PAM-4 Behavioral Modeling with Laguerre-Volterra Feed Forward Neural Network and Eye-Diagram Generation in ADS

1st Xinying Wang
dept. Electrical&Computer Engineering
University of Illinois Urbana-Champaign
Urbana, IL, USA
xinying@illinois.edu

2nd Thong Nguyen
dept. Electrical&Computer Engineering
University of Illinois Urbana-Champaign
Urbana, IL, USA
tnnguye3@illinois.edu

3rd Jose E. Schutt-Aine
dept. Electrical&Computer Engineering
University of Illinois Urbana-Champaign
Urbana, IL, USA
jesa@illinois.edu

***Abstract*—In this paper, we demonstrate a PAM-4 IBIS-AMI model obtained from machine learning for time domain simulation in ADS. More specifically, we report a Laguerre-Volterra-expanded Feed-Forward neural Network (LVFFN) with only one hidden layer and 10 neurons to model the 28Gb/s PAM-4 high speed link buffer. the LVFFN model reduces the model size and improves the computational efficiency dramatically compared to the Volterra series model and other transitional artificial neural network models. The LVFFN PAM-4 model is implemented in IBIS-AMI, a industrial standard for high speed link modeling. Eye-diagram analysis is conducted on LVFFN PAM-4 IBIS-AMI model in Keysight ADS. Our results show that our model is faster to simulate 1 million bits and is easier to implement compared to the standard PAM-4 IBIS-AMI model.**

***Index Terms*—PAM-4, IBIS-AMI, Laguerre, Behavior Modeling, Eye Diagram**

I. Introduction

The growth of new technologies such as big data, cloud computing, the Internet-of-Things, 5G, and artificial intelligence, has driven demand for instant data exchange around the world. High speed link(HSL) technology's limitations in data transmission is one of the bottle-neck threatens the entire infrastructure.

HSL technology can be divided into two categories: Non-Return-to-Zero(NRZ), also known as Pulse Modulation 2-level(PAM-2) and Pulse Modulation 4-level(PAM-4). Over the past five decades, NRZ technology has been evolving from 10Gb/s to current 100Gb/s and it will continue to scale up to next generation 400Gb/s HSL technology. PAM-4 is an emerging technology targeting 400Gb/s (8lanes, 56Gb/s/lane). There are challenges when linearly scaling up from 100Gb/s to 400Gb/s. At 400Gb/s, these include totally closed eye, shorter unit interval(UI), and tighter jitter requirements. The eye closing issue may be addressed by applying more aggressive equalization technologies such as improved Decision-Feedback Equalizer(DFE) and Continuous Time Linear Equalizer (CTLE). The shorter UI, only 17ps at 400Gb/s, is more challenging because of the resulting tighter jitter budget which may be below the intrinsic jitter budget of the testing or support equipments.

PAM-4, on the other hand, will alleviate some of challenges of NRZ technology such as the tighter jitter budget and worse channel loss etc. PAM-4 technology has four digital amplitude levels(-3, -1, 1 and 3). Each symbol carries 2-bit information. This means that at the same throughput as NRZ, PAM-4 only requires half of the bandwidth. This addresses the shorter jitter budget problem that NRZ technology faces. However it alters the design, testing, and verification of the system due to multi-level signaling. Unlike NRZ, for which there is decades of experience on which to back new developments, there is limited information on PAM-4 to which designers and testers can refer. Therefore, instead of designing and building PAM-4 testing and measurement equipment, simulating and validating PAM-4 system is crucial.

Growing interests on applying machine learning approaches to model HSL buffers and other electronic systems are indicated by the increasing number of new publications on this topic. Neural networks are not a new topic. They have long been used to model electronic systems. However, due to computational resources limit and the lack of an efficient optimization/training algorithm, the method has been growing slowly. With recent advancements in back-propagation [1] training algorithms, deep neuron network has progressed in a variety of fields. Recent publications on modeling HSL system and other electronic circuit/system is an evidence that deep neural network is thriving in the field of electronic system and IC design [2]–[6].

