbol{W}_{rx}(n,m) = \frac{1}{\sqrt{N_u}} e^{j\frac{2\pi}{N_u}mn} \quad \text{if} \quad N_{\text{CP}} < n \leq N_{\text{CP-OFDM}}$$

$\boldsymbol{C} \in \mathbb{C}^{N\times N}$ is the channel convolution matrix of the transmitted signal block, where $\boldsymbol{C}(i,j) = c(i-j)$ with $c(l)$ being the lth tap of the channel impulse response.

$\mathcal{D} = \boldsymbol{F}_{rx}^H \boldsymbol{C} \boldsymbol{F}_{tx}$ represents the system transmission matrix of the CP-OFDM system. When the channel is time unvarying and the CP length is bigger than the maximum delay spread of the channel, $\mathcal{D}$ becomes diagonal with $\mathfrak{c} = diag(\mathcal{D})$ denoting its diagonal vector. Hence, the demodulated symbol $y_{m,k}$ could be equalized by dividing it with the corresponding System Transmission Matrix (STM) value $\mathfrak{c}_{m,k} = \mathcal{D}(mk, mk)$:

$$\hat{x}_{m,k} = \frac{y_{m,k}}{\hat{\mathfrak{c}}_{m,k}}, \tag{5}$$

To obtain the channel estimate $\hat{\mathfrak{c}}_{m,k}$, different techniques could be considered such as LS and LMMSE channel estimators.

III. Extreme Learning Machine Based Equalizer

The extreme learning machine is considered as a general form of single layer feed-forward neural networks where the ELM hidden layer parameters are generated randomly. In other words, hidden layer outputs are always known. Hence, this structure allows the analytical calculation of the output weights during the training phase by means of least square solutions.

Figure 1 depicts the ELM-based equalizer structure. We design the ELM network so that the number of neurons at its input and output layers is equal to the number of subcarriers M. We represent the number of neurons in the hidden layer by $\tilde{N}$. The input to the ELM network is denoted by the matrix $\boldsymbol{Y} \in \mathbb{C}^{N_x\times M}$ which contains N_x concatenated OFDM demodulated symbols. $N_x \in \{N_p, N_d\}$, where N_p represents the number of OFDM pilot symbols used to train the ELM network, while N_d is the number of OFDM data symbols that the ELM will predict, with $K = N_p + N_d$. The corresponding output of the ELM network $\mathcal{F}(\boldsymbol{Y}) \in \mathbb{C}^{N_x\times M}$ is written as follows:

$$\mathcal{F}(\boldsymbol{Y}) = \underbrace{\boldsymbol{G}\Big(\boldsymbol{Y}\boldsymbol{\alpha} + \boldsymbol{B}\Big)}_{\boldsymbol{H}}\boldsymbol{\beta}. \tag{6}$$

$\boldsymbol{\alpha} \in \mathbb{C}^{M\times\tilde{N}}$ is the matrix of hidden layer weights where $\boldsymbol{\alpha}(:,i) = [\alpha_{1,i}, \alpha_{2,i}, \ldots, \alpha_{M,i}]^T$ represents the weights connecting the M input nodes to the ith hidden node. $\boldsymbol{B} = \boldsymbol{b}^T \otimes \mathbf{1}_{N_x}$ with $\boldsymbol{b} = [b_1 b_2 \ldots b_{\tilde{N}}]^T \in \mathbb{C}^{\tilde{N}\times 1}$ representing the bias vector added to the hidden nodes. $\boldsymbol{\beta} \in \mathbb{C}^{\tilde{N}\times M}$ is the matrix of output weights where $\boldsymbol{\beta}(:,i) = [\beta_{1i}, \beta_{2i}, \ldots, \beta_{\tilde{N}i}]^T$ contains the weights connecting the $\tilde{N}$ hidden nodes to the ith output node. $\boldsymbol{G}(.)$ denotes the element-wise activation function. In this work, we consider the inverse hyperbolic sine activation function.

