논문 21-46-01-03 The Journal of Korean Institute of Communications and Information Sciences '21-01 Vol.46 No.01
https://doi.org/10.7840/kics.2021.46.1.18

대규모 다중 연결을 위한 인공 지능 메타 표면 통신

할디 조나단◆, 최 계 원°

Artificial Intelligence Metasurface Communications for Massive Connectivity

Jonathan Hardi◆, Kae Won Choi°

요 약

본 논문에서는 기존 MU-MIMO 시스템의 한계를 극복할 수 있는 방안으로서 재설정 가능 메타표면이 결합된 인공 지능 기반의 통신 시스템을 제안한다. 재설정 가능 메타표면은 단위셀의 반응을 디지털 입력을 통해 제어함으로써 소프트웨어 친화적인 데이터 형식을 처리하며 대면적의 평면 변조를 통해 다중 사용자를 지원할 수 있다. 본 논문에서는 재설정 가능 메타표면으로 구성된 송신단 제어를 위한 오토 인코더 기반 변조 알고리즘 제안하며 시뮬레이션 결과를 바탕으로 제안 된 시스템이 채널에 적응하여 적절한 변조를 수행할 수 있음을 보인다.

Key Words : Metasurface, Reconfigurable intelligent surface, Artificial intelligence, Deep learning

ABSTRACT

MU-MIMO systems today are limited by the tradeoff between capacity and cost. In this paper, we propose an artificial intelligence-based communication system as a new approach with potency to eliminate these tradeoff. Metasurface as a reconfigurable intelligent surface chosen to be the main transmitting component based on its capability to process software-friendly data format, which possibly increases flexibility on various multi-user cases. Based on simulation results, we show that the proposed system is advantageous in terms of significant design simplification with auto-encoder based deep learning systems.

Ⅰ. Introduction

Conventional massive MIMO systems nowadays are still very expensive to build. Mainly it is caused by the cost to build the system with full RF chains. Reconfigurable metasurface-integrated base station (BS) then seems as a considerable alternative, which is cheaper in cost for fabricating a large-aperture antenna array system. The use of reconfigurable metasurface-integrated BS can substitute the need of full RF chains for massive MIMO BS. In the long term, finding a replacement for the need for expensive RF chains then also brings us to the right path to the development of 5G communication, as massive MIMO systems has been one of its main key enablers[1].

※ This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (2014R1A5A1011478)

◆ First Author : Sungkyunkwan University, Department of Electronic, Electrical, and Computer Engineering, jo2019@g.skku.edu, 학생회원

° Corresponding Author : Sungkyunkwan University, Department of Electronic, Electrical, and Computer Engineering, kaewonchoi@skku.edu, 종신회원

논문번호 : 202009-241-A-RU, Received September 26, 2020; Revised October 17, 2020; Accepted October 27, 2020

Reconfigurable metasurface has been used to generate a complex-shaped beam in several studies like [2], [3], and [4]. In this paper we propose a down link system that directly encodes the reconfigurable metasurface at BS to generate a complex-shaped beam towards multiple receivers. We then optimize the communication system to make the signal constellations can be correctly formed in the receivers. Encoder and the reconfigurable metasurface at BS be the main thing focused to be designed, as this part is basically differentiating from the conventional massive MIMO systems.

Artificial intelligence (AI) here then has a high prominence since it is almost impossible to manually encode the optimal metasurface excitation (in form of coding pattern) to achieve high correctness of signal constellation at the receivers. The use of AI for communication systems at physical layers like this has been done before as in [5] and [6] with conventional plain direct transmitters used to produce the beam. Because metasurface is built from switch-able cells, it makes it easier for us to implement AI for encode the cells with a producible coding pattern. This paper follows the auto-encoder main principle to make sure communication is well performed in terms of reliability, to then train the reconfigurable metasurface coding pattern modulator based on that reliability error between the transmitted and received data.

This paper initiate novel design of using metasurface to generate signal at the transmitter side of an end-to-end communication system. The use of metasurface-integrated BS will open higher design flexibility and possibility to build a more efficient massive MIMO systems in term of cost (by the need of cheap large-aperture antenna array system fabrication) and time for according design complexity (because AI make no more need for manual encoding work). There will be presented the details of the metasurface system model, and proposed architecture of the machine learning system based on the auto-encoder algorithm. We also provide the training and testing simulation results at the end of the paper.

The rest of the paper is organized as follows. We present the system model in Section II. We investigate metasurface direct modulation schemes in Section III. The deep-learning-based metasurface modulation details are provided in Section IV. Section V presents the experimental results, and Section VI concludes the paper.

Ⅱ. System Model

2.1 Multi-User Metasurface Communication Architecture

As described on Fig. 1, the architecture of the proposed system can be separated into 3 parts: transmitter, channel, and receiver. At the transmitter part, there is data (in certain modulation format) and channel inputted to an encoder block. The encoder block then produces a pattern for a metasurface plane as the end of the transmitter part.

Metasurface plane then receives a signal from a horn shaped feeding antenna and directs the reflection to the user(s). Feeding antenna positioned 30 degree below x-axis and 30 degree toward y-axis to not block the intelligent reflection surface (IRS). Here signal outputted by the metasurface plane multiplied by channels and added by noises as in (19). These signals from the channel part are then received by each user and compared to the original data to gather specific loss values in the training phase.

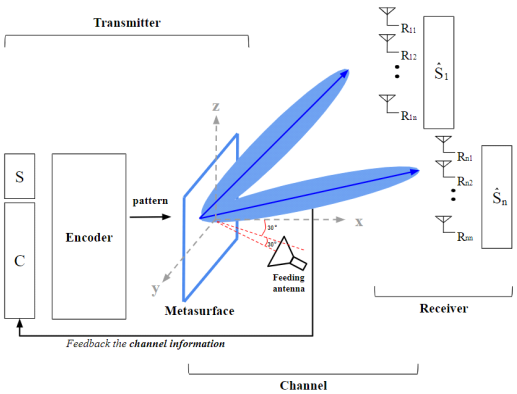

Fig. 1. MU-Metasurface Communication Architecture

2.2 Spatial Model of Metasurface Unit Cells and Receivers

In this subsection, we define the spatial model for 'the metasurface and receivers. We consider the planar metasurface on the y-z plane with N_y and N_x unit cells along y and z directions, respectively. The total number of unit cells is denoted by $N = N_y \times N_z$, and each unit cell is indexed by $n = 1,...,N$, The spacing between adjacent unit cells is denoted by d_y and d_z in y and z directions, respectively. Let $\boldsymbol{c}_n^u$ denote the position of unit cell n, and

