IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 5, 2020, accepted July 26, 2020, date of publication August 3, 2020, date of current version August 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013940

Online Regularization of Complex-Valued Neural Networks for Structure Optimization in Wireless-Communication Channel Prediction

TIANBEN DING[1] AND AKIRA HIROSE[2], (Fellow, IEEE)
[1]Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA
[2]Department of Electrical Engineering and Information Systems, The University of Tokyo, Tokyo 113-8656, Japan

Corresponding author: Akira Hirose (ahirose@ee.t.u-tokyo.ac.jp)

This work was supported in part by the JSPS KAKENHI under Grant 18H04105, and in part by the Cooperative Research Project Program of Research Institute of Electrical Communication (RIEC), Tohoku University.

ABSTRACT This article proposes online-learning complex-valued neural networks (CVNNs) to predict future channel states in fast-fading multipath mobile communications. CVNN is suitable for dealing with a fading communication channel as a single complex-valued entity. This framework makes it possible to realize accurate channel prediction by utilizing its high generalization ability in the complex domain. However, actual communication environments are marked by rapid and irregular changes, thus causing fluctuation of communication channel states. Hence, an empirically selected stationary network gives only limited prediction accuracy. In this article, we introduce regularization in updates of the CVNN weights to develop online dynamics that can self-optimize its effective network size in response to such channel-state changes. It realizes online adaptive, highly accurate and robust channel prediction with dynamical adjustment of the network size. We characterize its online adaptability in a series of simulations and our practical wireless-propagation experiments demonstrate that the proposed channel prediction scheme provides 2.5 dB and 5.5 dB improvement of bit error rate (BER) at 10^{-3} and 5×10^{-4}, and achieves 10^{-5} BER with $E_b/N_0 = 23 - 24$ dB.

INDEX TERMS Adaptive communications, channel prediction, channel state information (CSI), complex-valued neural network (CVNN), fading, 5G wireless communications (5G-NR).

I. INTRODUCTION

Performance of mobile communications always suffers from signal degradation, namely fading, due to path loss, shadowing, interference and channel state changes caused by movement of users [1]. In principle, fading can be mitigated by pre-equalization such as zero-forcing [2] or minimum-mean-square-error (MMSE) equalization [3]. Transmission power control is another countermeasure against the fading phenomenon [4]. These methods rely on accurate estimation of channel state information at communication ends. However, in practical mobile communications, the channel state, or simply channel, changes rapidly and irregularly due to time-varying multipath environments caused by movement of mobile users and their surroundings. The time fluctuation outdates the estimated channel and degrades the communication quality significantly. Channel prediction is an effective way to overcome this problem by forecasting channel changes over time based on preceding information. An accurate channel prediction is required for high communication quality and adaptive transmission in the next-generation communications [5], [6].

Several articles exist on the channel prediction in mobile communications including, for example, methods based on linear [7], [8] and autoregressive (AR) model extrapolation [9]–[12]. Although the low computational complexity in these methods is suitable for real-time operation in mobile communications, such simple linear or AR-model-based methods provide limited performance on predicting rapid and complicated channel changes [13]. Neural-network-based channel prediction methods have also attracted attention due to the recent successful development of artificial neural networks in various engineering fields. The generalization ability of neural networks provides flexible representation

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

of complicated channel-state changes and high prediction capability. For instance, prediction methods based on an echo state network (ESN) [14] and an extreme learning machine (ELM) [15] as well as real-valued recurrent neural network (RNN) [16], [17] have been reported, and their prediction performance has been evaluated in some simulated communication situations. Luo *et al.* recently combined a convolutional neural network and a long short-term memory (LSTM) network for learning and predicting channel states under some communication situations with static communication ends [18]. Similarly, one may employ a deep neural network for learning channel behavior in specific communication environment [19], [20]. However, channel states exhibit countless changes from moment to moment in actual mobile communications. Hence, a pre-trained network is not sufficient to represent all possible channel conditions in time-varying communications. Moreover, it is unpractical to re-train a huge neural network during communications due to its large computational cost and limited resources at mobile ends.

On the other hand, we also proposed a prediction method based on a multiple-layer complex-valued neural network (ML-CVNN) by focusing rotary motion of the channel state in the complex plane. In this method, an online training scheme was introduced in order to follow time varying channel states [13]. An empirically selected CVNN structure, however, is not sufficient for accurately predicting channels in some practical communications with rapid fading although it provides superior channel prediction performance in most other communication scenarios. Generally, in applications of neural networks, size of networks is critical to the application performance because it affects generalization characteristics and calculation costs [21], [22]. For example, a too small network is not enough to represent complexity of targets, resulting low convergence properties. In contrast, a too large network requires expensive calculation costs, and most importantly, it causes overfitting. Despite its importance, the structure of the network is typically defined based on a rule of thumb by users. One may start with an arbitrary structure and evaluate its learning performance using a large amount of training data by increasing or decreasing the number of neurons and network connections until the best structure is found. This manual pre-tuning of the network parameters is time consuming and inefficient. Moreover, from a mobile communications point of view, channel prediction is forced to work in more diverse communication environments, and experiences more rapid and various fluctuations in the real world than those in a training set. As a result, an *a priori* tuned structure under a situation is no longer optimal for other practical communication environments. The most suitable neural-network structure in a channel prediction method should also be dynamical according to the changes of communication environments. These motivate us to develop a neural-network scheme that optimizes network structures dynamically and adaptively but at the same time maintains the structures as small as possible for the channel prediction.

In this article, in order to realize a dynamically optimized network structure to suit best to the fading channel at each moment, we propose a new ML-CVNN-based channel prediction method by introducing regularization. We work with a moderate-size network platform and then let it automatically find, or self-adjust to, a suitable structure within the platform that uses only a limited portion of the network in order to achieve a good generalization. The self-adjustment is performed by imposing sparse constraints [23], [24] to connection weight updates. The sparse constraints suppress redundant connection weights to be zero, and equivalently construct a smaller scaled network using only the remaining non-zero connections [25]. Although introduction of sparsity in neural network is a relatively standard strategy for reduction of overfitting problem [26] and/or computational requirement upon practical implementation [27], [28], it has not been carefully discussed in the literature of complex-valued neural networks and in the context of online regularization. Here, we introduce L_1-norm (Lasso) and $L_{2,1}$-norm (group-Lasso) penalty into ML-CVNN updates as the constraints and validate performance of weight-level and neuron-level sparsity in the channel prediction. In order to follow time fluctuation of channel states, we develop an online training-and-prediction framework. We update the network by using a set of the most recent channels immediately before the prediction with a small number of learning iterations. We keep the updated network structure temporarily for the next training-and-predicting time frame. In this way, the non-zero connection distribution changes from time to time in the structure so that it keeps the most suitable size of the network for each prediction situation.

