TELKOMNIKA, Vol.16, No.4, August 2018, pp. 1607~1615
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v16i4.8176

Deep Learning for Tuning Optical Beamforming Networks

Herminarto Nugroho*, Wahyu Kunto Wibowo, Aulia Rahma Annisa, Hanny Megawati Rosalinda
Department of Electrical Engineering, Universitas Pertamina.
Jalan Teuku Nyak Arief, Kebayoran Lama, Jakarta, 12220, Indonesia
Corresponding author, e-mail: herminarto.nugroho@universitaspertamina.ac.id[1], wahyu.kw@universitaspertamina.ac.id[2], aulia.ra@universitaspertamina.ac.id[3], hanny.mr@universitaspertamina.ac.id[4]

Abstract

In communication between planes and satellites, Optical Beamforming Networks (OBFNs), which rely on many small and flat Phased Array Antennas (PAAs), need to be tuned in order to receive signals from specific angles. In this paper, we develop a deep neural network representation of tuning OBFNs. The problem of tuning an OBFN is in many aspects similar to training a deep neural network. We present a way to exploit the special structure of OBFNs into deep neural network and an algorithm for tuning OBFNs based on feedback that can be easily measured in real system. Training data, which consists of full signals, can be measured, and therefore is used in this paper. For pilot signals, the desired signal is known explicitly. Given the configuration of OBFNs and all nominal parameters required, it was verified in simulation that the deep neural network can be used to tune large scale OBFNs for any desired delays.

Keywords*: Artificial neural network, Deep learning, Optical beamforming network, Optical ring resonator, Phased array antenna*

1. Introduction

The demand of being able to connect to internet has been increasing recently, even on intercontinental flights. In order to transmit/receive RF signals to/from satellite, the planes should focus the transmission beams towards the satellite. Ordinary omni-directional antennas are not preferable because, although they are directional-sensitive, they have low gain [1]. The conventional solution is to steer dish antennas mechanically, which has some disadvantages such as high maintenance cost, large dimension and increased drag forces [2, 3].

Phased Array Antenna (PAA) system [4] is potential solution, because of its agility, low maintenance cost, and reduce drag forces [5]. A PAA system mainly consists of an array of antenna elements (AEs) and a beamforming network, as illustrated in Figure 1. Each AE receives a time-delayed version of desired signal from specific angle. The received signals will go through RF paths in which they are delayed with pre-determined delay values. This is done to make sure the signal arrives first to the bottom-most path, which will simplify the tuning process [2]. After that the signal will go through a beamforming network, where the delay values are tuned to match the desired delays [6].

The beamforming network is a delay-and-combine network, by which the desired signal adds up in phase [7]. In this paper, the desired signal and its time-delayed version are known explicitly. An optical beamforming network, with optical ring resonators (ORRs) as tunable delay elements, is used in this paper [8,9,10]. Non-linear programming optimization [11,12], genetic algorithm [13] were used to get the optimum parameter of the ORRs. However, they cannot exploit the special structure of OBFNs, making them not suitable for tuning large-scale OBFNs [14]. The problem of tuning OBFNs is in many aspects similar to training a deep neural network. It has been proven that neural network has been tremendously used to solve wide range of problems, such as in biomedics [15], fault detection system [16], power systems [17], face recognition [18], and telecommunication systems [19].

Received December 1, 2017; Revised April 18, 2018; Accepted May 10, 2018

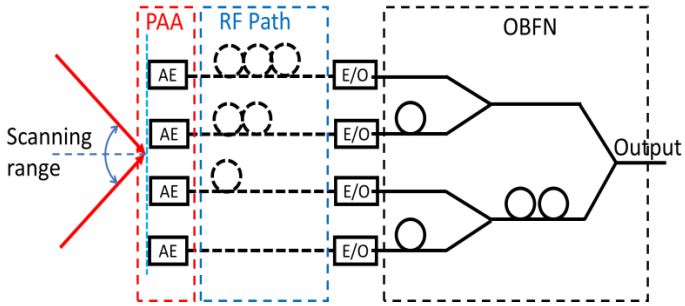

Figure 1. A phased array antenna (PAA) system, antenna elements (AEs) and optical beamforming network (OBFN) [2].

The paper is organized as follows. In Section 2, we derive the mathematical model of ORR. In Section 3, we describe how to exploit the special structure of OBFNs into deep neural network representation. In Section 4, the simulation results of tuning OBFNs using deep learning are presented. Finally, some conclusions are presented in Section 5.