In this study, we model PAM-4 HSL buffer with LVFFN and implement the model in IBIS-AMI. We aim to develop a compact behavioral model for HSL buffer to conduct rapid and accurate signal integrity verification. This model will be compatible with main stream EDA softwares to enable simulation and analysis under the universal industrial production environment. The structure of this paper is structured as follows: Section II presents the mathematical formulation

for LVFFN. Section III gives IBIS-AMI basics. Section IV shows a numerical example. Section V concludes the work and proposes the next steps.

II. LVFFN

For discretized input-output data, the Volterra series of causal, stable, nonlinear, time-invariant system is given by

$$\begin{aligned} y(n) =& h_0 + \sum_{m=0}^{M} h_1(m)u(n-m) \\ &+ \sum_{m_1=0}^{M}\sum_{m_2=0}^{M} h_2(m_1,m_2)u(n-m_1)u(n-m_2) \\ &+ \sum_{m_1=0}^{M}\sum_{m_2=0}^{M}\sum_{m_3=0}^{M} h_3(m_1,m_2,m_3)u(n-m_1) \\ &\times u(n-m_2)u(n-m_3) \\ &+ ... \end{aligned} \tag{1}$$

where n denotes the time step, $u(n)$ the input data sequence, $y(n)$ the output data sequence, M the system memory length and h_i are the i^{th}- order Volterra kernels (VKs).The number of VK needs to be estimated is a function of memory length and system order. The curse of dimensionality for Volterra series has greatly limited its applications. At the certain point, the number of kernels exceeds the data length significantly, which will become an unbearable burden due to the vast number of parameters. Such curse of dimensionality will lead to computational cumbersome and over-fitting.

To reduce the curse of dimensionality associated with Volterra series, one of the solution is projecting VKs on an set of orthonormal basis functions [7]. Laguerre functions are among the most popular ones. Laguerre functions are the solutions of Laguerre differential function. The general form of discrete Laguerre functions are shown below [8]:

$$\phi_r(\tau) = \alpha^{\frac{\tau-r}{2}}(1-\alpha)^{\frac{1}{2}}\sum_{k=0}^{r}(-1)^k\binom{\tau}{k}\binom{r}{k}\alpha^{r-k}(1-\alpha)^k \tag{2}$$

where α denotes the decay factor; r denotes order of Laguerre function; τ denotes the discrete time step. All Laguerre functions are inter-orthonormal, which given by

$$\forall i,j : \langle \phi_i, \phi_j \rangle = \begin{cases} \delta_{ij}, & i = j \\ 0, & i \neq j \end{cases} \tag{3}$$

The Laguerre expanded Volterra series can be expressed by:

$$\begin{aligned} y(n) =& \theta_0 + \sum_{r=1}^{R}\theta_r\phi_r + \sum_{r_1=1}^{r_1=R}\sum_{r_2=1}^{r_2=R}\theta_{r_1,r_2}\phi_{r_1}\phi_{r_2} \\ &+ ... + \sum_{r_1=1}^{r_1=R}...\sum_{r_n=1}^{r_n=R}\theta_{r_1,...,r_n}\prod_{i=1}^{n}\phi_{r_i} \end{aligned} \tag{4}$$

Equation 4 has the same form as equation 1. Parameter identification shifts from VKs in equation 1 to Laguerre parameters θ_r in equation 4. The number of parameters need to be identified have been reduced dramatically. Typically number of Laguerre functions R employed for expansion is much smaller than the memory length M.

Parameter identification can be achieved with Monomial Power Series Neural Network (MPSNN) reported in our previous work [4], [6]. The Neural network architecture is shown in figure 1

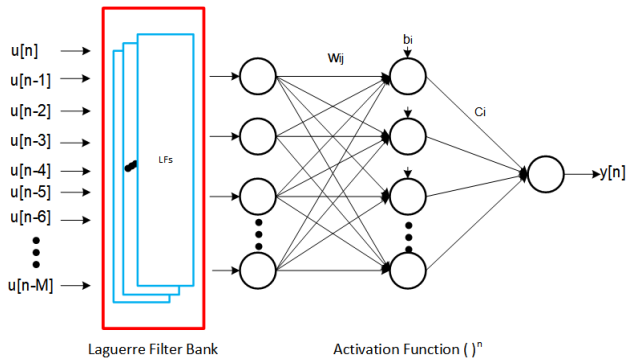

Fig. 1. LVFFN structure with one hidden layer

As seen in figure 1, the time series signal is first convolved with Laguerre functions, also known as Laguerre filters. The convolved outputs are then fed to the input layer of feed forward neural network (FFN). The number of neurons is equal to the number of Laguerre functions used in the Laguerre filter bank. The activation function is a monomial power series term which is cubic in this work. Once the network is trained, the Laguerre parameters θ_r can be identified by:

$$\theta_r(v_1,v_2,...,v_r) = \sum_{i=1}^{M} c_i a_{ri} w_{v_1 i} w_{v_2 i} ... w_{v_r i} \tag{5}$$

where c_i and w_{ij} denote the weights in hidden layer and output layer; M is the memory length; a_{ri} can be calculated by

$$a_{ri} = b_i^{k_b}\begin{pmatrix} n \\ k_0, k_1, k_2, ..., k_M \end{pmatrix}_{k_b+k_0+....+k_M=r} \tag{6}$$

III. IBIS-AMI BASICS

IBIS-AMI standard specifies the interface between SerDes behavior model and simulator. The model created under this specification is ensured to work with the simulator that fulfills the same standard. A typical IBIS-AMI model contains analog and algorithmic portions. The analog part includes the IBIS specifications and channel S-parameter files. Th algorithmic part contains executable files and .ami files.The executable files are Dynamic Link Library (DLL) files typically written in C++. The .ami files are ASCII text files specifying all the parameters that is used by the executables and the simulator.

IBIS-AMI specification defines the application program interface (API) functions which provides a platform for user to implement customized algorithmic model. The model is

specifically used for HSL simulation and analysis. Such API includes three main functions: *AMI_Init()*, *AMI_GetWave()*, and *AMI_Close()*. *AMI_Init()* is mainly doing initialization which includes allocating/initializing dynamic memory, parsing parameters from .ami file, and conducting impose response filtering. *AMI_GetWave()*, on the other hand, conducts bit-to-bit simulation on the input waveform for nonlinear and time-variant system. *AMI_GetWave()* is optional function which can be explicitly disabled in the .ami file. If this function is disabled, the simulator will use the impose response returned by *AMI_Init()* for simulation. Otherwise, the simulator will ignore the impose response returned by *AMI_Init()*. It will then take the waveform returned by *AMI_GetWave()* for further simulation. The last function, *AMI_Close()*, typically conducts post-simulation clean-up, which includes shutting down simulation engine, releasing memory allocated, informing simulator that the simulation has been completed successfully or with errors.

The whole IBIS-AMI simulation process starts from generation of PRBS excitation bit sequence. Then the simulator pre-processes .ami file and computes the impose response from the analog S-parameter files. In the following, it presents all the information to the Tx AMI model. After that, the simulator take the impose-response output from the *AMI_Init()* and convolve with initial excitation bit sequence, if the *AMI_GetWave()* is disabled. Otherwise, it takes the waveform from *AMI_GetWave()* and convolve with analog impose response. Next, the resulted waveform is presented at the Rx AMI model input. The process is repeated in Rx DLL. The resulted output is either impose response from Rx *AMI_Init()* or the waveform from Rx *AMI_GetWave()*. The simulator will take the output from Rx DLL to conduct the final eye diagram analysis.

IV. Numerical Example

a PAM-4 high speed link system is modeled with LVFFN. Figure 2 shows the input and output signals that are used for training the models. The input/output signals are generated from a Vendor provided PAM-4 IBIS-AMI model. The input signal is a 4-level PRBS with magnitude between -1.0 V and 1.0 V . The data rate is 28Gb/s. Total 16,651 samples are used for training and validation (70%/30%). The regular FFN model has three layers: input layer,hidden layer, and output layer. A third order monomial power series activation function is employed in the neural network. We use PyTorch framework for neural network training. The memory length plays an important role in getting a good network prediction accuracy. In our previous work, we have demonstrated that longer memory length resulted in a better accuracy [6]. There is elbow point for the memory length after which increasing length will not benefit accuracy obviously. We found that for PAM-4 system, memory length of 150 produces optimal result. In this work, we will use this memory length for both FFN model and LVFFN model.