In each training phase, we use a backpropagation of teacher signal (BPTS) [29], rather than the standard error-backpropagation. The BPTS-based update method is simpler and has a lower computational cost, which is preferred for mobile communications. We demonstrate that the new channel prediction methods with the online adaptive CVNN structures present highly accurate predictions under fluctuating communication environments in a series of simulations and experiments. Further, we closely observe and discuss the effects of the dynamically changing structures on the bit-error rate performance. Note that although fading typically refers all possible channel distortion due to various communication disturbance, here we mainly focus on fast time-varying fading caused by rapid movement of mobile users and multipath propagation.

This article extends our preliminary proposal [30] through further tuning of the network update by introducing group-Lasso in addition to Lasso, careful characterization of the proposed methods in numerical and actual experiments with practical 5G wireless communication scenarios, and detailed derivation of the complex-valued steepest descent methods with the penalties. The major contributions of our study can be summarized as follows:

1) Proposal of complex-valued update schemes that self-adjust network structures to provide suitable network size by responding complicated channel states;
2) Design of new channel prediction methods based on dynamic ML-CVNNs with the proposed network structures and BPTS for an adaptive prediction;
3) Verification of the fact that the proposed fast fading prediction has a performance superior to other approaches on simulated and experimentally observed channel states.

This article is organized as follows: Section II briefly introduces the channel model theory and path separation in the frequency domain. After reviewing the conventional CVNN-based channel prediction in Section III, we propose novel prediction methods based on ML-CVNNs with the dynamically changing structures in Section IV. Then, Sections V and VI present its performance in simulations and experiments, respectively. Finally, Section VII provides the conclusion.

II. CHANNEL MODEL AND MULTIPATH SEPARATION IN FREQUENCY DOMAIN

Channel states of communications are distorted mainly by multipath interference caused by scattering in the communication environment. In addition, movement of mobile users and/or scatterers causes rapid and irregular channel changes in time. Fig. 1 shows an example of fading channel states observed in actual mobile communications. The curve demonstrates irregularity and nonlinearity of channel changes in the complex domain, and express difficulty of channel prediction because of its inconsistent rotation-like changes. Generally, a signal received at a communication end $y(t)$ at time t is modeled with time-varying channel $c(t)$ as

$$y(t) = c(t)\, s(t) \; + n(t) \tag{1}$$

where $s(t)$ and $n(t)$ are a transmitted signal and additive white Gaussian noise (AWGN), respectively. According to the Jakes model [1], [31], a fading channel $c(t)$ as a function of time t can be modeled as a summation of individual M complex signal paths $c_m(t)$ at a receiver and expressed as

$$c(t) = \sum_{m=1}^{M} c_m(t) = \sum_{m=1}^{M} a_m e^{j(2\pi f_m t + \phi_m)} \tag{2}$$

where a_m, f_m, and ϕ_m are amplitude, Doppler frequency and phase shift of each single path m, and M is the total path number. The Doppler frequency due to movement of a mobile user is given by

$$f_m = \frac{f_c}{c} v \cos \psi_m \tag{3}$$

where v and c are speed of the mobile user and the speed of light, respectively, f_c is a carrier frequency of the communication, and ψ_m is an incident radio-wave angle with respect to motion of a mobile user. Fig. 2 illustrates relationship of a base station, a mobile user, and scatterers in a multipath mobile communication that suffers from fading.

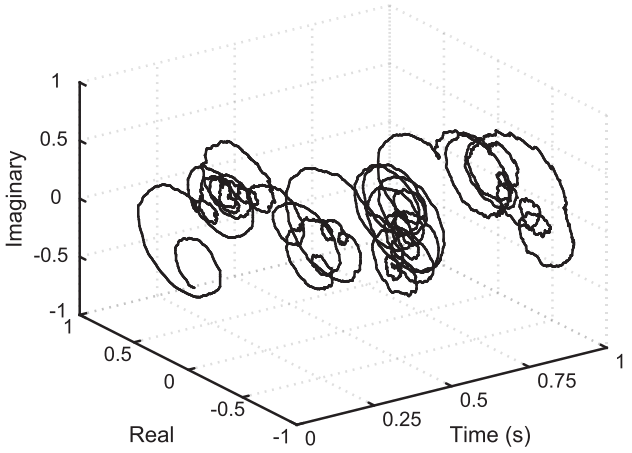


FIGURE 1. An example of time-varying fading channel states in the complex domain measured in an actual mobile communication.

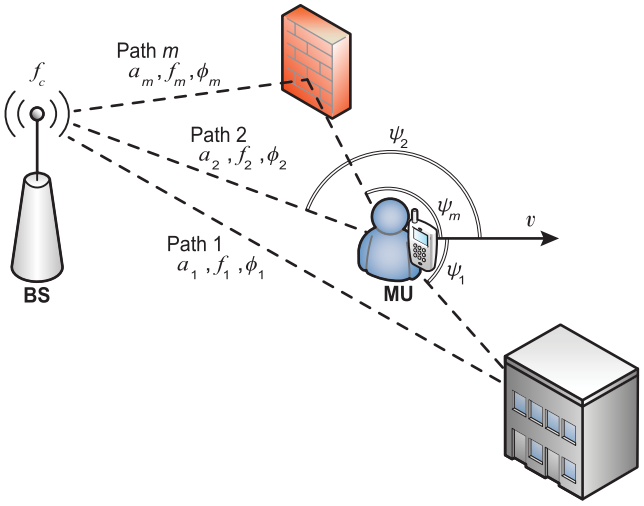


FIGURE 2. Jakes multipath model. Communication channel between a base station (BS) and a mobile user (MU) is distorted by interference of multipath and movement of communication ends and/or scatterers.