$$\boldsymbol{c}_n^u = (c^{u_{x,n}}, c^{u_{y,n}}, c^{u_{z,n}}) = (0, (n_{y-1})d_y, (n_z - 1)d_z)^T, \tag{1}$$

where n_y and n_z are respectively the indices of unit cells towards y and z directions such that

$$n_y = ((n-1) \bmod N_y) + 1, \tag{2}$$

$$n_z = \lfloor (n-1)/N_y \rfloor + 1. \tag{3}$$

We assume that the feeding antenna to the metasurface is the fictitious isotropic antenna located at the Cartesian coordinate $\boldsymbol{c}^f = (c^{f_x}, c^{f_y}, c^{f_z})$. Let r^{fu_n}, θ^{fu_n}, and ϕ^{fu_n} respectively denote the radius, elevation, and azimuth of the spherical coordinate representation of the vector from unit cell n to the feeding antenna. Then, we have $r^{fu_n} = \| \boldsymbol{c}^f - \boldsymbol{c}_n^u \|$, $\theta^{fu_n} = \arccos((c^{f_z} - c^{u_{z,n}})/r^{fu_n})$, and $\phi^{fu_n} = \arctan2((c^{f_y} - c^{u_{y,n}}), (c^{f_x} - c^{u_{x,n}}))$. The power of the wave from the feeding antenna is denoted by P^f.

There are K receivers, which are located in the far-field region of the metasurface. The spherical coordinate of receiver k is denoted by $\boldsymbol{s}_k^{rx} = (r^{rx_k}, \theta^{rx_k}, \phi^{rx_k})^T$, where r^{rx_k}, θ^{rx_k}, and ϕ^{rx_k} are the radius, elevation, and azimuth of receiver k, respectively. Then, the Cartesian coordinate of receiver k is given by

$$\boldsymbol{c}_k^{rx} = r_k \boldsymbol{\Omega}(\theta^{rx_k}, \phi^{rx_k}), \tag{4}$$

where $\boldsymbol{\Omega}(\theta,\phi)$ is the unit vector towards the (θ,ϕ) -direction such that

$$\boldsymbol{\Omega}(\theta,\phi) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)^T. \tag{5}$$

We assume that the maximum gain direction of a receiver antenna faces towards the metasurface, and the maximum gain of a receiver antenna is denoted by G^{rx}. The receiver antenna is linearly polarized, and the polarization matches with the metasurface and feeding antenna. The effective length of the receiver antenna, denoted by δ^{rx}, is given as

$$\delta^{rx} = \frac{\lambda}{j}\sqrt{\frac{r^r}{4\pi\eta}}\sqrt{G^{rx}}, \tag{6}$$

where λ is the wavelength, j is the imaginary unit, η is the free space impedance (i.e., $\eta = 377\Omega$), and r^r is the radiation resistance of the receiver antenna.

2.3 Metasurface Unit Cell Model

In this paper, we consider a guided-wave type unit cell with one-bit control. It is noted that the proposed model is extended to more than one bit without significant modification. In Fig. 1, we consider the coordinate system where unit cell n is located at the origin and faces towards the x-direction. We assume that the unit cell is linearly polarized, and the E-fields of the incident and reflected waves point towards the elevation direction.

The incident wave comes from the direction (θ^i, ϕ^i), and the E-field of the incident wave at unit cell n is denoted by E^{i_n}. Unit cell n reflects the incident wave, and the E-field of the reflected wave observed at the spherical coordinate (r^r, θ^r, ϕ^r) is denoted by $E^{r_n}(r^r, \theta^r, \phi^r)$.

The incident free space wave is converted into a guided wave in the unit cell. The incident guided wave is reflected, and is radiated again to the air as the reflected free space wave.

The radiation resistance of the unit cell is denoted

by Z_u, and it is assumed that Z_u has no reactance (i.e., Z_u is a real value). We model each unit cell as an antenna of which the antenna port is connected to the circuit with a controllable impedance. Here, Z_u is the defined as the radiation resistance of this antenna model. The equivalent circuit is terminated by the controllable impedance $Z_c(b_n)$, where b_n is the control bit of unit cell n. We define $Z_c(b_n)$ as the complex impedance generated by the reactance load (e.g. PIN diode) according to the unit cell state b_n. The controllable impedance has two states, i.e. $Z_c(0)$ and $Z_c(1)$. The reference impedance of the guided wave can be arbitrarily determined for mathematical convenience. Here, we use the radiation resistance as the reference impedance. The reference impedance of the guided wave is Z_u, and the voltages of the incident and reflected guided waves of unit cell n are denoted by V^{i_n} and V^{r_n}, respectively.

The incident free space wave is converted into the incident guided wave according to the following formula.

$$V^{i_n} = \frac{\lambda}{j}\sqrt{\frac{Z_u}{4\pi\eta}}\Phi(\theta^i,\phi^i)E^{i_n}, \tag{7}$$

where $\Phi(\theta,\phi)$ is the radiation pattern of the unit cell. It is noted that the squared magnitude of the radiation pattern is equal to the directivity of the unit cell. The incident guided wave is reflected by the controllable impedance as in the following equation.

$$V_n^r = \Gamma(b_n)\, V_n^i, \tag{8}$$

where $\Gamma(b_n)$ is the reflection coefficient such that

$$\Gamma(b_n) = \frac{Z_c(b_n) - Z_u}{Z_c(b_n) + Z_u}. \tag{9}$$

The reflected guided wave is radiated and converted to the reflected free space wave as

$$E^{r_n}(r^r,\theta^r,\phi^r) = \sqrt{\frac{\eta}{4\pi Z_u}}\Phi(\theta^r,\phi^r)\frac{\exp(-jkr^r)}{r^r}V^{r_n}, \tag{10}$$

where $\kappa = 2\pi/\lambda$ is the wave number. From (7), (8), and (10), the relationship between the incident and reflected free space wave is given by

$$E_n^r(r^r,\theta^r,\phi^r) = \frac{\lambda}{4\pi j}\Phi(\theta^r,\phi^r)\Phi(\theta^i,\phi^i)\Gamma(b_n) \times \frac{\exp(-jkr^r)}{r^r}\cdot E_n^i, \tag{11}$$

In (11), we can see that the reflected wave from unit cell n can be controlled by the control bit b_n.