2. Mathematical Model Of Optical Ring Resonator

A simple one-input one-output single-stage ORR is illustrated in Figure 2 (Left). It consists of a ring-shaped and a straight waveguide. The parameter κ is the power coupling coefficient, which has a value between 0 and 1, L_R is the round-trip length of the ring-shaped waveguide, T is the round-trip period, and ϕ is the extra phase-shift due to heater on the top of the ring.

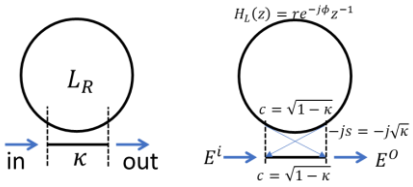

Figure 2. (Left) Structure of a 1 × 1 single stage ORR. (Right) z-transform schematic of ORR [2].

The Z-transform of an ORR is illustrated in Figure 2 (Right). Let the signal at the right and left side of the ring be E^r and E^l respectively, then one can derive the following relations:

$$E^r = -j\sqrt{\kappa}E^i + \sqrt{1-\kappa}E^l \quad (1)$$

$$E^l = \frac{-j\sqrt{\kappa}\, r\, z^{-1}e^{-j\phi}E^i}{1-r\sqrt{1-\kappa}\, z^{-1}e^{-j\phi}} \quad (2)$$

$$E^o = \frac{\sqrt{1-\kappa}-rz^{-1}e^{-j\phi}}{1-r\sqrt{1-\kappa}z^{-1}e^{-j\phi}}E^i \quad (3)$$

where r defines the power loss. Since the frequency response is defined as $H(f) = E^o/E^i$, and substituting $z^{-1} = e^{-2\pi jfT}$ with T being the round-trip period, we obtain the equation for frequency response of an ORR:

$$H(f) = \frac{\sqrt{1-\kappa}-re^{-j(2\pi fT-\phi)}}{1-r\sqrt{1-\kappa}e^{-j(2\pi fT-\phi)}} \quad (4)$$

Equation (4) is the same as the equation (2.18) in [2] and equation (2.52) in [20].

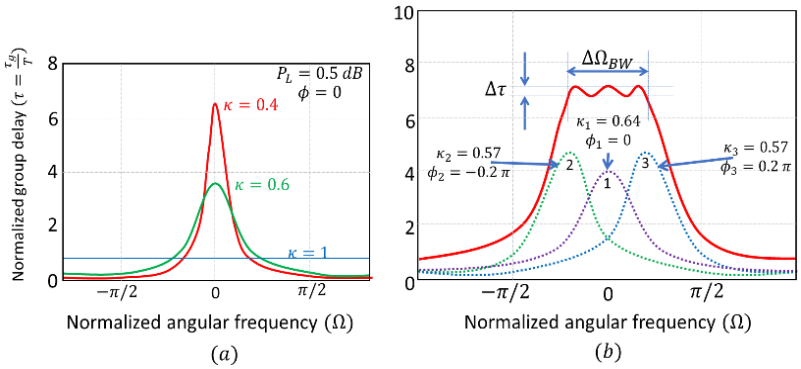

Figure 3. Group delay response of (a) single ORR, (b) cascade of multiple ORRs.

Figure 3 (a) shows when the group delay increases, the width of the delay curve decreases. This is due to the fact that the area under the group delay curve represents the phase shift of the ORR, which is constant 2π for one free spectral range (FSR) [21]. This observation reveals the tradeoff between the delay value and the bandwidth. A single ORR will not be able to cover large bandwidth and high desired delays at the same time. We can use cascade of multiple ORRs to solve this problem, as illustrated in Figure 3(b). The frequency response of N-stage cascade ORRs in normalized angular frequency is defined by the product of those of the individual single-stage ORRs:

$$H_{total}(f) = \prod_{i=1}^{N} H_i(f), \tag{5}$$

where $H_i(f)$ is the frequency response of a single ORR in the stage i.

3. Deep Neural Network Representation of Optical Beamforming Networks

3.1. Feed-forward Neural Network System

The OBFN structure is desired to be low-cost, scalable, has minimum number of ORRs, and able to cover wide bandwidth. Zhuang [2] found in his experiment that, for the same antenna specifications and bandwidth, asymmetrical binary-tree-structured OBFNs, illustrated in Figure 1 and its neural network representation in Figure 4, is scalable and has the least number of ORR.