An observed channel $c(t)$ in an actual communication can be decomposed into multiple path components $c_m(t)$ in the frequency domain based on this model. Different path components with different incident angles ψ_m appear as separated peaks in a Doppler frequency spectrum. Hence, the parameters of each path component can be estimated by finding peak amplitudes and Doppler frequencies for a_m and f_m in the Doppler spectrum and the corresponding phase shifts for ϕ_m in its phase spectrum [13]. Chirp Z-transform (CZT) with a Hann window provides low calculation cost and a smooth frequency-domain interpolation that is useful for an accurate estimation of the parameters in the region close to zero frequency [32]. By sliding a Hann window on preceding channel states and by repeating the parameter estimation process, we can obtain separated path components at different time points. We focus on the fact that the separated channel states $c_m(t)$ have rotary locus in the complex plane and

predict its change in time for obtaining the future channel by using CVNNs.

III. CONVENTIONAL CVNN-BASED CHANNEL PREDICTION WITH A PRE-DEFINED NETWORK STRUCTURE

The changes in the separated channel components $c_m(t)$ can be predicted by ML-CVNNs [13]. CVNN is a framework suitable for treating signal rotation and scale in the complex plane by use of its high generalization ability [21], [33], [34]. It has been receiving more attention in various applications that intrinsically require dealing with complex values [35]–[38]. With a basic ML-CVNN consisting of a layer of I_{ML} input terminals, a hidden-neuron layer with K_{ML} neurons and an output neuron, we can predict the complex-valued $c_m(t)$ from a set of past channel components, $c_m(t-1), \ldots, c_m(t-I_{\mathrm{ML}})$ for each path $m = 1, \ldots, M$. The input terminals distribute input signals, $c_m(t-1), \ldots, c_m(t-I_{\mathrm{ML}})$, to the hidden-layer neurons as their inputs $\boldsymbol{z}_1$. In the same way, the outputs of the hidden-layer neurons $\boldsymbol{z}_2$ are passed to the output-layer neuron as its inputs. The neurons in the hidden layer are fully connected with the input terminals and the output-layer neuron. The output of the output-layer neuron z_3 is the prediction result $\tilde{c}_m(t)$. The connection weight w_{lkj} to kth output of jth neuron/input terminal in layer l is expressed by its amplitude $|w_{lkj}|$ and phase θ_{lkj}. The internal state $u_{(l+1)k}$ of kth neuron in $(l+1)$th layer is obtained as the summation of its inputs $\boldsymbol{z}_l$ weighted by $\boldsymbol{w}_{lk} = [w_{lkj}]$, i.e.,

$$u_{(l+1)k} = |u_{(l+1)k}|e^{i\theta_{(l+1)k}} \equiv \sum_j |w_{lkj}||z_{lj}|e^{i(\theta_{lkj}+\theta_{lj})} \tag{4}$$

where $z_{lj} = |z_{lj}|e^{j\theta_{lj}}$. The output $z_{(l+1)k}$ is given by adopting an amplitude-phase-type activation function f_{ap} to $u_{(l+1)k}$ as

$$z_{(l+1)k} \equiv f_{\mathrm{ap}}(u_{(l+1)k}) = \tanh(|u_{(l+1)k}|)e^{(i\theta_{(l+1)k})} \tag{5}$$

In our previous work, the connection weights $\boldsymbol{W}_l = [w_{lkj}]$ in the ML-CVNN were updated as follows. The ML-CVNN regarded a past known channel component $\hat{c}_m(t)$ as an output teacher signal, while its preceding channel components associated with the same path $\hat{c}_m(t-1), \ldots, \hat{c}_m(t-I_{\mathrm{ML}})$ were considered as input teacher signals. The weights have been updated based on the steepest descent method so that they minimize the difference

$$E_{(l+1)} \equiv \frac{1}{2}|\boldsymbol{z}_{(l+1)} - \hat{\boldsymbol{z}}_{(l+1)}|^2 \tag{6}$$

where $\boldsymbol{z}_{(l+1)}$ and $\hat{\boldsymbol{z}}_{(l+1)}$ denote temporary output signals and the teacher signals, respectively, in layer $(l+1)$. The teacher signals in the hidden layer $\hat{\boldsymbol{z}}_2$ were the signals obtained through the backpropagation of the teacher signal (BPTS) of the output layer $\hat{z}_3$ [21], [29], [39]. The weight updates were performed at each estimated channel components by sliding the teacher signal and the input set in the time domain. We have stopped the update at a certain small number of iteration R_{ML} in the update process for $\hat{c}_m(t)$ and kept the updated weights as the initial values in the following weight update for $\hat{c}_m(t+1)$ and prediction of future channels. With this procedure, we reduced the learning cost and followed the weak regularity of the separated channel components $c_m(t)$ for achieving a channel prediction with high accuracy. The network size, I_{ML} and K_{ML}, was empirically determined based on prediction performance in a series of simulated communication scenarios.

IV. PROPOSAL OF ONLINE SELF-OPTIMIZING CVNN

A number of previous studies have proposed different methods to finding optimized structures of neural networks in general [22], [40]–[42]. The so-called destructive neural networks start learning with a large structure, and then prune redundant connection weights and/or neurons to obtain an optimum network [43], [44], whereas the constructive neural networks raise the size from a small network to larger ones [45], [46].

In this article, we introduce regularization in the complex domain to achieve a dynamic CVNN that prunes and grows connections depending on fluctuating communication situations. Fig. 3 shows the construction of the CVNN. We want a CVNN that changes its connections according to prediction situations and dynamically keeps suitable network structures in a series of predictions without manual tuning. To realize such a network, we introduce a constraint for sparsity to the weight updates in order to restrict the connections of networks to a suitably small size. The L_0-norm is an exact sparsity measure, and our problem can be redefined as minimizing the error function of the weights (6) with the L_0-norm constraint on the connection weights. However, this problem has been shown to be NP-hard in general. Fortunately, under some conditions, the L_1-norm can serve as a sparsity measure for substituting the L_0-norm [23], [47], [48]. The L_1-norm of the weights is a practical sparsity measure since it is convex so that we can perform optimization more easily [24], [49], [50]. By introducing the sparse constraint as a penalty function in