The working principles of the ELM based equalizer can be divided into three main phases: Initialization, training and prediction.

A. Initialization phase

The novelty of the ELM technique lies in the fact that its hidden layer weights $\boldsymbol{\alpha}$ and biases $\boldsymbol{b}$ are not trained, but randomly generated and fixed afterward. As stated by [3], any continuous distribution could be assigned for both α and b. In our work, we consider a uniform distribution on $[-0.1, 0.1]$.

B. Training phase

After generating the hidden layer parameters in the initialization phase, the training phase aims to calculate the output weights parameters, *i.e.*, the matrix $\boldsymbol{\beta}$. To do so, three steps are conducted:

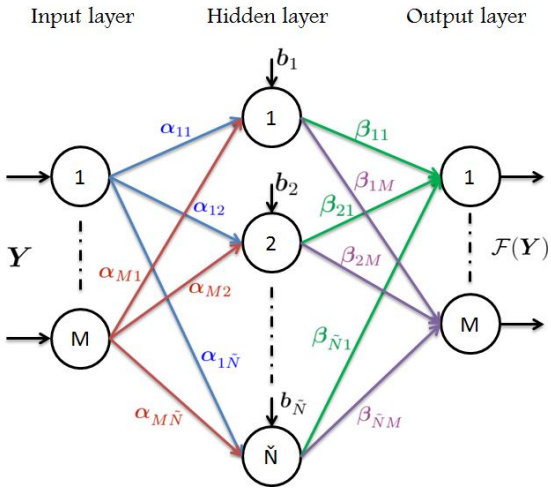

Figure 1. The structure of the ELM based equalizer

1) Generation of the training set: the ELM equalizer is provided with a training set $\left(\boldsymbol{Y}^{(p)}, \boldsymbol{X}^{(p)}\right)$. $\boldsymbol{Y}^{(p)} = [\boldsymbol{y}_1^T\, \boldsymbol{y}_2^T \ldots \boldsymbol{y}_{N_p}^T]^T$ represents the received block of demodulated OFDM symbols at pilots positions, while $\boldsymbol{X}^{(p)} = [\boldsymbol{x}_1^T\, \boldsymbol{x}_2^T \ldots \boldsymbol{x}_{N_p}^T]^T$ is the transmitted block of these OFDM pilots.

2) Calculation of the hidden layer output: $\boldsymbol{Y}^{(p)}$ is fed to the ELM network as its input $\boldsymbol{Y} = \boldsymbol{Y}^{(p)}$ in Eq. (6). Based on that, and on the fact that hidden layer parameters are known, the output of the hidden layer is directly found as:

$$\boldsymbol{H}^{(p)} = \boldsymbol{G}\Big(\boldsymbol{Y}^{(p)}\boldsymbol{\alpha} + \boldsymbol{B}\Big). \tag{7}$$

3) Calculation of the output layer weights $\boldsymbol{\beta}$: $\boldsymbol{X}^{(p)}$ is fed to the ELM based equalizer as its corresponding desired output $\mathcal{F}(\boldsymbol{Y}^{(p)}) = \boldsymbol{X}^{(p)}$. Hence, Eqs. (6) and (7) are used in order to write:

$$\boldsymbol{X}^{(p)} = \boldsymbol{H}^{(p)}\boldsymbol{\beta}. \tag{8}$$

Unlike most of the ML training techniques, Eq. (8) shows that ELM is trained in one shot by minimizing the corresponding LS cost function instead of conducting iterative training, *e.g.,* back propagation. We express the sum of squared errors as follows:

$$\mathcal{E} = \sum_{i=1}^{N_p} \|e_i\|^2, \tag{9}$$

with

$$e_i = \boldsymbol{X}^{(p)}(i,:) - \boldsymbol{H}^{(p)}(i,:)\boldsymbol{\beta}. \tag{10}$$

The least square solution for minimizing $\mathcal{E}$ in Eq. (9) is given by the pseudo-inverse method [8]:

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{H}^{(p)}\right)^{\dagger}\boldsymbol{X}^{(p)}, \tag{11}$$

where

$$\boldsymbol{H}^{\dagger} = \left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{H}^H. \tag{12}$$

C. Prediction phase

Based on the calculated output weights matrix $\hat{\boldsymbol{\beta}}$, the ELM equalizer can estimate the N_d transmitted OFDM data symbols $\boldsymbol{X}^{(d)} \in \mathbb{C}^{N_d \times M}$ in one shot. This prediction is performed by following two steps:

1) Calculation of the hidden layer output: The received OFDM data symbols $\boldsymbol{Y}^{(d)} \in \mathbb{C}^{N_d \times M}$ are fed to the ELM input to calculate the corresponding hidden layer output $\boldsymbol{H}^{(d)}$ using Eq. (7).

2) Equalization: Based on Eq. (6), the hidden layer output, $\boldsymbol{H}^{(d)}$ in this case, is used together with the previously calculated $\hat{\boldsymbol{\beta}}$ to estimate the corresponding transmitted OFDM data symbols:

$$\hat{\boldsymbol{X}}^{(d)} = \boldsymbol{H}^{(d)}\hat{\boldsymbol{\beta}}. \tag{13}$$

IV. Ensemble Extreme Learning Machine Based Equalizers

Ensemble learning aims at generating different ML models that can be combined in a certain way in order to construct a final decision with high prediction accuracy [9]. Projecting this concept into the ELM-based equalizer makes it possible to diversify the ELM structure. This allows countering the problem of the hidden layer output matrix being ill-conditioned and which is due to the random generation of the hidden layer parameters. Hence, we can write the model of the Ensemble Extreme Learning Machine based Equalizer as follows:

$$\mathcal{F}(\boldsymbol{Y}) = \sum_{r=1}^{R} \mathcal{W}_r \mathcal{F}_r(\boldsymbol{Y}). \tag{14}$$

R represents the number of ELM models and $\mathcal{W}_r$ is a weighting function of the rth ELM model $\mathcal{F}_r(\boldsymbol{Y})$ that could be designed in different ways as we show in section IV-C. Eq. (14) implies three main steps: Ensemble generation, ensemble pruning and ensemble integration.

A. Ensemble generation

This steps concerns the adopted method for the creation of the R ELM models. In the ELM framework, this can be achieved by producing different ELM models, each model having the same architecture and training sets while randomly generating its hidden layer parameters [10]. This diversified structure aims at handling the instability that ELM might suffer from due to possibly ill-conditioned matrix $\boldsymbol{H}$ induced by the randomly generated weights and biases of the hidden layer.

B. Ensemble pruning

Ensemble pruning aims at eliminating a subset of the models generated in the previous step based on a specific criterion. This step usually aims at improving the prediction accuracy or reducing the computational complexity. In this work, we compare the mean square error of each ELM model during the training phase, and retain the best Q models ($Q < R$) to participate in the decision making during the prediction phase.

C. Ensemble integration

The ensemble integration step is responsible for combining the predictions of the chosen ELM models in a certain way, thus resulting in the final prediction. One way to do so is to simply estimate the weighted average of the models predictions, where the weights can be given as follows:

$$\mathcal{W}_r = 1/R, \quad 1 \leq r \leq R. \tag{15}$$

Another way to perform this integration is to only take the median of the ELM models predictions. In a more complex method, we design the weight of each ELM model so that it is inversely proportional to its training error:

$$\mathcal{W}_r = \frac{-\log_{10}(MSE_r)}{\sum_{n=1}^{R} \log_{10}(1/MSE_n)}, \tag{16}$$

where $\left(\sum_{n=1}^{R} log_{10}(1/MSE_n)\right)$ is a normalization factor.