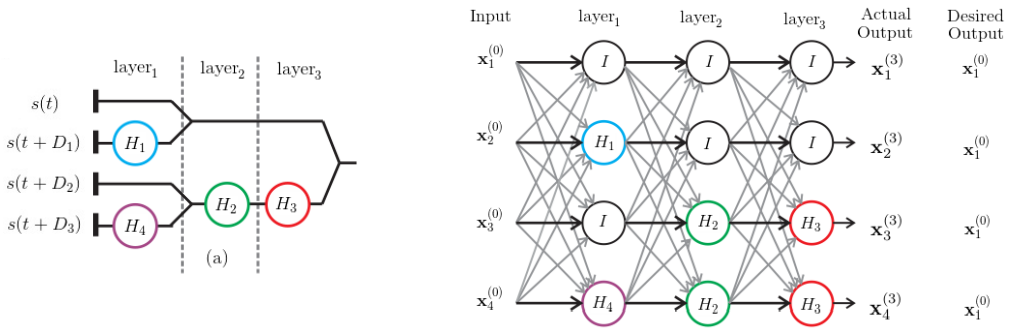

Figure 4. (left) 4×1 OBFN system, and (right) its neural network configuration.

3.2. Generating Training Examples

Training examples of a certain neural network consist of input vectors and their respective desired output. From Figure 4, the input to the neural network is the signal received by each antenna element. The signal received by the reference path is the desired output. In

this paper, it is assumed that the signal is coming from a satellite, which is very far, and without any aberration from the atmosphere. Therefore, noise and other signal from different direction are omitted, and input signals arriving at each antenna element are parallel to each other. The inputs shown in Figure 4 (a) are in time domain. Since, we will use frequency response of ORR, it is convenient to transform the inputs into frequency domain using Fourier transformation, as follows:

$$\begin{aligned} x_1^{(0)}(f) &= \mathcal{F}\{s(t)\}, \\ x_2^{(0)}(f) &= \mathcal{F}\{s(t+D_1)\} = x_1^{(0)}(f)e^{i2\pi f D_1}, \\ x_M^{(0)}(f) &= \mathcal{F}\{s(t+D_{M-1})\} = x_1^{(0)}(f)e^{i2\pi f D_{M-1}}. \end{aligned} \quad (6)$$

From equation 6, the input is frequency dependent. Since we want to obtain desired group delay response over a certain frequency range, an array of signal for a set of frequencies is needed. Given a frequency range $f \in R^N = [\in (f_1 \dots f_N)]$, the input array for path m is defined by

$$\mathbf{x}_m^{(0)} = \begin{bmatrix} x_m^{(0)}(f_1) \\ \vdots \\ x_m^{(0)}(f_N) \end{bmatrix} = \begin{bmatrix} x_1^{(0)}(f_1)e^{i2\pi f_1 D_{m-1}} \\ \vdots \\ x_1^{(0)}(f_N)e^{i2\pi f_N D_{m-1}} \end{bmatrix} \in \mathbb{C}^N \quad (7)$$

Consider a lossless system, i.e., power loss $r = 1$ and gain response $|H(f)| = 1$, then the desired output for path m of the network is defined by

$$\mathbf{d}_m = \begin{bmatrix} x_1^{(0)}(f_1) \\ \vdots \\ x_1^{(0)}(f_N) \end{bmatrix} \in \mathbb{C}^N \quad (8)$$

For a lossy system, where $r < 1$, the magnitude (gain) response of ORR $|H(f)| \neq 1$, will not be covered in this paper.

3.3. Weight Matrices

Consider an $M \times 1$ binary-tree OBFN, which is represented by a neural network whose $M \times N$ neurons and L layers, where N specifies the number of frequency of interest. The input vectors are propagated through layers using weight matrices $W^{(l)} \in C^{MN \times MN}$ as follows:

$$\mathbf{y}^{(l)} = \mathbf{W}^{(l)}\mathbf{x}^{(l-1)}, \text{and } \mathbf{x}^{(l)} = F\left(\mathbf{y}^{(l)}\right), \quad (9)$$

where l specifies the layer index, and F is the activation function, which is defined as