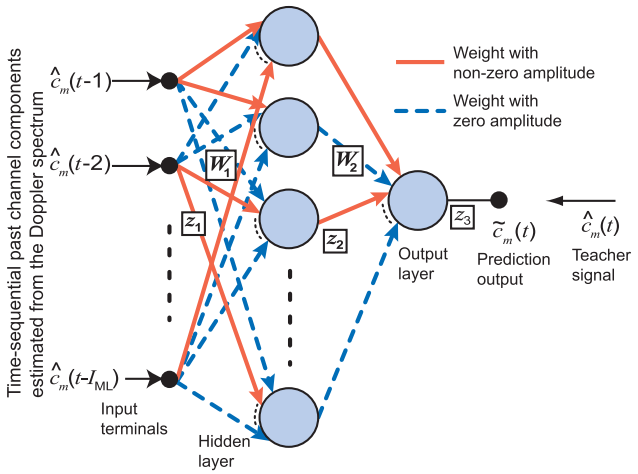


FIGURE 3. Construction of a sparse complex-valued neural network, in which solid red arrows show non-zero-amplitude connections while dashed blue arrows represent zero-amplitude ones. Modified from [30].

a ML-CVNN, the objective function we use to update the weights in layer l is expressed as

$$\arg\min_{\boldsymbol{W}_l} E^{\mathrm{S}}_{(l+1)} = \arg\min_{\boldsymbol{W}_l}\left(\frac{1}{2}|\boldsymbol{z}_{(l+1)} - \hat{\boldsymbol{z}}_{(l+1)}|^2 + \alpha\|\boldsymbol{W}_l\|_1\right) \tag{7}$$

where α is a coefficient to express degrees of the penalty. Minimizing the second term of the right-hand side of (7) means restricting the non-zero weight number to get its minimal number in the network. This is effectively equivalent to pruning connection weights. In other words, the penalty function introduces sparsity to the weight updates so that the remaining weights form an effective structure for representing the output signals. We use the steepest descent method in the complex domain to update the weights here (see Appendix A for derivation). Thus, the weight amplitude $|w_{lkj}|$ and the phase θ_{lkj} are renewed as

$$\begin{aligned}
&|w_{lkj}|(r+1)\\
&= |w_{lkj}|(r) - \kappa_1 \frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial(|w_{lkj}|)}\\
&= |w_{lkj}|(r) - \kappa_1\Big\{(1-|z_{(l+1)k}|^2)\\
&\quad\times\big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\big)|z_{lj}|\cos\theta^{\mathrm{rot}}_{lkj}\\
&\quad - |z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\frac{|z_{lj}|}{|u_{(l+1)k}|}\sin\theta^{\mathrm{rot}}_{lkj}\\
&\quad + \alpha\Big\}
\end{aligned} \tag{8}$$

$$\begin{aligned}
&\theta_{lkj}(r+1)\\
&= \theta_{lkj}(r) - \kappa_2 \frac{1}{|w_{lkj}|}\frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial\theta_{lkj}}\\
&= \theta_{lkj}(r) - \kappa_2\Big\{(1-|z_{(l+1)k}|^2)\\
&\quad\times\big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\big)|z_{lj}|\sin\theta^{\mathrm{rot}}_{lkj}\\
&\quad + |z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\frac{|z_{lj}|}{|u_{(l+1)k}|}\cos\theta^{\mathrm{rot}}_{lkj}\Big\}
\end{aligned} \tag{9}$$

where $\theta^{\mathrm{rot}}_{lkj} \equiv \theta_{(l+1)k} - \theta_{lj} - \theta_{lkj}$, r is an index of learning iteration, and κ_1 and κ_2 are learning constants. This update rule has an additional term $+\alpha$ in the amplitude $|w_{lkj}|$ update in comparison to the conventional complex-valued steepest descent method [13], [21] because of the penalty term.

In addition to the L_1-norm regularization, we also introduce group sparse term [27], [51], [52] as a penalty of the weight updates in this article. While L_1-norm imposes sparsity on weight connections by considering each weight as a single unit, the group sparse penalty introduced by $L_{2,1}$-norm regularization imposes sparsity on input terminals and neurons in a network by considering all outgoing wights from an input terminal or a neuron as a single group. We prune input terminals and/or neurons by redefining the objective function as

$$\begin{aligned}
\arg\min_{\boldsymbol{W}_l} E^{\mathrm{GS}}_{(l+1)} &= \arg\min_{\boldsymbol{W}_l}\left(\frac{1}{2}|\boldsymbol{z}_{(l+1)} - \hat{\boldsymbol{z}}_{(l+1)}|^2 + \alpha\|\boldsymbol{W}_l\|_{2,1}\right)\\
&= \arg\min_{\boldsymbol{W}_l}\Big(\frac{1}{2}|\boldsymbol{z}_{(l+1)} - \hat{\boldsymbol{z}}_{(l+1)}|^2\\
&\quad + \alpha\sum_j \sqrt{|\boldsymbol{w}_{lj}|}\sqrt{\sum_k |w_{lkj}|^2}\Big)
\end{aligned} \tag{10}$$

where $\boldsymbol{w}_{lj} = [w_{lkj}]$ and $|\boldsymbol{w}_{lj}|$ denotes the dimensionality of the vector $\boldsymbol{w}_{lj}$. Note that, in this work, $\sqrt{|\boldsymbol{w}_{lj}|}$ can be moved outside the summation of j for consisting isotropic $L_{2,1}$-norm due to the same dimensionality of weights in a layer resulting from the fully connected structure. Minimizing the $L_{2,1}$-norm term means restricting the input terminal/neuron numbers by setting all weight connections from a terminal or a neuron to be either simultaneously zeros or none of them are. We use the same steepest descent method to update the weights based on this objective (see Appendix A for derivation) and get a new update rule for the complex-valued group sparsity as

$$\begin{aligned}
&|w_{lkj}|(r+1)\\
&= |w_{lkj}|(r) - \kappa_1 \frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial(|w_{lkj}|)}\\
&= |w_{lkj}|(r) - \kappa_1\Big\{(1-|z_{(l+1)k}|^2)\\
&\quad\times\big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\big)|z_{lj}|\cos\theta^{\mathrm{rot}}_{lkj}\\
&\quad - |z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\frac{|z_{lj}|}{|u_{(l+1)k}|}\sin\theta^{\mathrm{rot}}_{lkj}\\
&\quad + \alpha\sqrt{|\boldsymbol{w}_{lj}|}\frac{|w_{lkj}|}{\|\boldsymbol{w}_{lj}\|_2}\Big\}
\end{aligned} \tag{11}$$

$$\begin{aligned}
&\theta_{lkj}(r+1)\\
&= \theta_{lkj}(r) - \kappa_2 \frac{1}{|w_{lkj}|}\frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial\theta_{lkj}}\\
&= \theta_{lkj}(r) - \kappa_2\Big\{(1-|z_{(l+1)k}|^2)\\
&\quad\times\big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\big)|z_{lj}|\sin\theta^{\mathrm{rot}}_{lkj}\\
&\quad + |z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\frac{|z_{lj}|}{|u_{(l+1)k}|}\cos\theta^{\mathrm{rot}}_{lkj}\Big\}
\end{aligned} \tag{12}$$

This update rule has an additional term $+\alpha\sqrt{|\boldsymbol{w}_{lj}|}\frac{|w_{lkj}|}{\|\boldsymbol{w}_{lj}\|_2}$ in the amplitude $|w_{lkj}|$ update in comparison to the conventional complex-valued steepest descent method due to the $L_{2,1}$-norm penalty.