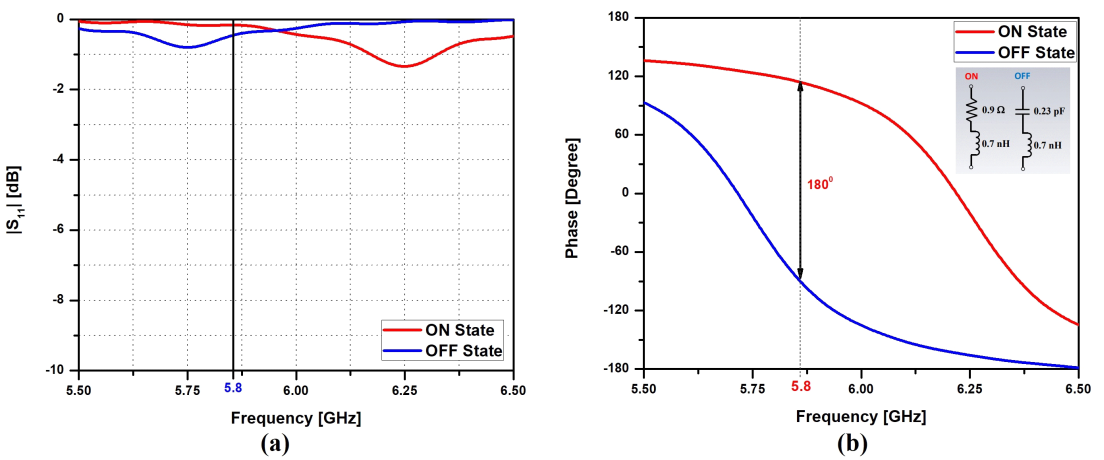

Fig. 2. Reflection coefficient to frequency

Here assumed $Z_c(b_n)$ is carefully designed so $\Gamma(b_n)$ has 180 degree phase difference when b_n is 0 and 1. The graph of the reflection coefficient $\Gamma_c(1)$ (ON) and $\Gamma_c(0)$ (OFF) according to the frequency change can be seen in Fig. 2[3].

2.4 Metasurface Reflection, Radiation, and Channel Model

Unit cell n receives the incident wave from the feeding antenna in the direction of $(r^{fu_n},\theta^{fu_n},\phi^{fu_n})$. The power density of the radiated wave from the isotropic feeding antenna is $P^f/(4\pi(r^r)^2)$ at the distance r^r. Then, the E-field of the incident wave at unit cell n is given by

$$E^{i_n}=\sqrt{\frac{\eta P_f}{2\pi}}\frac{\exp(-j\kappa r^{fu_n})}{r^{fu_n}}. \tag{12}$$

The E-field of the reflected wave from the metasurface observed at the spherical coordinate (r,θ,ϕ) within the far-field region is denoted by $E(r,\theta,\phi)$. We can calculate $E(r,\theta,\phi)$ as the summation of the reflected wave from all unit cells. From (11), (12), and the far-field approximation, we have

$$\begin{aligned}E(r,\theta,\phi)&=\sum_{n=1}^{N}E_n^r(r-\boldsymbol{c}_n^u\cdot\boldsymbol{\Omega}(\theta,\phi),\theta,\phi)\\&=\sum_{n=1}^{N}q_n(r,\theta,\phi)\Gamma(b_n),\end{aligned} \tag{13}$$

where

$$\begin{aligned}q_n(r,\theta,\phi)=&\frac{\lambda}{4\pi j}\sqrt{\frac{\eta P_f}{2\pi}}\Phi(\theta,\phi)\Phi(\theta^{fu_n},\phi^{fu_n})\\&\times\frac{\exp(-j\kappa(r+r^{fu_n}-\boldsymbol{c}_n^u\cdot\boldsymbol{\Omega}(\theta,\phi)))}{r\cdot r^{fu_n}},\end{aligned} \tag{14}$$

In (13) and (14), $\boldsymbol{c}_n^u\cdot\boldsymbol{\Omega}(\theta,\phi)$ is the deviation of the propagation distance due to the displacement of the unit cells from the origin, and is calculated from (1) and (5) as

$$\boldsymbol{c}_n^u\cdot\boldsymbol{\Omega}(\theta,\phi)=(n_y-1)d_y\sin\theta\sin\phi+(n_z-1)d_z\cos\theta. \tag{15}$$

We introduce the unit cell control variable x_n for unit cell n, which is 1 if $b_n=1$, and is -1 if $b_n=0$. Then, $\Gamma(b_n)$ is rewritten as

$$\Gamma(b_n)=\alpha x_n+\beta, \tag{16}$$

where $\alpha=(\Gamma(1)-\Gamma(0))/2$ and $\beta=(\Gamma(1)+\Gamma(0))/2$. Plugging (16) into (13) leads to the following linear equation in terms of x_n.

$$E(r,\theta,\phi)=\sum_{n=1}^{N}\alpha q_n(r,\theta,\phi)\cdot x_n+\sum_{n=1}^{N}\beta q_n(r,\theta,\phi). \tag{17}$$

Now, we can see that the E-field in (17) is the linear equation in terms of the unit cell control variable x_n.

The receive signal of the voltage at the antenna of receiver k is denoted by y_k.

Then, y_k is the E-field of the wave reflected from the metasurface multiplied by the effective length of the receiver antenna as in the following equation.

$$\begin{aligned}y_k&=\delta^{rx}E(r^{rx_k},\theta^{rx_k},\phi^{rx_k})\\&=\boldsymbol{h}_k^T\boldsymbol{x}+m_{k,}\end{aligned} \tag{18}$$

where $h_{k,n}=\delta^{rx}\alpha q_n(r^{rx_k},\theta^{rx_k},\phi^{rx_k})$ is the channel gain, $\boldsymbol{h}_k=(h_{k,1},...,h_{k,N})^T$ is the channel gain vector for receiver k, $\boldsymbol{x}=(x_1,...,x_N)^T$ is the metasurface control vector, and $m_k=\sum_{n=1}^{N}\delta^{rx}\beta q_n(r^{rx_k},\theta^{rx_k},\phi^{rx_k})$ is the uncontrollable reflection. Let $\boldsymbol{y}=(y_1,...,y_K)^T$ denote the receive signal vector. Then, the receive signal vector is given by

$$\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{m}, \tag{19}$$

where $\boldsymbol{H}=(\boldsymbol{h}_{1,...,}\boldsymbol{h}_K)^T$ is the channel gain matrix, and $\boldsymbol{m}=(m_1,...,m_K)^T$ is the uncontrollable

reflection vector.