$$F : \mathbb{R}^N \longrightarrow \mathbb{R}^N, F(s) = s\,. \quad (10)$$

Note that the structure of the weight matrices depends on the configuration of the neural network, because different neural network leads to different location of frequency response H_j inside weight matrices. However, determining the weight matrices from a given neural network configuration is straight forward. Consider a neural network representation of 4×1 OBFN shown in Figure 5, the respective weight matrices are

$$\mathrm{W}^{(l)} = \begin{bmatrix} \mathrm{I} & 0 & 0 & 0 \\ 0 & \mathrm{H}_1 & 0 & 0 \\ 0 & 0 & \mathrm{I} & 0 \\ 0 & 0 & 0 & \mathrm{H}_4 \end{bmatrix}, \mathrm{W}^{(2)} = \begin{bmatrix} \mathrm{I} & 0 & 0 & 0 \\ 0 & \mathrm{I} & 0 & 0 \\ 0 & 0 & \mathrm{H}_2 & 0 \\ 0 & 0 & 0 & \mathrm{H}_2 \end{bmatrix}, \mathrm{W}^{(3)} = \begin{bmatrix} \mathrm{I} & 0 & 0 & 0 \\ 0 & \mathrm{I} & 0 & 0 \\ 0 & 0 & \mathrm{H}_3 & 0 \\ 0 & 0 & 0 & \mathrm{H}_3 \end{bmatrix}, \quad (11)$$

where $\mathbf{0} \in R^{N \times N}$ specifies the zero matrix, $\mathbf{I} \in R^{N \times N}$ specifies the identity matrix and $\mathbf{H}_j$ represents the frequency response matrix of ORR number j, which is defined by

$$\mathbf{H}_j = \begin{bmatrix} H_j(f_1) & 0 & \dots & \vdots \\ 0 & H_j(f_2) & \dots & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & \dots & H_j(f_N) \end{bmatrix} \in \mathbb{C}^{N\times N}, \quad (12)$$

Note that we should carefully consider coupling weights, in which some weights represent the same frequency response of a certain ORR, as we can observe in equation 11 and 12.

3.4. Non-linear Optimization

Given S number of training examples, the deep learning algorithm trains the neural network to obtain the optimum value of all R number of ORR's parameters (all κ and ϕ) by minimizing the cost function

$$C(\kappa_1, \phi_1, \dots, \kappa_R, \phi_R) = \frac{1}{2}\frac{1}{U}\sum_{\alpha=1}^{U}\left\|\mathbf{d}[\alpha] - \mathbf{x}^{(L)}[\alpha]\right\|^2 \quad (13)$$

subject to constraints

$$\epsilon \le \kappa_j^{(i)} \le 1-\epsilon \text{ , and } 0 \le \phi_j^{(i)} \le 2\pi \quad (14)$$

via stochastic gradient projection, with $\epsilon > 0$ is a small number, α is the training example index, and U is the number of stochastic training examples randomly chosen from S training examples.

Note that stochastic gradient projection [22] is used because the constraints make the gradient descent does not work. Stochastic gradient descent is an appropriate choice because of its simplicity which results in faster computation. The parameters of ORRs are updated via gradient projection methods as follows:

$$\begin{bmatrix}\kappa_1\\ \vdots \\ \kappa_R\end{bmatrix} \leftarrow \begin{bmatrix}\kappa_1\\ \vdots \\ \kappa_R\end{bmatrix} - \lambda\mathbf{P}\begin{bmatrix}\frac{\partial C}{\partial \kappa_1}\\ \vdots \\ \frac{\partial C}{\partial \kappa_R}\end{bmatrix} \text{ and } \begin{bmatrix}\phi_1\\ \vdots \\ \phi_R\end{bmatrix} \leftarrow \begin{bmatrix}\phi_1\\ \vdots \\ \phi_R\end{bmatrix} - \lambda\mathbf{P}\begin{bmatrix}\frac{\partial C}{\partial \phi_1}\\ \vdots \\ \frac{\partial C}{\partial \phi_R}\end{bmatrix} \quad (15)$$

where λ is sufficiently small learning rate and $\mathbf{P}$ is the projection matrix. The term $\frac{\partial C}{\partial \kappa_n}$ and $\frac{\partial C}{\partial \phi_n}$ specify the gradient of the cost function with respect to κ and ϕ respectively, which are formulated by

$$\frac{\partial C}{\partial \kappa_r} = \frac{\partial C}{\partial w_{i,j}}\frac{\partial w_{i,j}}{\partial \kappa_r} \text{ and } \frac{\partial C}{\partial \phi_r} = \frac{\partial C}{\partial w_{i,j}}\frac{\partial w_{i,j}}{\partial \phi_r} \quad (16)$$

where $w_{i,j}$ specifies the (i,j)-th element of weight matrix.