For simplicity and lower computational consumption, BPTS is used in the both update schemes for getting the teacher signal $\hat{z}_2$ in the hidden layer from the teacher signal in the output layer $\hat{z}_3$ as

$$\hat{z}_2 = \big(f_{\mathrm{ap}}(\hat{z}^*_3\boldsymbol{W}_2)\big)^* \tag{13}$$

where $(\cdot)^*$ represents the complex conjugate or hermite conjugate.

To predict fading channels, we update the connection weights by time-sliding the inputs and output teacher signals on the estimated $\hat{c}_m(t)$ sequences as we performed in the previous work [13]. That is, a set of updated weights using the complex-valued estimation $\hat{c}_m(t)$ as the output teacher signal

for $\tilde{c}_m(t)$ in Fig. 3 and $\hat{c}_m(t-1), \ldots, \hat{c}_m(t-I_{\mathrm{ML}})$ as the input signals are kept in the network and used as the initial weights in the following update for $\tilde{c}_m(t+1)$ by regarding $\hat{c}_m(t+1)$ as the new output teacher signal and $\hat{c}_m(t), \ldots, \hat{c}_m(t-I_{\mathrm{ML}}+1)$ as the new input signals. The weight update is performed until the latest channel component is used and the most up-to-dated weight connections predict the future channel states. The combinations of the penalty terms and the prediction scheme in the time domain are expected to keep the structure to be a suitable size for the channel prediction depending on the fluctuating communication environment.

The computation complexity of the online training part is $O(M \cdot R_{\mathrm{ML}} \cdot K_{\mathrm{ML}} \cdot I_{\mathrm{ML}})$ where M, R_{ML}, K_{ML}, and I_{ML} are the detected path number, the weight update iterations, the hidden-neuron number and the input terminal number, respectively. This complexity is equivalent to that of the conventional ML-CVNN updates [13] and typically much smaller than the complexity of CZT calculation, $O(N\log N)$ where N is the symbol number in a CZT window. The additional penalty functions based on L_1-norm and $L_{2,1}$-norm realize the per-weight and per-neuron sparsity within a CVNN structure without increase of calculation complexity for the weight updates.

V. NUMERICAL EXPERIMENTS

In the following two sections, we evaluate the performance of the channel prediction methods based on the ML-CVNNs with the penalties in simulations and experiments. We assume orthogonal frequency-division multiplexing (OFDM) with quadrature phase shift keying (QPSK) modulation and time division duplex (TDD) as the communication scheme in this article. For the future compatibility with 5G communications, cyclic-prefix (CP) OFDM with the system parameters listed on Table 1 are used in this work.

TABLE 1. Communication parameters.

Parameter	Value
QPSK symbol number	12852
Number of OFDM subcarriers	2048
Number of OFDM guard bands	106 left, 106 right
Number of OFDM symbols	7
Length of OFDM cyclic prefix	[160, 144×6]
TDD sub-frame length	0.512 ms
TDD symbol number in a sub-frame	15360
TDD frame length	10 sub-frames
Sampling rate	30 MHz

In this section, we characterize the performance of the proposed channel prediction methods with various degree of penalty α by using simulated fading channels. The geometrical setup of the simulation is shown in Fig. 4. We consider communications between a base station (BS) and a mobile user (MU) moving away from the BS at v m/s with a certain moving angle. There are two scatterers making 2 paths in addition to the line-of-sight path. The carrier frequency is 2 GHz here.

We predict channel changes in a TDD frame based on its preceding channel states. The past path characteristics are estimated by using CZT with a Hann window. A window with 8-TDD-frame length is applied to the past channel states for estimating the path parameters, $a_m(t)$, $f_m(t)$, $\phi_m(t)$, based on peaks in Doppler spectra and corresponding phase spectra. Then, the past path characteristics $c_m(t)$ are composed by using the parameters and assigned as the estimated characteristics at the center of the window. We shift the window center at a TDD-frame, i.e., 10 sub-frames, interval for estimating multipath characteristics at every TDD frame. The details of the time frames are explained in our previous work [13].

To evaluate the performance in various channel changes, we changed the scatterer distance Δx shown in Fig. 4 from $\Delta x = 0.5$ to 20 m with 0.5 m steps, and performed 100 independent predictions at different time points along the movement of MU at a speed of 12 m/s in each scatterer arrangement (4,000 predictions in total). We started with the neural network with the parameters listed in Table 2. The penalties prune and grow the network connections, 30×30 in the hidden layer and 30×1 in the output layer, in a timely manner as the communication situation changes.

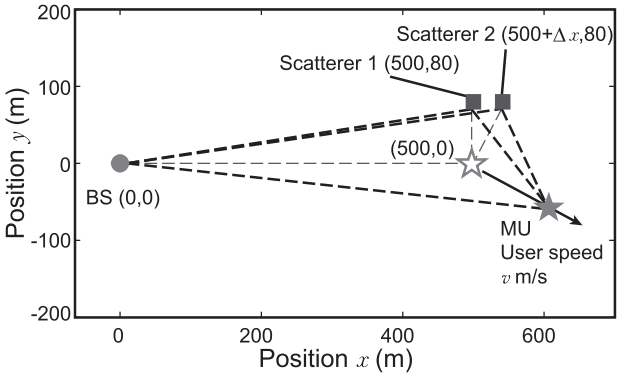


FIGURE 4. Geometrical setup used in the simulation. There are two scatterers separated by Δx m, a base station (BS) and a mobile user (MU) in an open communication space. The line of sight between the BS and the MU is considered. The MU moves in the direction of the arrow ($-30°$ from the x axis) with a velocity of v m/s. Modified from [30].

TABLE 2. Channel prediction parameters.