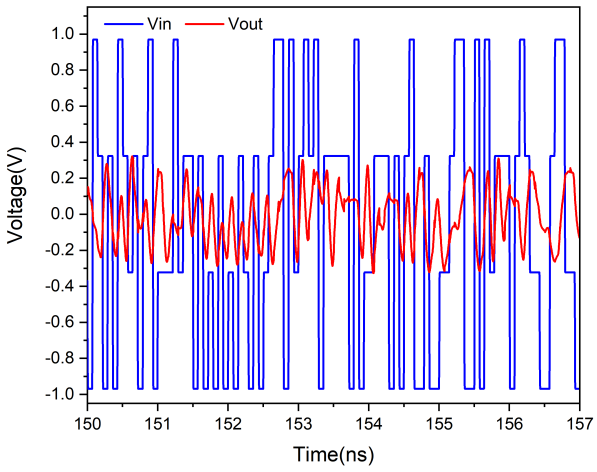

Fig. 2. PAM-4 training signal for LVFFN

Finding the correct decay factor α and number of Laguerre functions R are crucial for model success. We employed brute force method to obtain the optimal α and the truncated number of Laguerre function. The optimal value of α and number of Laguerre functions for the PAM-4 system are determined to be 0.93 and 10, respectively.

Output of LVFFN is plotted with reference signal and signal obtained from Recurrent Neural Network (RNN) model in figure 3. The model training/prediction accuracy are %98/%97 with α = 0.93 and using the first 10 Laguerre functions. Training converges in less than 100 epochs and takes less than 5 minutes on a regular Desktop computer with quad-core i7 processor.

To achieve similar prediction accuracy, FFN requires one hidden layer and 150 neurons in the layer, while RNN requires 6 stacked layer with 20 neurons in each layer and 100 memory length [5]. The model size reduction is presented in figure 4. As seen, LVFFN model with 111 parameters can model whole PAM-4 system, while the Volterra series, RNN, and FFN requires more than 1.6 million, 454 thousand, and 22 thousand parameters, respectively. LVFFN model demonstrate great efficiency in model size reduction.

We implement LVFFN PAM-4 model in IBIS-AMI and we conduct eye-diagram analysis in Keysight ADS. The result is demonstrated in figure 5. The LVFFN model is implemented in receiver dynamic link library (DLL). The transmitter DLL is responsible for excitation generation. Simulating one million bits takes only 142s, less than 3 minutes. The standard PAM-4 IBIS-AMI model requires around 5 minutes for simulating the same amount of bits [9]. The eye-diagram is plotted in ADS as

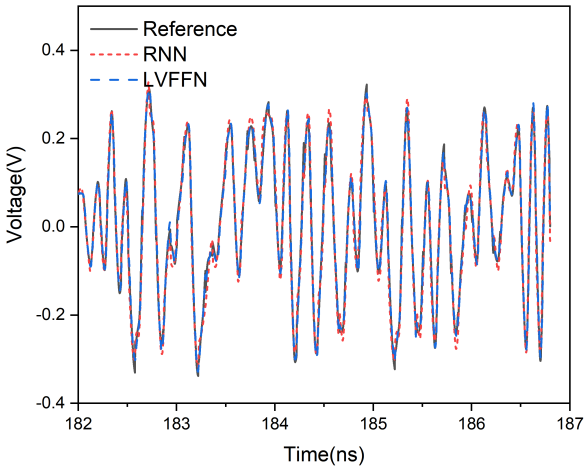

Fig. 3. Model output comparison of LVFFN, RNN and the reference signal

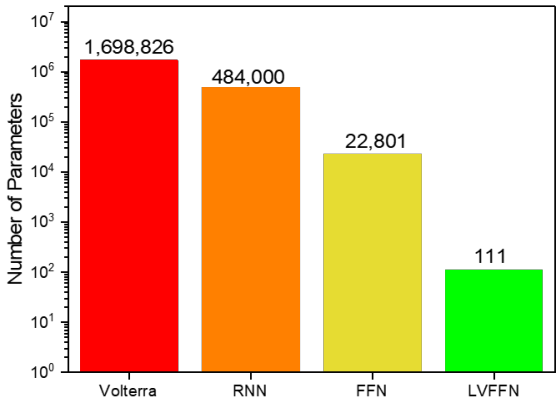

Fig. 4. Comparison of model size for Volterra series, RNN, FFN, and LVFFN

well. The eye shapes for LVFFN model and reference signal are similar.

