Published in the 2020 IEEE Electrical Design of Advanced Packaging and Systems (EDAPS) symposium. Not available yet in the IEEE Xplore as of 12/17/20.

Invertible Neural Networks for Inverse Design of CTLE in High-speed Channels

Majid Ahadi Dolatsara*, Huan Yu*, Jose Ale Hejase†, Wiren Dale Becker‡, Madhavan Swaminathan*
*School of Electrical and Computer Engineering, 3D Systems Packaging Research Center (PRC)
Georgia Institute of Technology, Atlanta, GA, USA
†NVIDIA, Austin, TX, USA[1]
‡IBM, Poughkeepsie, NY, USA

***Abstract*—Designing CTLE of high-speed channels can be complicated and time consuming. To alleviate this issue, this paper investigates the invertible neural networks (INNs) for inverse design of the CTLE. In this approach, a desired eye height and eye width is given, and the algorithm finds the corresponding peaking frequency and gain value of the CTLE. INN is a special type of neural networks that can be traversed in both forward and reverse directions. An advantage of this network is producing distribution of the input variables based on the desired output. This feature enables the algorithm to provide multiple solutions when a multi-modal distribution is produced. Thus, the user can choose the appropriate solution based on other constraints. A numerical example for inverse design of CTLE of a SerDes channel is provided, which results in moderate accuracy. However, other variations of the example show that the accuracy is case dependent which implies improvements on the algorithm is needed.**

Keywords— CTLE, eye diagram, high-speed channel, inverse design, invertible neural network, machine learning, SerDes

I. Introduction

With the exponential increase in the data rate of high-speed serial channels, their design has become more challenging. Designers need to set many design parameters and consider several constraints to satisfy the performance criteria, including a low bit error rate (BER). One of the critical tasks in this process is design of the equalization, which is often done with feed forward equalizer (FFE), decision feedback equalizer (DFE), and continuous linear time equalizer (CTLE). FFE and DFE coefficients are often calculated adaptively. Although the frequency response of CTLE can be found theoretically, in reality it does not always provide a satisfactory result. Therefore, engineers depend on simulating the channel with all practical CTLE frequency responses. Unfortunately, this method can be very time consuming. It is possible to utilize human tuning or optimization methods; however, the best solution may still not be achieved, or several satisfactory possibilities could be ignored due to the nature of these algorithms. To address these issues, we propose an inverse design approach to find the CTLE settings.

In the traditional design and modeling process, from a combination of design parameters (inputs) the output of the system is found. In contrast, in the inverse approach we start from the output objectives and derive the corresponding input parameters that satisfy these objectives. The inverse problem has been a popular concept for decades. However, recently with the advancements in machine learning (ML), there has been several attempts for inverse design of high-speed electronics by using ML [1]-[6]. From these techniques, INNs [7] have shown a great potential. A main advantage of these networks is providing distribution of the design parameters instead of deterministic values. This advantage can be used to deal with the non-uniqueness of the solution issue, which can be a major problem in inverse design with traditional approaches. Using INNs, we can derive several possible combinations of the design parameters instead of one. Then a satisfactory design can be selected based on other constraints. We have previously used INNs for design of SIW filters in D-band [6]. In this work, we investigate to see if a similar approach can be used to derive the CTLE settings from the desired eye height and eye width. This is a challenging problem because the considered CTLE settings are discrete. In addition, the outputs have a nonlinear relationship with the design parameters.

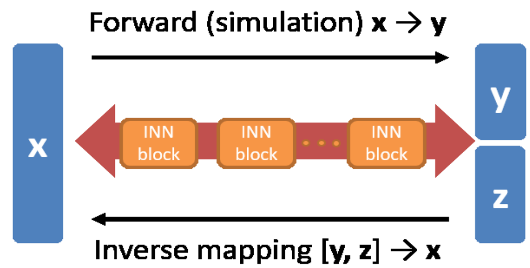

Fig. 1. Structure of the invertible neural networks.

II. Proposed Approach

A. Invertible neural networks

In the proposed inverse design approach INNs [7] are used. The general INN network is illustrated in Fig. 1, which shows it is comprised of several reversible blocks. This reversible structure permits bidirectional training of the network. In this figure, **X** shows the input parameters, and **Y** is the output. **Z** is a set of latent variables with normal distribution that do not exist in the actual system. These variables are added in the output to store the lost information in the forward mapping from **X** to **Y**. In the training process, a supervised loss function, such as the mean square error, is used for **Y** since it represents deterministic variables. On the other hand, because **X** and **Z** are stochastic variables and represent distributions, the maximum mean discrepancy (MMD) is used as their loss function. MMD is an unsupervised loss function, and it only needs samples from two distributions to compare them.