Ⅲ. Metasurface Direct Modulation

3.1 Motivation

High performance communication systems nowadays have big trade offs with device cost and power efficiency. Looking forward, it is needed as 6G's more stringent requirement to achieve very high data rates (up to 1 Tb/s), very high energy efficiency (with the ability to support battery-free IoT devices), massive low-latency control (less than 1 millisecond end-to-end latency), very broad frequency bands (e.g. 73GHz−140GHz and 1THz −3THz[7]), ubiquitous always-on broadband global network coverage by integrating terrestrial wireless with satellite systems, and connected intelligence with machine learning capability[8]. The use of artificial intelligence possibly removes some complexity on designing with limited resources and low-cost devices. Metasurface based direct modulation then can simplify the need of massive antennas with capability to scatter signals for multiple users' cases.

3.2 Capacity Enhancement over Massive MU-MIMO

The use of modified auto-encoder algorithms potentially give a lot of eases in massive MU-MIMO build and maintenance. First, data trains can be easily provided, since auto-encoder is an unsupervised learning method that doesn't need label-embedded data for the training process[9]. In real implementation cases, this kind of data can just be retrieved directly from the environment. Second, since system parameters are set during the training process, complexity of more massive system issues is less vital. In hardware side, with using metasurface this eases potentially gained more since it has software-friendly input data type requirement, and also has more capability in beam-forming for serving the user(s).

Adaptivity reached the channel provided as input for the encoder. Channel estimation problems in different cases can then be solved with the same trained system. Although this concept is well implemented in simulation, it brings some challenges as explained at C subsection.

3.3 Challenges and Research Directions

Since this paper focuses on design analysis and software-based simulation implementation, the real situation cases bring some challenge to this study in the future. First, since channel retrieval in simulation is provided by random generated values for each frame, the real implementation will need more setup for extract channels by comparing some transmitted and received signals. Second, there will be hard to build a setup for doing signal correction at the receiver's side (as done in Section IV part D), since original modulated data are available at the transmitter's side. Third, is the long training time as explained in Section V, while real time situations may need more shorter training periods to keep updating and ensure system reliability periodically.

Ⅳ. Deep Learning-Based Metasurface Modulation

The proposed metasurface modulation scheme described as Fig. 3. The following subsections describe about the data sets used, how the encoder blocks encode the modulated data into coding

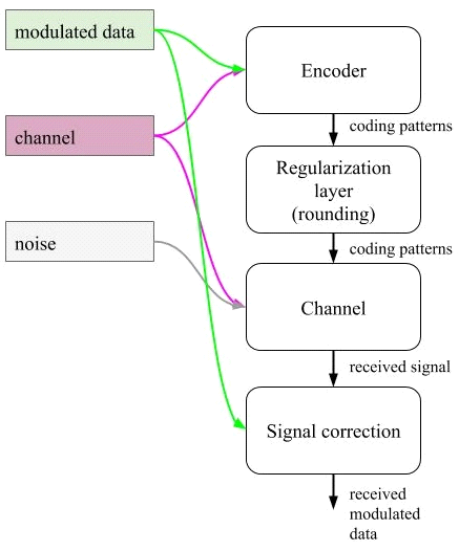

Fig. 3. Full System Block Diagram

patterns, the use of regularization layer, the signal transmission process through the channel, and signal correction process at the receiver side.

4.1 Data sets

Both on training and testing processes, the system retrieves 3 components of data sets: *modulated data*, *channel*, and *noise*. These data sets in real situations can be retrieved by observing the transmitted and received signal at certain areas as done by [3]. In this study, the data sets will be generated randomly, as the system basically is an unsupervised learning system which doesn't need any specific data sets. Data sets presented in NumPy's complex64 data format.

Modulated data set produced based on certain modulation types. In this study we provide a generator for QPSK modulation type data. The size of modulated data defined as [frame_size * frame_num, user_num]. The size of the noise data set is the same as the modulated data, and randomly (in normal distribution) generated for each sample. Each noise data set will be added to the received signal at the channel block.

Channel data set produced by simulating the metasurface's channel (as described in detail on Section II) with pre-defined feeding signal, size values, and other parameters needed. As every frame simulates a certain channel in a certain situation, the channel produces as much as the number of frames, then copied as much as the frame size in each frame. The size of the channel data set can then be defined as [frame_size * frame_num, user_num, cell_num], with each frame's samples being the same valued. The use of frames then will also be explained for signal correction in D subsection.

Noise data set produced by generating random value at specific ranges. The noise is set randomly in normal distribution with $(0+j0)$ average and $(0.224\times noise\ var + j0.224\times noise\ var)$ standard deviation, with $noise\ var$ defined at the beginning of training. The noise presence here is quite important, since without noise the data is possibly too close (in value) to each other, while the networks will be fooled with better score given to the regularization layer as less distance made, and regularization layer always expect the nearest result to -1 or 1. The noise data set forces the data to have enough distance from each other, resulting in a not over-fitted result, which won't change too much while the regularization layer rounding the pattern's value to -1 or 1 in testing phase.

4.2 Encoder Block Layers

The encoder block has 2 inputs: modulated data and channel, which use the data sets for this simulation. The data sets as much as the number of samples (defined as size of data frame multiplied by the number of the frames) is processed as described in detail in Fig. 4, which in each training batch, modulated data set processed as in Fig. 4a, channel data set processed as in Fig. 4b, and the concatenated results of those 2 processes that then finally processed as in Fig. 4c.