Parameter	Value
Channel estimation	
CZT size	8 TDD frames
Down-sampled signal rate for CZT	500 kHz
ML-CVNN channel prediction	
Input terminals I_{ML}	30
Hidden-neuron number K_{ML}	30
Weight update iterations per prediction R_{ML}	10
Optimizer	CV steepest descent

Figs. 5(a) and 6(a) show the mean of the network size versus scatterer distance for the methods with the L_1-norm and the $L_{2,1}$-norm penalties, respectively. A connection weight is counted as non-zero here if its amplitude satisfies

$$|w_{lkj}| \geq \max(|\boldsymbol{W}_l|)/100 \tag{14}$$

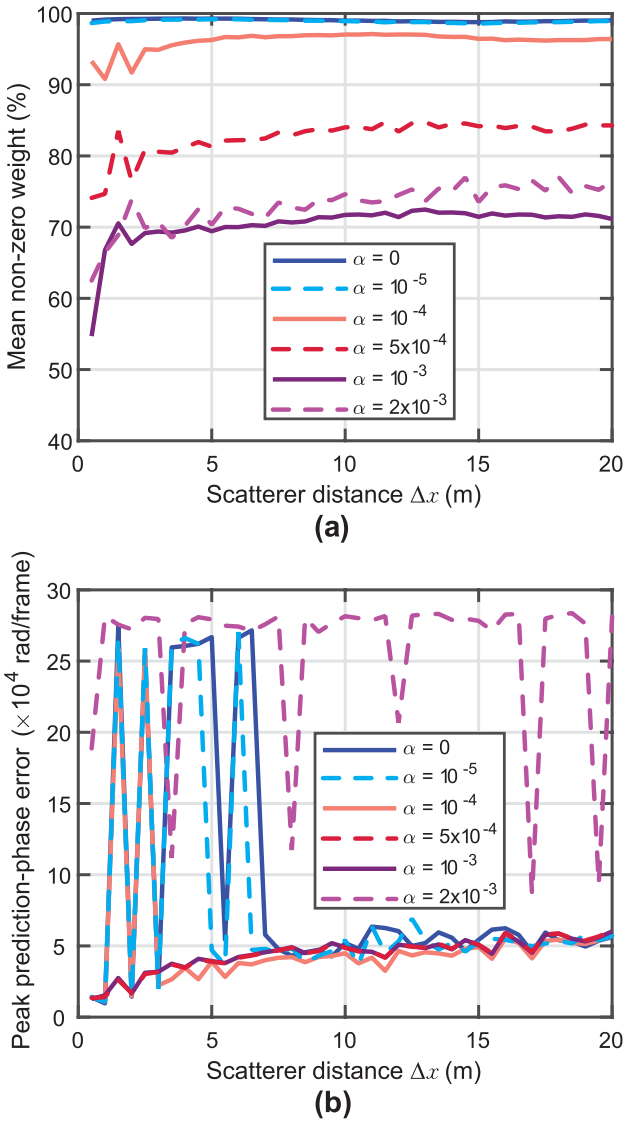


FIGURE 5. Simulated prediction results of the L_1-norm penalized ML-CVNN prediction method with various penalty coefficients α. (a) Averaged non-zero weight ratios (network size) and (b) maximum predicted phase errors (prediction stability) against scatterer distance Δx in Fig. 4 (communication situations).

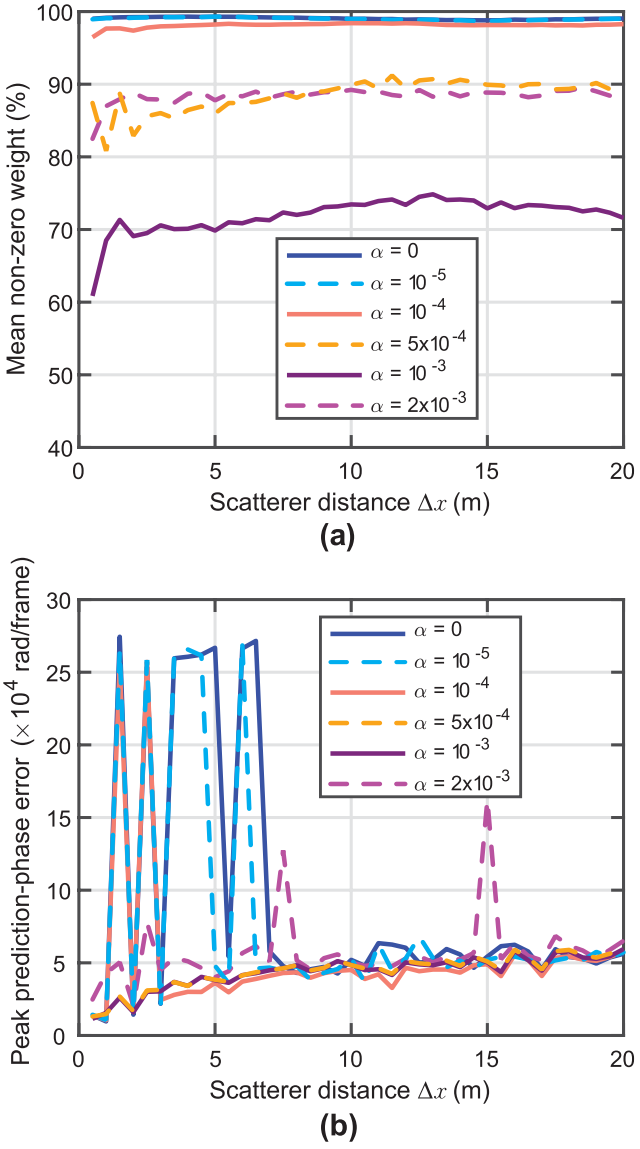


FIGURE 6. Simulated prediction results of the $L_{2,1}$-norm penalized ML-CVNN prediction method with various penalty coefficients α. (a) Averaged non-zero weight ratios (network size) and (b) maximum predicted phase errors (prediction stability) against scatterer distance Δx in Fig. 4 (communication situations).

Otherwise, the weight is considered as a zero weight. If a weight in the output layer ($l = 2$) is counted as a zero weight, all the weights in the hidden layer connecting themselves to the corresponding neuron are also considered as zero weights in order to fairly compare the penalty effect in the entire network. The network sizes of the ML-CVNN with various penalty coefficients ($\alpha = 0, 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-2}$) have been evaluated, and the mean connection numbers for the 100 trials in each condition have been normalized by the maximum possible connections to show the non-zero connection ratio. Corresponding prediction accuracy is also calculated by accumulating predicted phase errors within the prediction frame. Figs. 5(b) and 6(b) present the maximum estimated phase errors in each communication condition out of the 100 predictions, showing stability of the predictions with the L_1-norm and the $L_{2,1}$-norm penalties, respectively.