1- At the time of his contributions, Jose Ale Hejase was with IBM, Austin, TX, USA.

978-1-7281-9412-7/20/$31.00 ©2020 IEEE

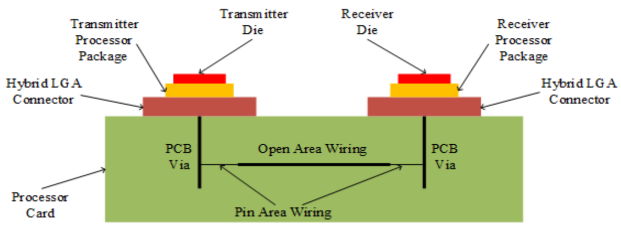

Fig. 2. High-speed SerDes channel in the numerical example.

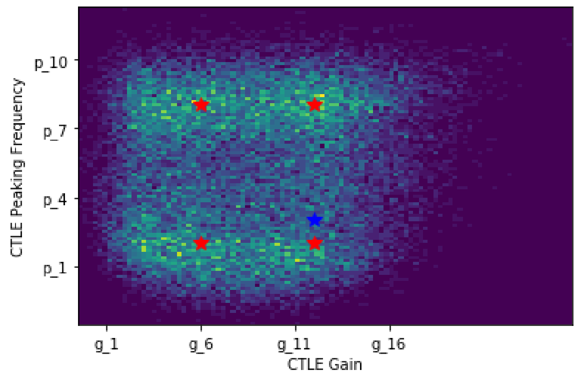

Fig. 3. Joint distribution of CTLE peaking frequency (p_i) and CTLE gain (g_j) in the numerical example. Candidate designs are shown with red stars and the accurate design is shown with a blue star.

Each INN block needs an even and equal number of inputs and outputs, which is enforced with zero padding if needed. Inputs and outputs are divided into two halves, which are shown as $[\boldsymbol{x_1}, \boldsymbol{x_2}]$ and $[\boldsymbol{y_1}, \boldsymbol{y_2}]$, respectively. The forward path of each block is equivalent to:

$$\begin{aligned} \boldsymbol{y_1} &= \boldsymbol{x_1} \,.\, exp\big(S_2(\boldsymbol{x_2})\big) + t_2(\boldsymbol{x_2}), \\ \boldsymbol{y_2} &= \boldsymbol{x_2} \,.\, exp\big(S_1(\boldsymbol{y_1})\big) + t_1(\boldsymbol{y_1}), \end{aligned} \quad (1)$$

And the reverse path of each block is equivalent to:

$$\begin{aligned} \boldsymbol{x_2} &= \big(\boldsymbol{y_2} - t_1(\boldsymbol{y_1})\big) \,.\, exp\big(-S_1(\boldsymbol{y_1})\big), \\ \boldsymbol{x_1} &= \big(\boldsymbol{y_1} - t_2(\boldsymbol{x_2})\big) \,.\, exp\big(-S_2(\boldsymbol{x_2})\big), \end{aligned} \quad (2)$$

where, S_1, S_2, t_1, and t_2 are neural networks themselves. Note that although these subnetworks are not invertible, (1) and (2) are always invertible. The individual blocks are connected with shuffling layers. For additional details refer to [7].

B. Application to CTLE

CTLE is a high path filter that is intended to reverse the low path filtering effect of the channel. A first order high path filter can be characterized by a peaking frequency and gain value. Goal of the proposed approach is to derive these parameters from desired eye height (EH) and eye width (EW) values, using the INN.

Although frequency and gain are continuous variables, in reality a limited number of CTLE hardwares are available to the designer. Therefore, the input design parameters are discrete. We show possible peaking frequencies and gain values with $[p_1, p_2, \ldots p_N]$ and $[g_1, g_2, \ldots g_M]$, respectively, where, N and M are the number of possible implementations. Thus, there are $N * M$ selections for the CTLE design. The inverse problem is equivalent to:

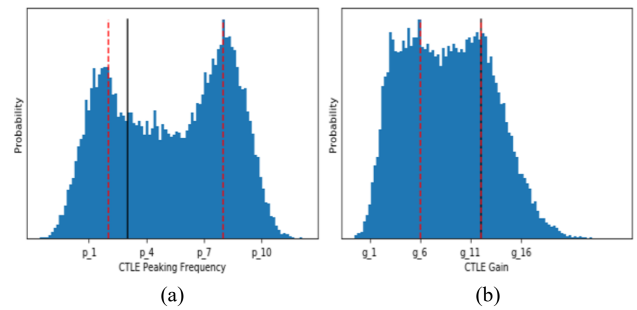

Fig. 4. Marginal distributions of the CTLE parameters in the numerical example. Candidate designs are shown with red dashed lines and the accurate design is shown with a black line. a) CTLE peaking frequency (p_i). b) CTLE gain (g_j).

$$[p_i, g_j] = f^{-1}(EH, EW, \boldsymbol{Z}), \quad (3)$$

where, $\boldsymbol{Z}$ is the latent variables, and f^{-1} is the inverse mapping, which is found by training the INN. After training the network, (3) is evaluated numerously to derive the joint distribution of p_i and g_j. Note that in these evaluations EH and EW are fixed while $\boldsymbol{Z}$ is sampled from its normal distribution. Afterwards, the closest available input parameters to the most likely point from distribution of p_i and g_j is selected as peaking frequency and gain value. If the distribution is multi-modal, multiple candidate designs are produced. Finally, the eye diagram is simulated for the design(s) to evaluate the resulting eye height and eye width, and compare with the target values. The proposed approach is implemented in Python 3.7 using the INN source code published in [8].

III. Numerical Example

To investigate effectiveness of the proposed approach, inverse design of CTLE for the SerDes channel, pictured in Fig. 2, is considered. This channel includes two processor packages, connected to the board with hybrid land grid array connectors. The board contains 4 inches of differential wiring in total, which is connected to the connectors with differential vias. No crosstalk is considered. This channel is simulated with a custom-build solver named HSSCDR [9], which is developed by IBM. Furthermore, the channel operates at 16 Gb/s; resulting in a unit interval of 62.5 ps. For the CTLE design, 10 possible peaking frequencies and 16 possible gain values are considered. In this paper, these values are shown symbolically as $[p_1, p_2, \ldots p_{10}]$ and $[g_1, g_2, \ldots g_{16}]$, which are ordered sequentially.

As a rule of thumb, often about 80% of the total number of samples is selected for training and validation, which is subsequently divided to 80% and 20% sections for training and validation, respectively. The remaining 20% of the total number of samples is used for testing, Therefore, From the 160 combinations of p_i and g_j, 102 samples are randomly selected for training the network. Another 25 random samples are used for validation and tuning the hyperparameters of the network. After tuning, number of the latent variables in $\boldsymbol{Z}$ is set to 2. INN is comprised of 4 reversible blocks. S_1, S_2, t_1, and t_2 are fully connected neural networks, and each of them has 1 hidden layer with 100 nodes and the ReLU activation function. Number of dimensions in the input and output is increased to 16 with zero-padding, and the training takes 200 epochs.

TABLE I. ACCURATE AND CANDIDATE CTLE DESIGNS AND THEIR CORRESPONDING EH AND EW.

Design	Peaking frequency	Gain value	EH (mV)	EW (ps)
Accurate	p_3	g_{12}	175	44.6
INN1	p_2	g_6	186	49.9
INN2	p_2	g_{12}	153	42.4
INN3	p_8	g_6	235	37.1
INN4	p_8	g_{12}	292	44.3

The remaining 33 samples are used for testing. The resulting EH and EW values, which are yielded by the inverse design shows a wide range of accuracy. For some test samples a good match with the desired EH and EW is achieved. While, for some others the results were not satisfactory. Next, we show the results for a case with moderate accuracy, and discuss the other test cases afterwards. In this test case EH is 175 mV and EW is 44.6 ps. We sampled (3) for 30,000 times to derive distribution of p_i and g_j. Note that this evaluation is almost instant because INN translates to an analytical calculation. Joint distribution of p_i and g_j, and their marginal distributions are shown in Fig. 3 and Fig. 4, respectively. It is seen that the distribution is multi-modal. In other words, the proposed approach suggests four possible solutions which are (p_2, g_6), (p_2, g_{12}), (p_8, g_6), and (p_8, g_{12}). These solutions and their corresponding EH and EW values are presented in Table I. In addition, the accurate selection of p_i and g_j for the desired EH and EW is included in Fig. 3, Fig. 4, and Table I. We know the accurate p_i and g_j since we had swept over all of their possible values. From the results in Table I it is observed that the INN approach has achieved a design (INN2) which is only one step away from the accurate design in peaking frequency, and it has the same gain value. EH and EW of this design are close to the target values. On the other hand, results of the INN1 design are also close to the target values, while its gain is not close to the accurate gain. This design can be selected if INN2 is not possible due to other constraints, and it shows that the INN approach can find multiple solutions for a single target objective. The eye diagram obtained from the INN2 design is illustrated in Fig. 5.