There are 3 blue colored blocks in Fig. 4 that are able to be heuristically customized to achieve optimal results. Each data that enters these blocks must not be in complex format, so there are some concatenations of real and imaginary value processes that can be found in the networks. The fully connected layers block in Fig. 4a can be filled with

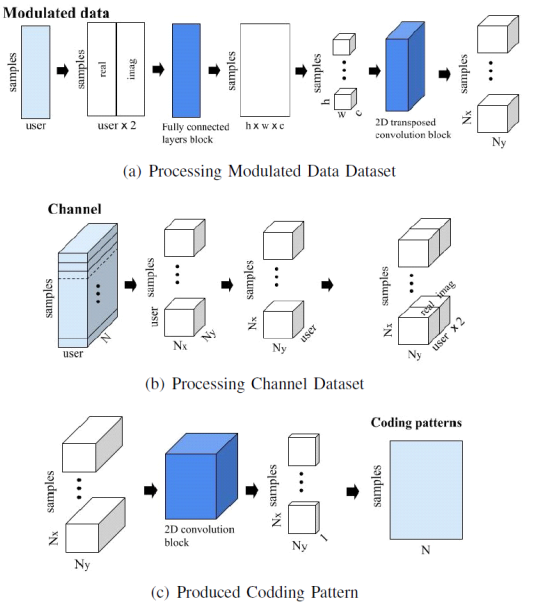

Fig. 4. Layers of Encoder Block

a neural-network layer to widen the data from [user_num*2] size to [h*w*c] size. The 2D transposed-convolution block in Fig. 4a then can be filled with a transposed-convolution layer(s) to widen the data from [h, w, c] size to [Nx, Ny, x] size with x as a flexibly defined variable. Finally the 2D convolution block in Fig. 4c can be filled with convolution layer(s) to extract the data from [Nx, Ny, x] size to [Nx, Ny, 1] size as each sample will produce a full coding pattern for numbers of cells of the defined metasurface plane.

As in Fig. 3, there is a regularization layer for the codding pattern before entering the channel block. This layer is presented in purpose to make the result as close as actual coding pattern by adding some loss value to the loss function with the distance of the results with -1 or 1.

4.3 Metasurface's Channel

Coding patterns produced by the encoder block then implemented to the metasurface cells. These coding patterns will be rounded to 1-bit state (-1 or 1) in testing phase only. In this study, simulation is done by multiplying the coding patterns to the channel data set, then adding the result with the noise data set, as in (19). This process output is the received signal at the receiver's side, with the same size as the modulated data set.

4.4 Signal Correction at Receiver's Side

Using the original modulated data $\bar{x}=(\bar{x_1},...,\bar{x_N})$ and received signal $x=(x_1,...,x_N)$, each frame of sample will be corrected at the signal correction block. The output of signal correction block is as in (24)

$$E[\bar{x}]=\frac{1}{N}\sum_{N}^{i=1}\bar{x_i}, \tag{20}$$

$$E[x]=\frac{1}{N}\sum_{N}^{i=1}x_i, \tag{21}$$

$$\alpha=\frac{(x-E[x])^H\cdot(\bar{x}-E[\bar{x}])}{(x-E[x])^H\cdot(x-E[x])}, \tag{22}$$

$$\beta=E[x]-(\alpha\cdot E[\bar{x}]), \tag{23}$$

$$F(\bar{x},x)=\alpha x+\beta, \tag{24}$$

while α (22) and β (23) are retrieved from the derived customized mean square error function that will be explained at the E subsection.

4.5 Loss Function

The loss function used in this system consists of pattern's regularization loss function (which mentioned before at the end of B subsection) and a customized mean squared error function. Loss value from these functions is then used for updating the weights of each node at each neural network layer of this system.

The pattern's regularization loss function as in

$$F(x)=(|x|-1)^2, \tag{25}$$

which means, as the value closer to -1 or 1, the loss value will be smaller. This function then used to proceed the coding pattern at the regularization layer block. The customized mean squared error function then is as in

$$F(\bar{x},x)=|\bar{x}-\alpha x+\beta|^2, \tag{26}$$

$$F(\bar{x},x)=|\bar{x}-E[\bar{x}]-\alpha(x-E[x])|^2, \tag{27}$$

Deriving this loss function will resulted α and β equation as in (22) and (23), which represent the equation that possibly gives the most minimum mean square error value.

V. Numerical Results

Simulation done for this paper using TensorFlow 2.0 library for Python 3.7. In the training phase, we run 180,000 batch of training, which in each batch using 128 data frames, with 64 samples in each frame, and presented for 2 and 4 users with Quadrature Phase Shift Keying (QPSK) and 16 Quadrature Amplitude Modulation (QAM) data modulation protocol.

The metasurface simulated has 16×16 cells ($N=256$) with $5.8GHz$ frequency ($\lambda=0.0517m$), $0.3\lambda\times0.3\lambda$ cell size, $0.7\lambda\times0.7\lambda$ cells spacing, $1W$ feeding antenna power from $(0.5,0,0)$ Cartesian coordinate, and 50Ω of radiation resistance of receive antenna. The noise data set (Section IV part A) will be generated at the beginning of the training with 1×10^{-6} of $noise\ var$ coefficient.

Parameters of layers (Section IV part B) are set in fixed value. The fully connected layer got $h=4$, $w=4$, and $c=2$. The transposed-convolution block is formed with parameters as in Table I. The convolution block layers with different number of layer and number of filter at each layer as Network A, Network B, Network C, and Network D (with the formers represent the smaller-sized networks, and the laters represent. the larger-sized networks). The details of convolution blocks layers shown in Table II.

Table 1. Transposed-Convolution Block Layers

	Num. of filters	Kernel Size	Strides
Layer 1	2	3×3	2×2
Layer 2	2	5×5	2×2

Table 2. Convolution Block Layers

Network A

	Num. of filters	Kernel Size	Strides
Layer 1	4	5×5	1×1
Layer 2	4	5×5	1×1

Network B

	Num. of filters	Kernel Size	Strides
Layer 1	64	5×5	1×1
Layer 2	32	5×5	1×1
Layer 3	4	5×5	1×1
Layer 4	4	5×5	1×1

Network C

	Num. of filters	Kernel Size	Strides
Layer 1	64	5×5	1×1
Layer 2	64	5×5	1×1
Layer 3	32	5×5	1×1
Layer 4	16	5×5	1×1
Layer 5	4	5×5	1×1
Layer 6	4	5×5	1×1

Network D

	Num. of filters	Kernel Size	Strides
Layer 1	64	5×5	1×1
Layer 2	64	5×5	1×1
Layer 3	32	5×5	1×1
Layer 4	32	5×5	1×1
Layer 5	16	5×5	1×1
Layer 6	16	5×5	1×1
Layer 7	4	5×5	1×1
Layer 8	4	5×5	1×1

5.1 Training

Training phase takes for training time about 10-20 minutes for 1000 training batches in each iteration. The regularization coefficient (as a multiplier factor of the layer loss) needed to be not activated at early batches, then the regularization layer loss factor could start to be initiated after the loss value seems like reaching it's optimal lowest value, signed with small loss value changes through iterations. As data shown in Fig. 5, here we initiate the regularization layer loss factor to 0.01 only after 57 batches of each case's training, because found that by heuristic approach, the training can't be optimized when trained with a synchronous start time, so it must be tuned sequentially.