We found in Figs. 5(a) and 6(a) that the non-zero weight number consisting an effective network decreases as the penalty coefficient α increases as expected, whereas a network without the penalties ($\alpha = 0$) keeps almost all of the connections active regardless of the change of the communication environment. In Fig. 5(b) and 6(b), the smaller networks achieved by the penalties show better prediction stability compared to the conventional ML-CVNN-based method without any constraint ($\alpha = 0$). The results also present that the proposed prediction methods reach its best performance with a penalty coefficient around $\alpha = 5 \times 10^{-4} \sim 10^{-3}$ and

α larger than this value introduces instability to the channel prediction again. Note that, the L_1-norm penalty tends to prune more connection weights within the networks than the $L_{2,1}$-norm does for the penalty coefficient $\alpha \leq 10^{-3}$ and for almost all of the scatterer arrangements we evaluated. Interestingly, Figs. 5 and 6 also depict that the prediction stability of different coefficients, i.e., $\alpha = 5 \times 10^{-4}$ and $\alpha = 10^{-3}$, only show little difference for both the regularization methods even though the non-zero weight ratios of those are different from each other. This fact demonstrates that the differently connected networks generated by the L_1-norm and $L_{2,1}$-norm penalized CVNNs provide comparable prediction performance although the internal realizations are different.

Next, we have conducted additional simulations for studying the update speed of the methods to follow time-varying channels by evaluating their prediction performance (prediction-phase errors) and network sizes (non-zero weight ratios) against various mobile user speeds (Figs. 7 and 8) and update iterations (Fig. 9). For the user speed comparison, the movement speed v m/s of MU in Fig. 4 was changed from $v = 6$ to 20 m/s with 2 m/s steps. At each speed, the scatterer distance Δx was varied from 0.5 to 20 m and 100 independent predictions per scatterer distance were performed similarly as the first evaluation for mimicking various channel prediction situations. In the same manner, the prediction performance and the network sizes were compared against the update iterations per prediction $R_{\mathrm{ML}} = 1$, 10, 20, 30, 40, and 50 with a fixed MU speed $v = 12$ m/s.

The proposed online methods with an appropriate penalty coefficient, e.g., $\alpha = 5 \times 10^{-4}$, gradually increase their network sizes as increase of user speed for accommodating more complicated channel variations introduced by the fast movement of the communication terminal (Figs. 7(a) and 8(a)). The prediction stability of the methods with this coefficient also show superior performance for the mobile speed $v \leq 14$ m/s whereas the conventional method ($\alpha = 0$) introduces instability even for slower movement (Figs. 7(b) and 8(b)). Note that the current prediction schemes show limited prediction accuracy for the mobile speed larger than 16 m/s even with the aforementioned penalty coefficient. This instability, however, could be mitigated by updating the online networks more frequently, e.g., with 5 TDD sub-frame steps instead of the 10 sub-frame steps, for following the fast changes of channel states. On the other hand, the evaluation against update iterations also depicts an interesting property of the online method. As shown in Fig. 9, the conventional method and the methods with weak penalties exhibit clear deterioration on the prediction performance as increase of update iterations per prediction R_{ML}. This is a signature of overfitting of the networks to the preceding channel states. The proposed methods with an appropriate penalty $\alpha = 5 \times 10^{-4}$, in contrast, shows low prediction errors by pruning redundant connections within networks. This result also verifies that the update iteration $R_{ML} = 10$ used in this work is enough to provide stable prediction.

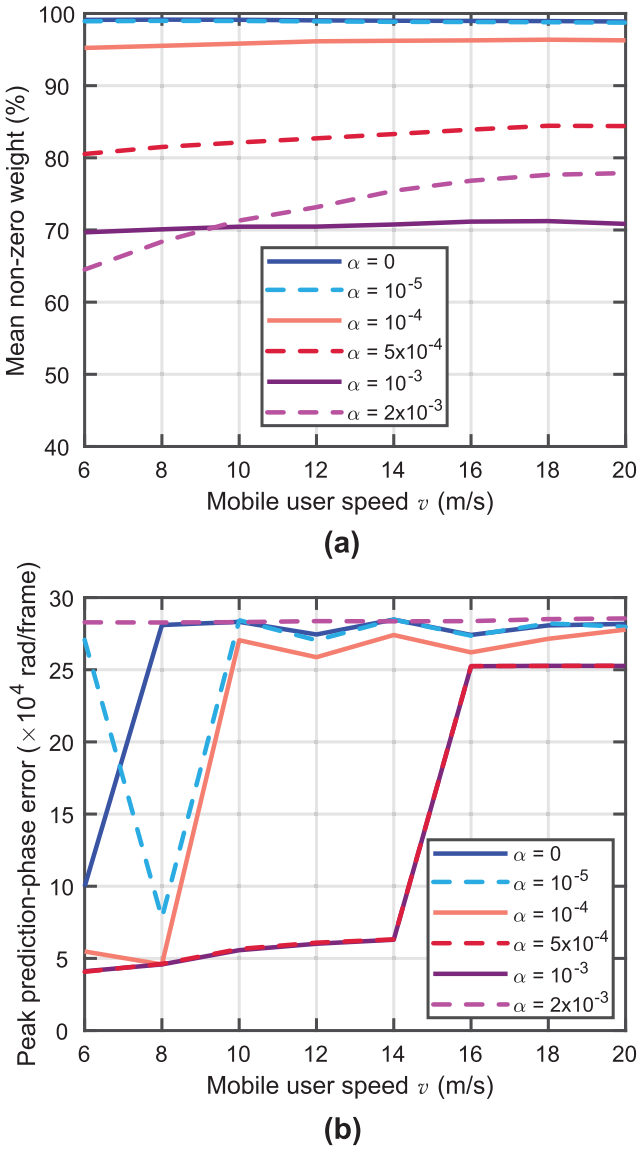


FIGURE 7. Simulated prediction results of the L_1-norm penalized ML-CVNN prediction method with various penalty coefficients α. (a) Averaged non-zero weight ratios (network size) and (b) maximum predicted phase errors (prediction stability) against mobile speed v m/s in Fig. 4 (communication situations).