3.5. Backpropagation Algorithm

The non-linear optimization process using stochastic gradient projection requires the information of the gradient of the cost function with respect to all parameters in equation 16. Backpropagation algorithm is the most efficient way to find the gradient of the cost function by applying the chain rule in reverse order.

3.5.1. Gradient for the last layer (layer L)

From equation 13, we can determine the partial derivative of the cost function with respect to $x_k^{(L)}$, where $x_k^{(L)}$ specifies each element of matrix $\boldsymbol{x}^{(L)}$, as follows:

$$\frac{\partial C}{\partial x_k^{(L)}} = -\left(d_k - x_k^{(L)}\right), \quad (17)$$

where $d_k$specifies each element of matrix **d.** Using identity activation function (Equation (10)) and the chain rule, the partial derivative of the cost function with respect to $w_{i,j}^{(L)}$ is given by

$$\frac{\partial C}{\partial w_{i,j}^{(L)}} = \frac{\partial C}{\partial x_k^{(L)}} x_j^{L-1}. \quad (18)$$

3.5.2. Gradient for layer $l-1$

Let l any number in $\{2, \dots, L\}$ specifies the layer index. The partial derivative of the cost function with respect to $x_k^{(l-1)}$ is

$$\frac{\partial C}{\partial x_k^{(l-1)}} = \sum_{q=1}^{Q} \frac{\partial C}{\partial x_q^{(l)}} w_{q,k}^{(l)}, \quad (19)$$

where $q \in \{1, 2, \dots, M \times N\}$ defines the index of neuron. Then, the partial derivative of the cost function with respect to $w_{i,j}^{(l-1)}$ is

$$\frac{\partial C}{\partial w_{i,j}^{(l-1)}} = \frac{\partial C}{\partial x_k^{(l-1)}} x_j^{l-2}. \quad (20)$$

Since we should carefully consider coupling weights, i.e., some weights represent the same frequency response, equation 16 should be modified such that

$$\frac{\partial C}{\partial \kappa_r} = \sum_{\forall w_{i,j}=H_r} \frac{\partial C}{\partial w_{i,j}} \frac{\partial w_{i,j}}{\partial \kappa_r}, \text{ and } \frac{\partial C}{\partial \phi_r} = \sum_{\forall w_{i,j}=H_r} \frac{\partial C}{\partial w_{i,j}} \frac{\partial w_{i,j}}{\partial \phi_r}. \quad (21)$$

The remaining terms in equation (21) are the derivatives of weights (i.e., frequency response of ORR) with respect to parameters κ and ϕ. From the frequency response mentioned in equation 4, we obtain

$$\frac{\partial w_{i,j}}{\partial \kappa_r} = \frac{e^{-2i(2\pi fT+\phi_r)}-1}{2\sqrt{1-\kappa_r}\left(1-\sqrt{1-\kappa_r}e^{-i(2\pi fT+\phi_r)}\right)^2}, \text{ and } \frac{\partial w_{i,j}}{\partial \phi_r} = \frac{i\kappa_r e^{-i(2\pi fT+\phi_r)}}{\left(1-\sqrt{1-\kappa_r}e^{-i(2\pi fT+\phi_r)}\right)^2}. \quad (22)$$

This completes the formula to find the gradient of the cost function with respect to all parameters κ and ϕ, which enables the implementation of gradient projection method.

4. Simulation

4.1. Simulation Setup

The nominal parameters of OBFN setup simulated in this project are similar to the ones used in [14]. Table I shows all parameters needed and their value. The desired delays and training examples are given. Initial guesses of all parameter κ and ϕ are similar to [14].