Fig. 5 shows the loss values through training for 4 different cases, which set with different data modulation complexity (QPSK and 16-QAM) and different number of receiving users (2 and 4). Each graph presented 5 values (learning rate, regularization coefficient, regularization loss, mean squared error (MSE) loss, and bit error rate (BER) loss) in the same time frame, just to show the correlation between each data trend, and have no relevant meaning between each values discreetly. As more relevant to capture the training values trend of network with the best result with lowest BER loss value, here at Fig. 5 we only capture the "16-QAM for 4 users" case by Network D, and the rest by Network C.

As MSE loss gathered by average of distances of every received data from the actual data, it will show different information for us rather than the BER loss. By Fig. 6, we can find quiet similar values of MSE loss between "QPSK for 2 users" case and "QPSK for 4 users" case (which is around 0.01), but quite far difference of BER loss (around 0.001 for 2 users, and around 0.003 for 4 users). As

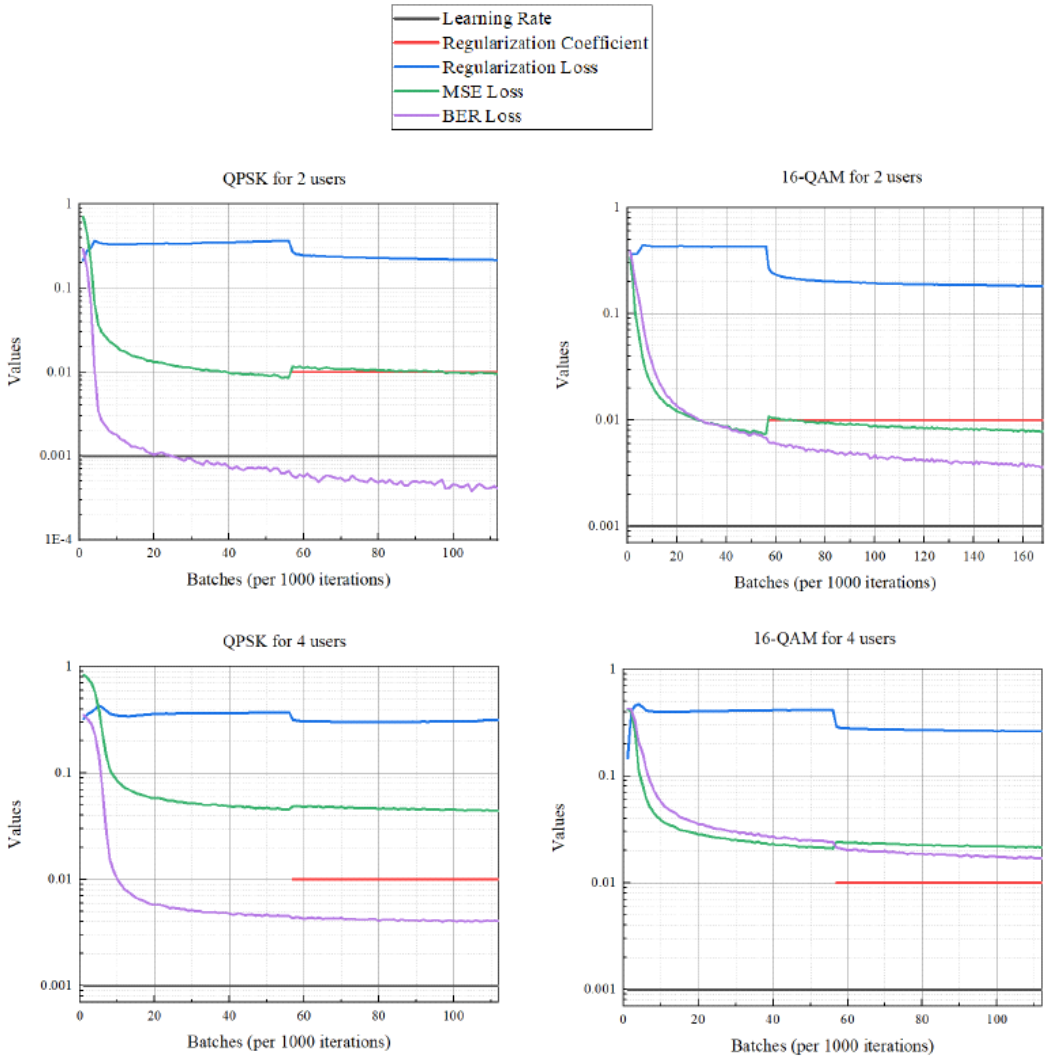

Fig. 5. Loss values through training

MSE loss being the parameter used for training the networks, this means within 2 cases the system has trained at the same level of MSE error, and just return worse end-result BER as higher case complexity in terms of number of users. The result for 16-QAM can't be compared the same way because we use different networks here.

Exclude "QPSK for 2 users" case, the other cases (which all have higher complexity) show more significant effect of regularization coefficient initialization. Regularization significantly raises the MSE loss, but also pushes down the regularization loss and BER loss value. This result aligned with

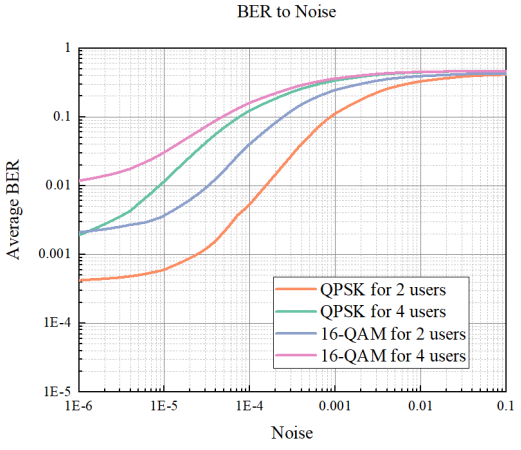

Fig. 6. BER test results to different factor of noise added to the original data before sent

the purpose of including regularization loss for the learning (network's weights value update) process to improve the actual system with binary (-1 or 1) metasurface coding pattern data.