V. Numerical Results & Discussion

A. Simulation results

In this section, we conduct Monte Carlo simulations to evaluate the performance of the proposed EELM-based equalizers, while comparing them to the ELM case used in the literature. We also consider the one-tap equalizer with Perfect Channel State Information (PCSI), Least Square estimated CSI (LSCSI) and LMMSE estimated CSI (LMMSE-CSI) as reference cases. Unless stated otherwise, we adopt the following simulations parameters. We consider a Tapped Delay Line model C (TDL-C) [11] power delay profile with an RMS delay spread of 200 ns. A cyclic prefix of $4.7\,\mu s$ is used with an OFDM system constituted of 16 or 64 subcarriers and a 16-QAM constellation map. The subcarrier spacing is set to 15 kHz with a sampling rate of 2.94 MHz. Concerning (E)ELM parameters, we choose $\tilde{N} = 32$ and $N_p = 100$ for the case of $M = 16$ subcarriers, while $\tilde{N} = 192$ and $N_p = 500$ when $M = 64$. R =30 ensemble functions are considered, among which the EELM with pruning retains the best $Q = 10$ ELMs with lowest training errors.

We start, in Figure 2, by presenting the Bit Error Ratio (BER) performance of our proposed EELM based equalizers for $M = 16$ subcarriers. The performance is evaluated for different constellation maps, namely the 4, 64, and 1024 QAM. First, we note that EELM-based equalizers outperform the ELM equalizer for all SNR values and within small and high constellation types. Furthermore, we observe that Average EELM and Weighted EELM have a similar performance and outperform all other equalization techniques, while approaching the PCSI case. Hence, only the average version of EELM equalizer will be shown in the subsequent graphs. It is also worth mentioning that while ELM performs better than LS-CSI for low QAM constellations, it performs badly for higher constellations and especially at high SNR values. In contrast, Average-based EELM proves to perform close to PCSI in high constellation and high SNR values.

In Figure 3, we assess the robustness of the average EELM equalizer against different channel scenarios, where the RMS

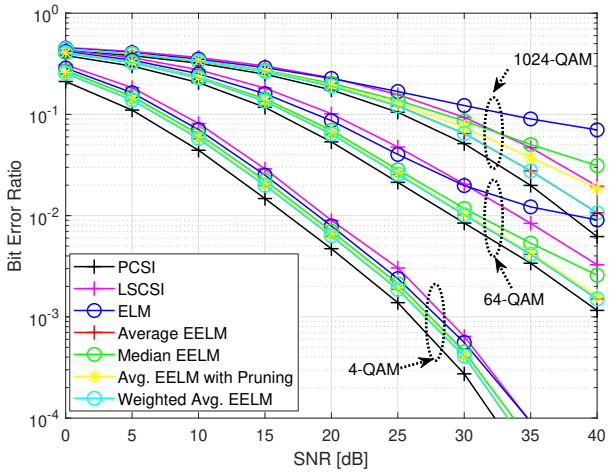

Figure 2. BER obtained for different equalization techniques with (4,64,1024)-QAM constellations, for TDL-C 200 ns

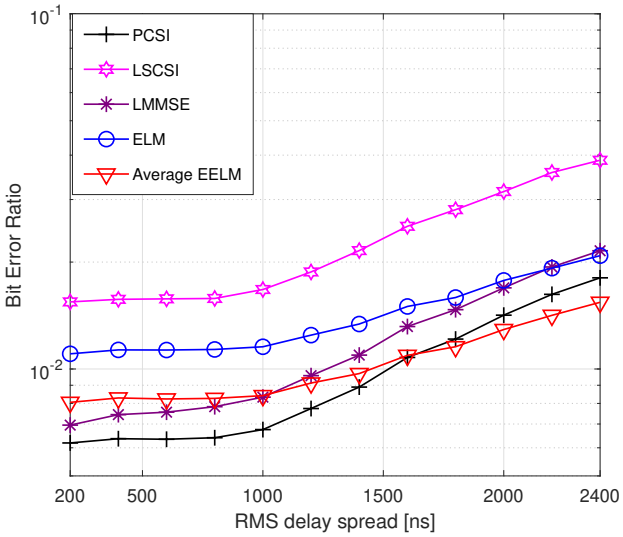

Figure 3. BER in terms of the RMS delay spread of TDL-C model, with 16-QAM and SNR = 25 dB