Table 1. The Nominal Parameters of the OBFN

Symbol	Quantity	Value	Symbol	Quantity	Value
r	Power factor	1	f_c	Frequency center	107.52 Hz
$\|H(f)\|$	Magnitude of frequency response	1	ω	Bandwidth of interest	2 GHz
T	Round-trip period	0.08 ns	N	Number of frequency	100
λ	wavelength	1550 nm			

The results will show the group delay response graph and test error. This test error refers to the difference between the desired output and the actual output of the neural network for given training examples, which is stated in equation 13. It is also necessary to compute normalized squared group delay error which is defined by

$$\mathrm{E} = \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} \frac{\left\|D_{des,m}(f_n) - D_{act,m}(f_n)\right\|^2}{\left\|D_{des,m}(f_n)\right\|^2} \quad (23)$$

where M specifies the number of AE, $D_{des,m}$ and $D_{act,m}$ specify the desired and actual delay response of the m-th path respectively. The normalized squared group delay error is essential since it gives the comparison how big the error is compared to the desired delay response.

4.2. Simulation Result

Figure 5 shows group delay response and test error of simulation result of a 4×1 OBFN with desired delay [0 0.1 0.2 0.3] ns. Table 2 shows the optimum value of κ^* and ϕ^*, and the initial and final normalized squared group delay error (E). These optimum parameters are similar to the result found in [14].

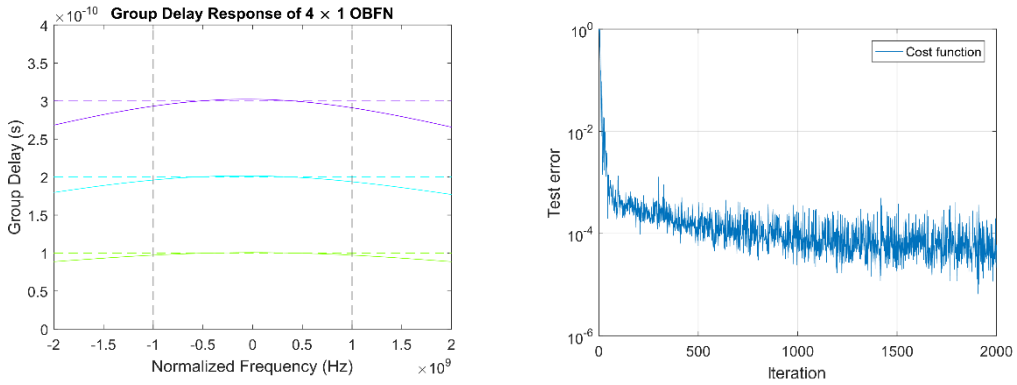

Figure 5. Group delay and test error of 4×1 OBFN with desired delay [0 0.1 0.2 0.3] ns.

Table 2. Comparison Between Initial - Optimum ORR Parameter Values

ORR	κ_0	ϕ_0	Initial E	κ^*	ϕ^*	Final E
1	0.9	0	0.2493	0.9859	0.0018	1.9×10^{-4}
2	0.9	5.88		0.9766	5.9191	
3	0.9	0.4		0.9726	0.3713	
4	0.9	0		0.9865	0.0027	

Figure 6 shows the group delay error for different values. Let the desired delays be $\delta \times [0\ 0.1\ 0.2\ 0.3]$ ns, where δ is a positive real number. One interesting thing is that the error increases as the delays becomes bigger. This is expected because of the trade-off mentioned in Section 2. When the delay becomes bigger, it is as expected that few ORRs cannot provide enough delay response, which will result in error becomes bigger as well.

Figure 7 show the group delay responses of 4×1 OBFN, with desired delay $\delta \times [0\ 0.1\ 0.2\ 0.3]$ ns where we use $\delta = 2$ and $\delta = 3$ respectively. We can observe that as the desired group delay increases, the ripple of the delay response will increase as well. This is why the error illustrated ini Figure 6 increases as desired delay increases.

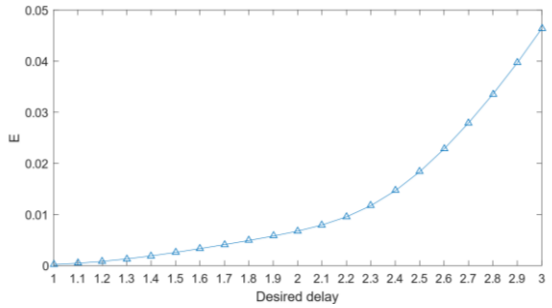

Figure 6. Group delay error of $\delta \times [0\ 0.1\ 0.2\ 0.3]$ ns of a 4×1 OBFN.