Fig. 5 shows how the 4 cases result as the response to raised up noise. The noise represent the external disturbance that may change the original data value. The x-axis shown at this graph refers to the $noise\ var$ use as explained before on Section IV part A.

5.2 Testing

In Fig. 7 and Fig. 8, visualized channel and constellation diagrams from the simulation testing of Network D for the "16-QAM for 4 users" case, and Network C for the rests (Networks refer to Table II). Each case captured at 5 different frames, sampled from 64 frames resulted from a test batch. The first column diagrams show the channel heat map that changed by each frame. In the channel heat map, the color of each pixel at each 16-by-16 pixels picture represents the phase of a channel gain with the color bar mapping the phase to the color. The second column diagrams then show the constellation diagram of corrected data compared to the original data. The original data drawn as the red dots, and indicate the spot where the corrected data expected to take place. In most cases, the received data are successfully placed at the same data bits as the original modulated data initialized at the transmitter's side.

The evaluation result for 4 cases summarized at Table III). In the process of designing the training scheme, there are plenty of parameters including learning rate value, initial iteration of using regularization coefficient, the value of regularization coefficient used, and the total iteration number. We try several possibilities heuristically, and just put the best result of each case.

Overall result as seen on Table III shows best result given by Network D for the "16-QAM for 4 users" case, and Network C for the rest of cases. Interestingly, Network D (the one with deepest network layers) does not even perform better from Network B except for the "16-QAM for 4 users" case. Here can be concluded the layer deepness can fit differently depending on the case complexity faced. When most cases won by Network C, we can also conclude the system performance not proportionally correlated with the size of the network (which do not go any better with smaller-sized networks like Network A and Network B, neither with larger-sized network like Network D).

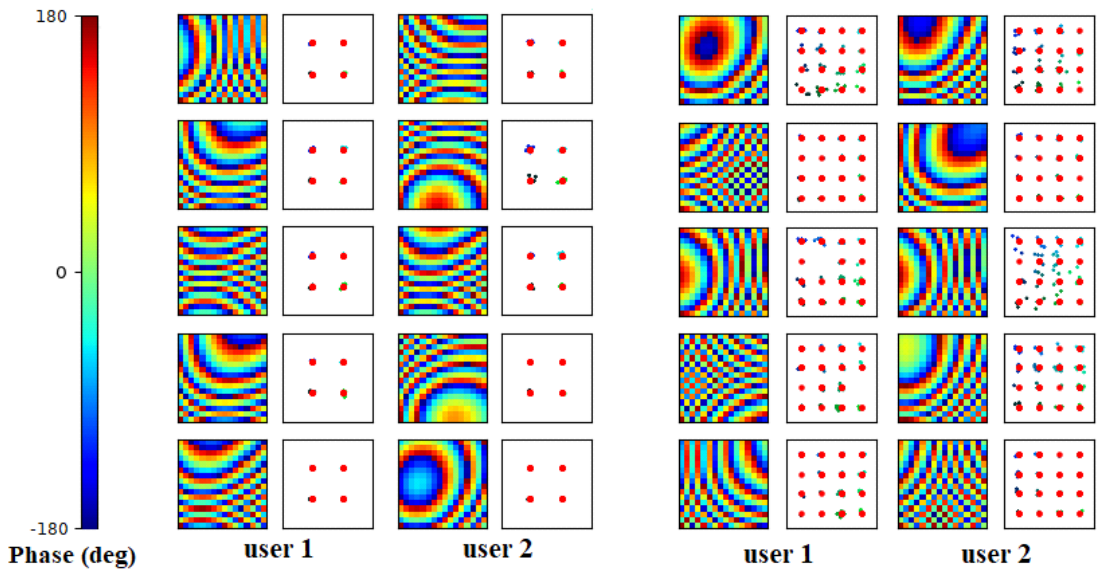

Fig. 7. Channel and Constellation Diagrams of; QPSK for 2 Users (left), 16-QAM for 2 users (right)