delay spread of the channel is varied from 200 to 2400 ns. Note that increasing the RMS value will cause the maximum delay spread to start exceeding the adopted CP, leading to inter-symbol interference. It can be clearly noticed how the gap between EELM and the PCSI starts decreasing until the reversal of the two methods performance order. The non-linear capabilities of the EELM equalizer allow to better deal with this imperfectness, thus outperforming the LMMSE estimator case, starting from an RMS at 1000 ns, and the PCSI, starting from an RMS around 1600 ns.

The predominance of the EELM equalizer over its ELM counterpart is highly dependant on the training size N_p. To better analyse this effect, Figure 4 presents the BER obtained with different sizes of the training set. The superiority of

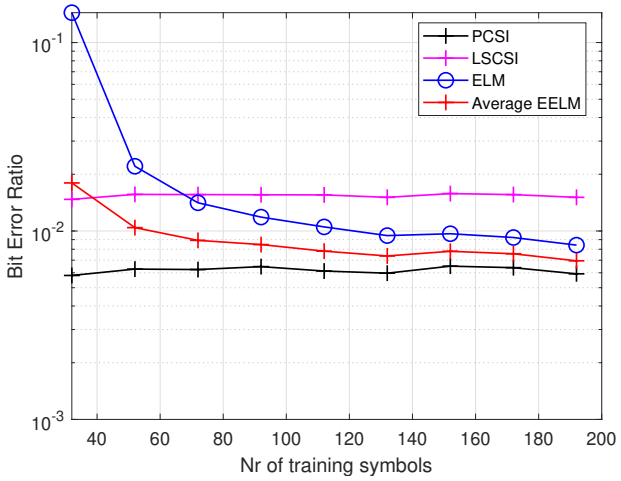

Figure 4. BER for different sizes of the training set with 16 subcarriers, 16-QAM and SNR=25 dB

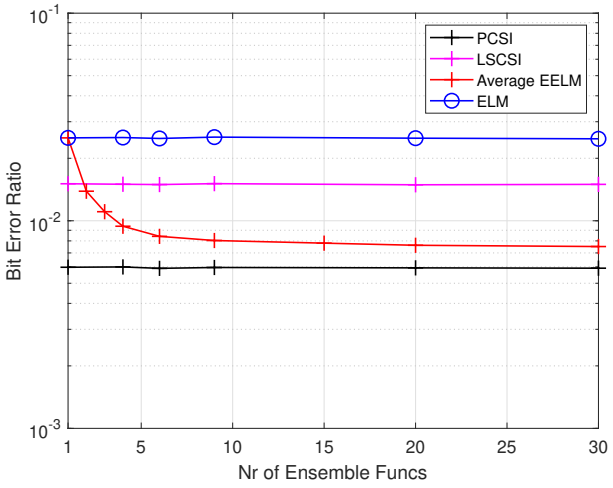

Figure 6. BER in terms of the number of ensemble functions with 64 subcarriers, 16-QAM and SNR=25 dB

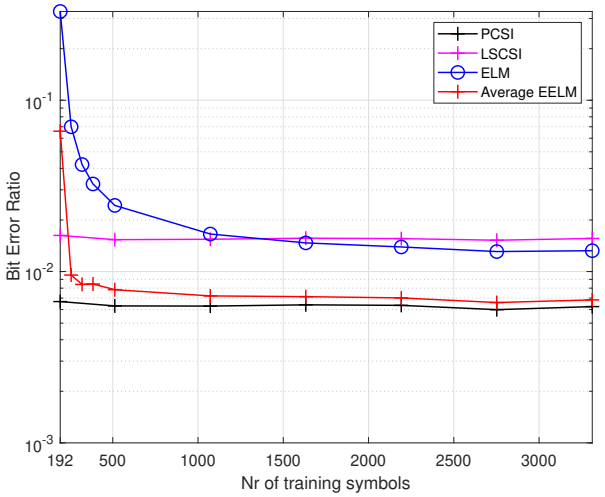

Figure 5. BER for different sizes of the training set with 64 subcarriers, 16-QAM and SNR= 25 dB

the proposed EELM-based equalizer is clearly cast by its capability of necessitating only a small number of training symbols to achieve high performance accuracy.