The deep learning algorithm aims to exploit the special structure of OBFN system such that it can tune large-scale OBFN setups. Figure 8 show the group delay responses of 8×1 and 16×1 OBFN setups. We can observe that the deep learning algorithm indeed can be used to tune larger OBFN setups.

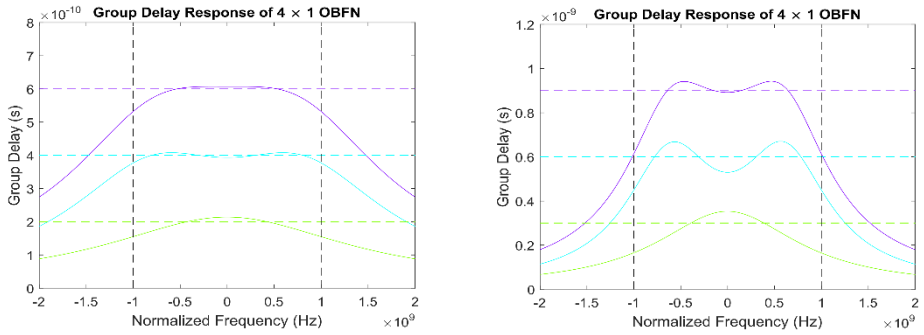

Figure 7. Simulation result of a 4×1 OBFN with desired delay (left) [0 0.2 0.4 0.6] ns and (right) [0 0.3 0.6 0.9] ns.

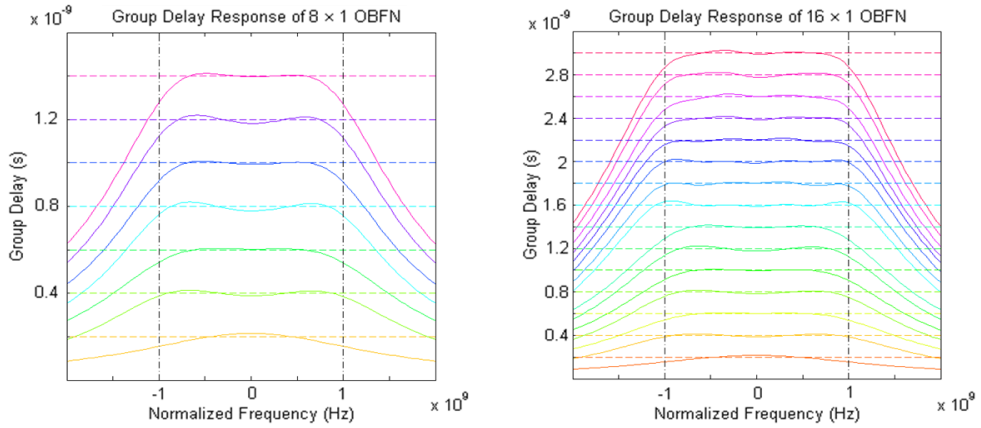

Figure 8. Simulation result of (left) 8×1 OBFN, and (right) 16×1 OBFN.

5. Conclusion

Optical Beamforming Networks (OBFNs) is used to control Phased Array Antennas (PAAs) such that planes can communicate to satellites. Tuning OBFNs is a highly non-linear and complex problem. An existing solution, a non-linear programming, is limited to small-scale OBFN setups. A deep learning algorithm, which can exploit the special structure of OBFN is proposed to tune large-scale OBFN setups. The special structure of OBFNs can be represented by a deep neural network. The weight matrices are composed of frequency response of some Optical Ring Resonators (ORRs) in the respective layer. Given a certain OBFN structure, a deep learning algorithm works well to find the optimum ORRs' parameters for 4×1, 8×1, and even 16×1 OBFN for any given desired delays. Another important thing is that the deep learning approach is data driven, which use measurable signal as a training examples. This is desirable because we can use real data as measurable signal, which is very essential for online tuning in future development.