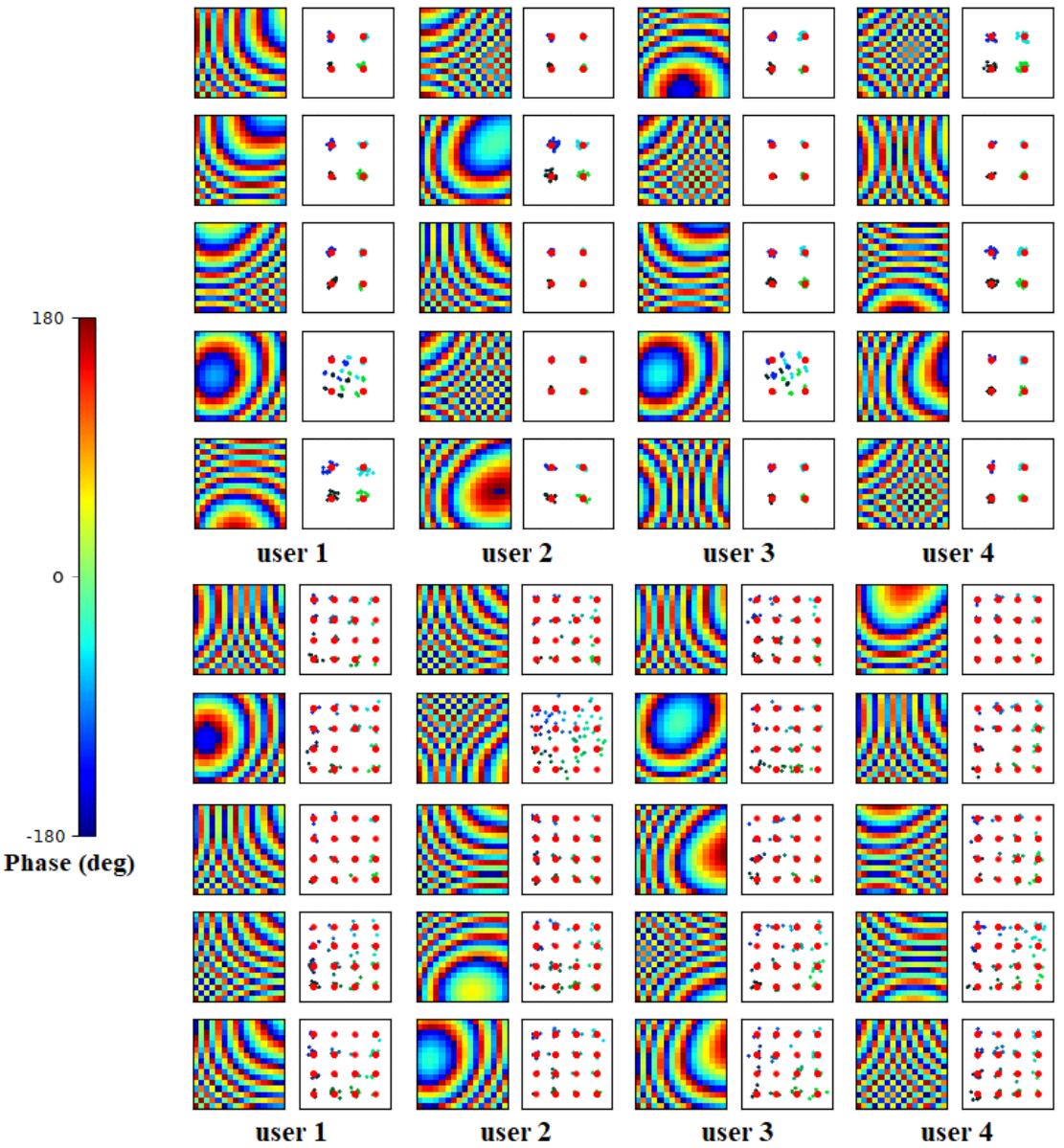

Fig. 8. Channel and Constellation Diagrams of; QPSK for 4 users (upper), and 16-QAM for 4 users (lower)

Ⅵ. Conclusion

In this paper, we have designed AI based MU-MIMO communication using metasurface, which proven through the learning process can work as a communication system for transfer data through differences in channel patterns. The system's components used including data sets, neural-network based encoder, and binary patterned metasurface transmitter also have been implemented with simulation on Python-TensorFlow platform and showing quite satisfying result with BER errror 0.04% for QPSK modulated data for 2 users, 0.41% for QPSK modulated data for 4 users, 0.36% for 16-QAM modulated data for 2 users, and 1.69% for 16-QAM modulated data for 4 users. By training simulation through this work, found also linear fitness relation between encoder network deepness

Table 3. Training Results Summary

Network A

	Reg. Loss	MSE	BER
QPSK – 2 Users	0.2713	0.0569	0.94%
QPSK – 4 Users	0.3323	0.1945	3.89%
16-QAM – 2 Users	0.2301	0.0229	7.72%
16-QAM – 4 Users	0.3593	0.3992	42.08%

Network B

	Reg. Loss	MSE	BER
QPSK – 2 Users	0.2029	0.0168	0.12%
QPSK – 4 Users	0.3001	0.0486	0.44%
16-QAM – 2 Users	0.1906	0.0106	0.54%
16-QAM – 4 Users	0.2778	0.0306	3.01%

Network C

	Reg. Loss	MSE	BER
QPSK – 2 Users	0.2148	0.0095	0.04%
QPSK – 4 Users	0.3136	0.0441	0.41%
16-QAM – 2 Users	0.1828	0.0078	0.36%
16-QAM – 4 Users	0.2527	0.0233	2.03%

Network D

	Reg. Loss	MSE	BER
QPSK – 2 Users	0.2654	0.0365	0.23%
QPSK – 4 Users	0.3090	0.0718	0.76%
16-QAM – 2 Users	0.2563	0.0253	4.26%
16-QAM – 4 Users	0.2641	0.0213	1.69%

with the case complexity, in terms of data modulation and number of users.

However, the use of data set may cannot be provided on the receiver side in real situations while the data only available at the transmitter side. We believe for the future research, the use of metasurface as transmitting media may give great flexibility through networking design, while great simplification of the designing process also can be achieved by the use of machine learning methods.

References

[1] M. Yao, M. Sohul, V. Marojevic, and J. H. Reed, "Artificial intelligencedefined 5G radio access networks," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 14-20, 2019.

[2] Q. Zhang, C. Liu, X. Wan, L. Zhang, S. Liu, Y. Yang, and T. J. Cui, "Machine-learning designs of anisotropic digital coding metasurfaces," *Advanced Theory and Simulations*, vol. 2, no. 2, p. 1800132, 2018.

[3] N. M. Tran, M. M. Amri, J. H. Park, S. I. Hwang, D. I. Kim, and K. W. Choi, "A novel coding metasurface for wireless power transfer applications," *Energies*, vol. 12, no. 23, p. 4488, 2019.

[4] L. Li, H. Ruan, C. Liu, Y. Li, Y. Shuang, A. Alù, C. W. Qiu, and T. J. Cui, "Machine-learning reprogrammable metasurface imager," *Nature Commun.*, vol. 10, no. 1, Jun. 2019.

[5] T. Oshea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognitive Commun. and Netw.*, vol. 3, no. 4, pp. 563-575, 2017.

[6] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive mimo for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027-3032, 2019.

[7] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 ghz: Opportunities and challenges for 6g and beyond," *IEEE Access*, vol. 7, pp. 78729-78757, 2019.

[8] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84-90, 2019.

[9] M. A. Kramer, "Nonlinear principal component analysis using auto associative neural networks," *AIChE J.*, vol. 37, no. 2, p. 233-243, 1991.

할디 조나단 (Jonathan Hardi)

2018년 5월 : Graduated from the Department of Electronics and Instrumentation, Gadjah Mada University

2019년 3월~현재 : 성균관대학교 전자전기컴퓨터공학과 석사과정

<관심분야> 인공 지능, 딥 러닝

최 계 원 (Kae Won Choi)

2007년 8월 : 서울대학교 전기컴퓨터공학부 박사

2010년 9월~2016년 8월 : 서울과학기술대학교 컴퓨터공학과 조교수

2016년 9월~현재 : 성균관대학교 전자전기컴퓨터공학과 부교수

<관심분야> 무선통신, 무선전력전송

[ORCID:0000-0002-3680-1403]