In order to handle higher numbers of OFDM subcarriers, the dimensions of the ELM-based system need to be increased significantly, which is reflected on the need for a higher amount of training pilot symbols. In Figure 5, we consider the $M = 64$ subcarriers scenario, where we set $\tilde{N} = 192$ hidden nodes and evaluate the BER for different training sizes. While EELM performs close to the PCSI case starting from almost 300 training samples, ELM needs about 1500 training samples to perform similarly to the LS-CSI case while being constantly outperformed by EELM.

In Figure 6, we show that EELM does not need a large amount of ensemble functions R to achieve this superiority. In fact, this is evident starting from a few ensemble functions $R \approx 5$ and it saturates at about $R \approx 10$.

B. Discussion on spectral and computational efficiencies

In this section, we discuss the aspects of spectral efficiency and computational complexity of our proposed EELM equalizer. In the training phase, the computational complexity comes mainly from equations (11) and (12) that we rewrite in one equation for clarity:

$$\hat{\boldsymbol{\beta}} = \left(\left(\boldsymbol{H}^{(p)} \right)^H \boldsymbol{H}^{(p)} \right)^{-1} \left(\boldsymbol{H}^{(p)} \right)^H \boldsymbol{X}^{(p)}. \qquad (17)$$

The computational complexity of the matrix inversion process is of order $O(\tilde{N}^3)$. However, knowing that $N_p > \tilde{N} > M$, we can note that the complexity of the multiplication $\boldsymbol{H}^H \boldsymbol{H}$ will dominate that of other operations. Thus, the ELM complexity will be given asymptotically by $O^{ELM}(\tilde{N}^2 N_p)$.

The computational complexity of the Average EELM based equalizer without pruning is in order of $O^{EELM}(R\tilde{N}^2 N_p)$. When using the same number of hidden nodes for both ELM and EELM, the difference in computational load stems from:

1) The number of ensemble functions R.
2) The number of needed training pilots N_p.

From Figure 6, we can note that EELM performance converges at about $R = 10$. On the other hand, Figures 4 and 5 clearly show that the number of training symbols N_p, that EELM needs to achieve a certain performance accuracy, is much smaller than that required by the ELM technique. Also, the gap increases when dealing with high numbers of subcarriers, *e.g.*, 64, 128, and so on.

1) 16 subcarriers case: For example, in the case of 16 subcarriers, we see, from Figure 4, that ELM needs about 200 training symbols to achieve the same performance as that obtained by EELM with 100 training symbols. Hence, EELM computational complexity can be written in terms of the ELM complexity as $O^{EELM} = \frac{R}{200/100} O^{ELM}$. With $R = 10$,

this means that EELM will have 5 times the computational complexity of ELM, while saving 50% of ELM used pilots.

2) Case of higher numbers of subcarriers: When the number of subcarriers starts increasing, EELM rises as a highly efficient solution compared to ELM in terms of both computational complexity and spectrum resources (*i.e.* number of training pilots). To support these claims, we consider the example of 64 subcarriers. As it can be seen from Figure 5, ELM requires more than 3000 training symbols to achieve the same performance accuracy of EELM with only 200 training symbols. Hence, the EELM computational complexity can be approximated as follows: $O^{EELM} < \frac{R}{3000/200} O^{ELM}$. With $R = 10$, we can say that EELM consumes less than 66% of ELM computational resources. In addition, EELM uses less than 6% of ELM spectral resources.