Acknowledgement

The authors wish to thank Dr. Sander Wahls and Laurens Bliek from Delft Center for Systems and Control (DCSC), TU Delft, The Netherlands, for their assistance and advice. This

research was funded by Directorate of Research and Community Service, Directorate General of Research and Development Strengthening, Ministry of Research, Technology, and Higher Education, Republic of Indonesia, with Research Contract Number: 041/KM/PNT/2018, Date: March 6, 2018

References

[1] D Cheng, Field and Wave Electromagnetics, ser. The Addison-Wesley Series in Electrical Engineering. Addison-Wesley Publishing Company, 1989.
[2] L Zhuang. Ring resonator-based broadband photonic beamformer for phased array antennas. Ph.D. dissertation, University of Twente, November 2010.
[3] T Wilson, K Rohlfs, S Hüttemeister. Tools of Radio Astronomy, ser. Astronomy and Astrophysics Library. *Springer Berlin Heidelberg*, 2013.
[4] C Balanis. Modern Antenna Handbook. Wiley, 2008.
[5] RC Hansen. Phased Array Antennas. John Wiley & Sons, 2009; 213.
[6] A Meijerink, C Roeloffzen, L Zhuang, D Marpaung, R Heideman, A Borreman, W van Etten. *Phased array antenna steering using a ring resonator-based optical beam forming network*. in Proceedings of the IEEE Symposium on Communications and Vehicular Technology, Liege, Belgium. 2006: 7-12.
[7] A Meijerink, C Roeloffzen, R Meijerink, L Zhuang, D Marpaung, M Bentum, M Burla, J Verpoorte, P Jorna, A Hulzinga, W van Etten. Novel ring resonatorbased integrated photonic beamformer for broadband phased array receive antennas - Part I: Design and performance analysis. *Journal of Lightwave Technology*. 2010; 28(1): 3-18.
[8] G Lenz, B Eggleton, CK Madsen, R Slusher. Optical delay lines based on optical filters. *IEEE Journal of Quantum Electronics*. 2001; 37(4): 525–532.
[9] L Zhuang, CG Roeloffzen, W Van Etten. *Continuously tunable optical delay line*. in Proceedings of the IEEE Symposium on Communications and Vehicular Technology, Twente, The Netherlands, November 2005: p23.
[10] L Zhuang. Time-delay properties of optical ring resonators. Master's thesis, University of Twente, 2005.
[11] MS Bazaraa, HD Sherali, CM Shetty, Nonlinear Programming: Theory and Algorithms. John Wiley & Sons, 2013.
[12] JC Boot et al., Quadratic Programming: Algorithms, Anomalies, Applications, ser. Studies in Mathematical and Managerial Economics. Amsterdam: North. Holland Publishing Company, 1964.
[13] A García García et al. Optical phase synchronization in coherent optical beamformers for phased array receive antennas, Master's thesis, University of Twente, Enschede, February 2009.
[14] R Blokpoel. Staggered delay tuning algorithms for ring resonators in optical beamforming networks. Master's thesis, University of Twente, August 2007. [Online]. Available: http://doc.utwente.nl/62127/
[15] Wiharto, W, Kusnanto, H, Herianto, H. Hybrid System of Tiered Multivariate Analysis and Artificial Neural Network for Coronary Heart Disease Diagnosis. *International Journal of Electrical and Computer Engineering (IJECE)*. 2017; 7(2), 1023-1031.
[16] Boukadida, S, Gdaim, S, Mtiba, A. Sensor Fault Detection and Isolation Based on Artificial Neural Networks and Fuzzy Logic Applicated on Induction Motor for Electrical Vehicle. *International Journal of Power Electronics and Drive Systems (IJPEDS)*. 2017; 8(2).
[17] Sun, X, Sun, L, Zhao, S. Harmonic Estimation Algorithm based on ESPRIT and Linear Neural Network in Power System. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 2016; 14(3A): 47-55.
[18] Achmad, B, Firdausy, K. Neural Network-based Face Pose Tracking for Interactive Face Recognition System. *International Journal on Advanced Science, Engineering and Information Technology*. 2012; 2(1): 105-108.
[19] Dahiya, M, Gill, S. Detection of Rogue Access Point in WLAN using Hopfield Neural Network. *International Journal of Electrical and Computer Engineering (IJECE)*, 2017; 7(2): 1060-1070.
[20] DG Rabus, Integrated Ring Resonators: The Compendium, ser. Springer Series in Optical Sciences. Berlin, Heidelberg: *Springer*, 2007.
[21] JG Proakis, DG Manolakis. Digital Signal Processing: Principles, Algorithms, and Applications, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
[22] T van den Boom, D Schutter. Optimization in Systems and Control: Lecture Notes for the Course SC4091. TU Delft, 2012. [Online]. Available: http://books.google.nl/books?id=8f_GnQEACAAJ