Although the results achieved in this test case are close to the target values, they are not a perfect match. In the 33 test cases, more accurate results were observed; however, some other test cases had a higher mismatch rate, including some unacceptable results. Overall, we conclude that the INN structure is not a universal solution in its current state and needs improvements. One of the issues that can cause the mismatch is handling of discrete variables. The proposed approach derives the CTLE variables by selecting the closest possible values to the candidate points taken from the distribution provided by INN. However, it is seen in Table I that even one step mismatch in the peaking frequency can result in nontrivial mismatch with the target values. In addition, in this work we performed inverse design for target values that have at least one existing solution. If the solution does not exist, the algorithm needs to provide the closest solution. Moreover, for this study it would be interesting to examine lossier channels as well, and see if the multi-modal behavior persists. Addressing these issues is left for future work.

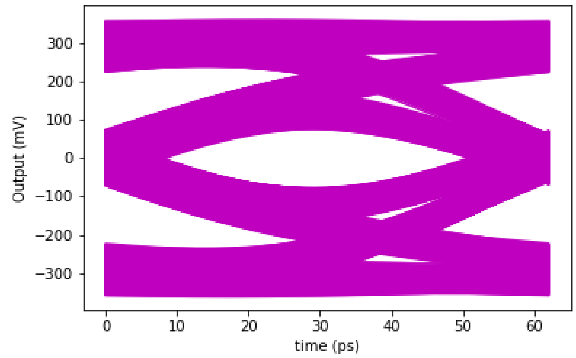

Fig. 5. Eye diagram of the channel in the numerical example when the INN2 design is used for CTLE.

IV. CONCLUSION

In this paper an approach for inverse design of CTLE of high-speed channels is proposed in order to increase the design efficiency. The algorithm receives the desired eye height and eye width, and it derives the required peaking frequency and gain of CTLE. This approach is based on invertible neural networks, which can be trained and used in both directions. An example with moderate accuracy is provided. However, it is observed that the algorithm can produce inaccurate results in some other test cases. Therefore, improvements to the algorithm are needed.

REFERENCES

[1] M. Ohira, A. Yamashita, Z. Ma and X. Wang, "Automated Microstrip Bandpass Filter Design Using Feedforward and Inverse Models of Neural Network," in *Asia-Pacific Microwave Conference (APMC)*, 2018.

[2] D. Zibar, A. M. R. Brusin, U. C. Moura, F. D. Ros, V. Curri and A. Carena, "Inverse System Design Using Machine Learning: The Raman Amplifier Case," *Journal of Lightwave Technology,* vol. 38, no. 4, pp. 736-753, 2020

[3] H. Ma, E. Li, A. C. Cangellaris and X. Chen, "High-Speed Link Design Optimization Using Machine Learning SVR-AS Method," in *IEEE 29th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2020.

[4] K. Roy, M. Ahadi Dolatsara, H. M. Torun, R. Trinchero and M. Swaminathan, "Inverse Design of Transmission Lines with Deep Learning," in *IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2019.

[5] R. Trinchero, M. Ahadi Dolatsara, K. Roy, M. Swaminathan and F. G. Canavero, "Design of high-speed links via a machine learning surrogate model for the inverse problem," in *Electrical Design of Advanced Packaging and Systems (EDAPS)*, 2019.

[6] H. Yu, H. M. Torun, M. U. Rehman and M. Swaminathan, "Design of SIW Filters in D-band Using Invertible Neural Nets," in *IEEE/MTT-S International Microwave Symposium (IMS)*, 2020.

[7] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother and U. Köthe, "Analyzing Inverse Problems with Invertible Neural Networks," in *International Conference on Learning Representations*, 2019.

[8] L. Ardizzone, J. Kruse and et al., "Framework for Easily Invertible Architectures (FrEIA)," Source code, [Online]. Available: https://github.com/VLL-HD/FrEIA. [Accessed June 2020].

[9] S. Chun and e. al., "Package and printed circuit board design of a 19.2 Gb/s data link for high-performance computing," in *IEEE 67th Electronic Components and Technology Conference (ECTC)*, 2017.