V. Conclusion and Future Work

In this work, LVFFN is proposed to model PAM-4 HSL system. The model size is reduced significantly compared to Volterra series, RNN, and FFN models, while the same accuracy is maintained. The model is implemented in IBIS-AMI, an industrial standard for HSL modeling. Eye-diagram analysis is conducted on LVFFN PAM-4 IBIS-AMI model in Keysight ADS. The model improves simulation speed compared to the standard PAM-4 IBIS-AMI model, while the implementation of LVFFN model in IBIS-AMI is much simpler than standard one.

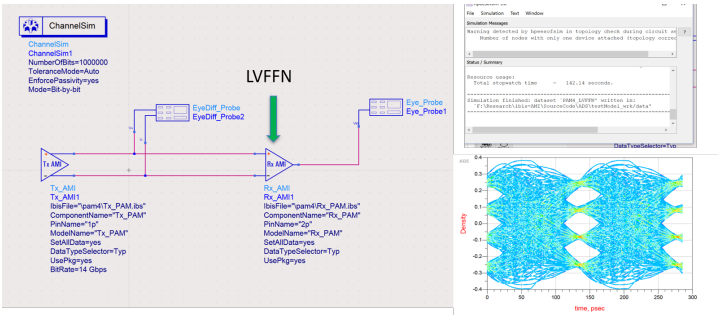

Fig. 5. Snapshot of eye-digram analysis with LVFFN PAM-4 IBIS-AMI model in ADS

Our LVFFN PAM-4 model includes transmitter, channel, and receiver. If there is change in channel, the model has to be regenerated. Therefore, this model is channel-dependent. For future work, we propose that instead of modeling the whole PAM-4 system including channel, we model the receiver using our LVFFN approach. This way, we can make our LVFFN model channel independent and add more flexibility.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CNS 16-24810 for Center for Advanced Electronics through Machine Learning (CAEML), the U.S Army Small Business Innovation Research (SBIR) Program office and the U.S. Army Research Office under Contract No.W911NF-16-C-0125 and by Zhejiang University under grant ZJU Research 083650.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015, doi:10.1038/nature14539.

[2] Z. Chen, M. Raginsky, and E. Rosenbaum, "Verilog-a compatible recurrent neural network model for transient circuit simulation," pp. 1–3, 2017.

[3] B. Li and P. Franzon, "Machine learning in physical design," May 2017, pp. 147–150.

[4] X. Wang, T. Nguyen, and J. Schutt-Aine, "Volterra kernel extraction through monomial power series feed forward neural network for behavior modeling of high speed i/o buffer," in *EMC Sapporo & APEMC 2019*, 2019, pp. 1–4.

[5] T. Nguyen, T. Lu, K. Wu, and J. Schutt-Aine, "Fast Transient Simulation of High-Speed Channels Using Recurrent Neural Network," 2019. [Online]. Available: https://arxiv.org/pdf/1902.02627.pdf

[6] T. Nguyen, X. Wang, X. Chen, and J. Schutt-Aine, "A deep learning approach for volterra kernel extraction for time domain simulation of weakly nonlinear circuits," in *2019 IEEE 69th Electronic Components and Technology Conference (ECTC)*, 2019, pp. 1889–1896.

[7] G. Mitsis, M. Poulin, P. Robbins, and V. Marmarelis, "Nonlinear modeling of the dynamic effects of arterial pressure and co2 variations on cerebral blood flow in healthy humans," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 51, no. 11, pp. 1932–1943, 2004.

[8] H. Ogura, "Estimation of wiener kernels of a nonlinear system and a fast algorithm using digital laguerre filters," in *15th NIBB Conference*, 1985, pp. 14–62.

[9] "PAM-4 Design Challenges and the Implications on Test ." [Online]. Available: http://literature.cdn.keysight.com/litweb/pdf/5992-0527EN.pdf