In fact, considering higher numbers of subcarriers, *e.g.,* 256, 512, and so on, will increase the gap between EELM and ELM even further, in terms of both spectral and computational efficiencies, thus leading to the inevitable adoption of EELM.

It is worth mentioning that the complexity of Average EELM is R times higher than ELM in the prediction phase and it depends on the number of predicted data symbols N_d and on the number of subcarriers. Overall, the dominant complexity component (in the prediction phase or training phase) depends on the number of data symbols (Short packet / Long packet) and on the number of subcarriers.

To sum up, we can say that:

- EELM has a higher spectral efficiency than that of ELM, the gap increasing with an increasing number of subcarriers.
- In the prediction phase: EELM has a higher computational complexity than ELM.
- In the training phase:
  - With a small number of subcarriers(~ 16): EELM has a higher computational complexity than ELM.
  - With a high number of subcarriers(≥ 64): EELM has a lower computational complexity than ELM. The gap increases with an increasing number of subcarriers.
- Determining whether ELM or EELM technique will have a higher overall computational complexity depends on the number of subcarriers and the number of data symbols.

VI. Conclusion

In this paper, we have proposed different extreme learning machine based equalizers for OFDM systems. Compared to traditional machine learning based equalizers, ELM has proven its fast training and high performance accuracy without requiring any knowledge of the channel statistics. While this allows ELM to be deployed in an online mode, it causes high losses in spectral resources in terms of training samples. To alleviate the ELM needs for large training sets, we have proposed to use the ensemble learning theory where a set of ELM networks is designed instead of one ELM. We have considered different Ensemble ELM techniques and showed that averaging the decisions of few ELMs is enough to achieve a performance accuracy much better than that of ELM alone, while exceeding the performance of LMMSE channel estimation in frequency selective channels. In addition, we have shown that training the EELM costs much less computational resources than ELM when using a relatively large number of subcarriers. To prove the robustness of our proposed technique, we have considered different constellation maps, channel scenarios, and numbers of OFDM subcarriers, and highlighted the EELM superiority by means of Monte Carlo simulations. In future works, we believe that EELM performance should be evaluated in time-varying channels, by the means of an adaptive online EELM-based receiver that could be deployed efficiently in doubly selective channel scenarios.

VII. Acknowledgements

This work was performed in the framework of the EL-SAT2020 project which is co-financed by the European Union with the European Regional Development Fund, the French state and the Hauts de France Region Council.

References

[1] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[2] J. Zhang, C.-K. Wen, S. Jin, and G. Y. Li, "Artificial intelligence-aided receiver for a cp-free ofdm system: Design, simulation, and experimental test," *IEEE Access*, vol. 7, pp. 58 901–58 914, 2019.

[3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[4] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, 2005.

[5] I. G. Muhammad, K. E. Tepe, and E. Abdel-Raheem, "QAM equalization and symbol detection in OFDM systems using extreme learning machine," *Neural Computing and Applications*, vol. 22, no. 3-4, pp. 491–500, 2013.

[6] L. Yang, Q. Zhao, and Y. Jing, "Channel equalization and detection with ELM-Based regressors for OFDM systems," *IEEE Communications Letters*, vol. 24, no. 1, pp. 86–89, 2020.

[7] J. Liu, K. Mei, X. Zhang, D. Ma, and J. Wei, "Online extreme learning machine-based channel estimation and equalization for OFDM systems," *IEEE Communications Letters*, vol. 23, no. 7, pp. 1276–1279, 2019.

[8] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2005.

[9] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression: A survey," *ACM computing surveys (CSUR)*, vol. 45, no. 1, pp. 1–40, 2012.

[10] S. G. Soares and R. Araújo, "An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction," *Neurocomputing*, vol. 171, pp. 693–707, 2016.

[11] 3GPP, ETSI TR 138 900 V14.2.0 (2017-06), "Study on channel model for frequency spectrum above 6 ghz."