Finally, we evaluated the impact of initial network configurations to the prediction performance and the final network sizes (Table 3). In this numerical experiment, the CVNN prediction methods with three different hidden-neuron numbers, namely $K_{\mathrm{ML}} = 5$, 30 and 60, were compared by setting the MU speed $v = 12$ m/s and the update iterations per prediction $R_{\mathrm{ML}} = 10$. The same input-terminal number, that is $I_{\mathrm{ML}} = 30$, was used for all conditions in order to exclude effect of different inputs and 4,000 predictions were conducted for each initial structural configuration. The CVNNs penalized by L_1-norm and $L_{2,1}$-norm reduced their effective network sizes (non-zero weight ratios) as the increase of the initial sizes as expected, whereas the

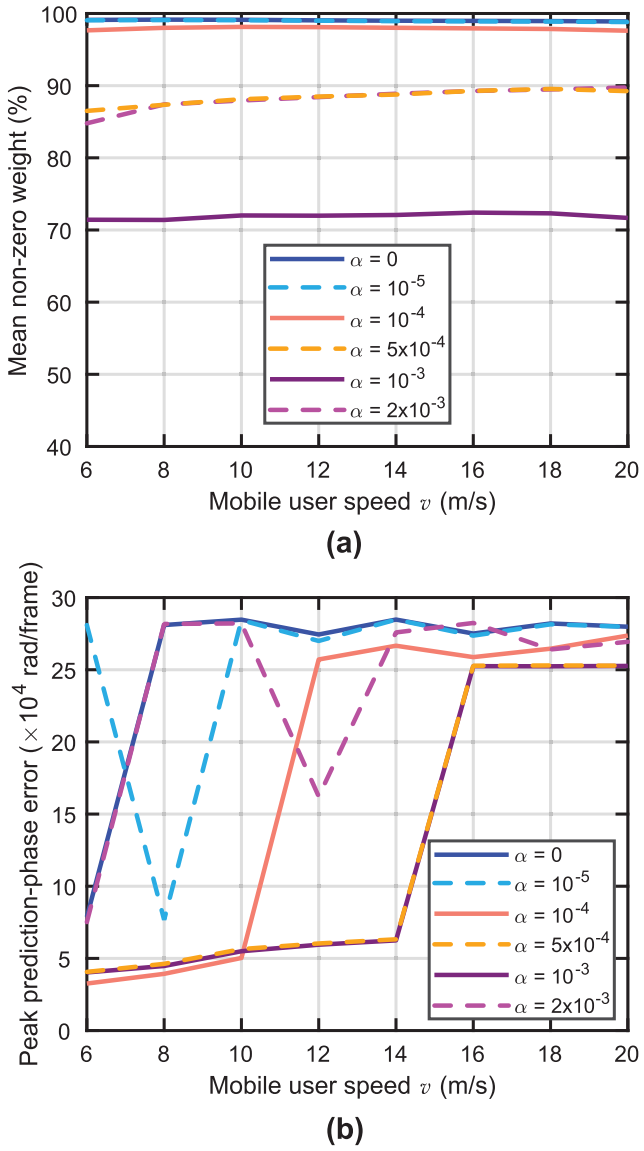


FIGURE 8. Simulated prediction results of the $L_{2,1}$-norm penalized ML-CVNN prediction method with various penalty coefficients α. (a) Averaged non-zero weight ratios (network size) and (b) maximum predicted phase errors (prediction stability) against mobile speed v m/s in Fig. 4 (communication situations).

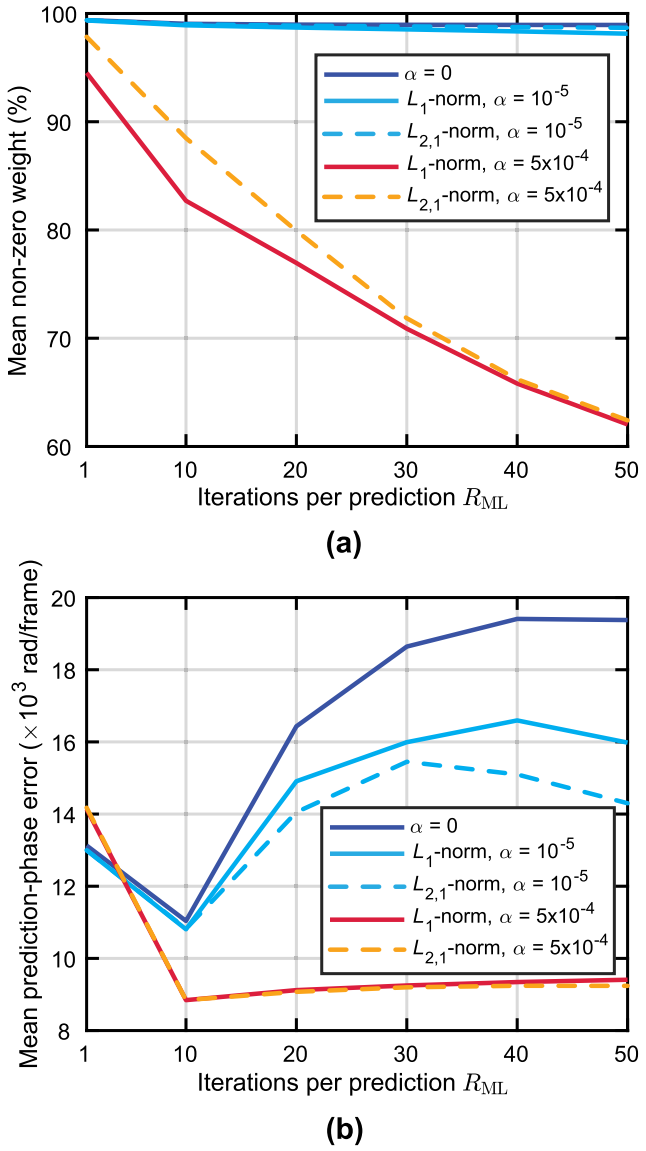


FIGURE 9. Simulated prediction results of the L_1-norm and $L_{2,1}$-norm penalized ML-CVNN prediction methods with various update iterations per prediction R_{ML}. (a) Averaged non-zero weight ratios (network size) and (b) averaged predicted phase errors (prediction performance) against update iterations R_{ML}.

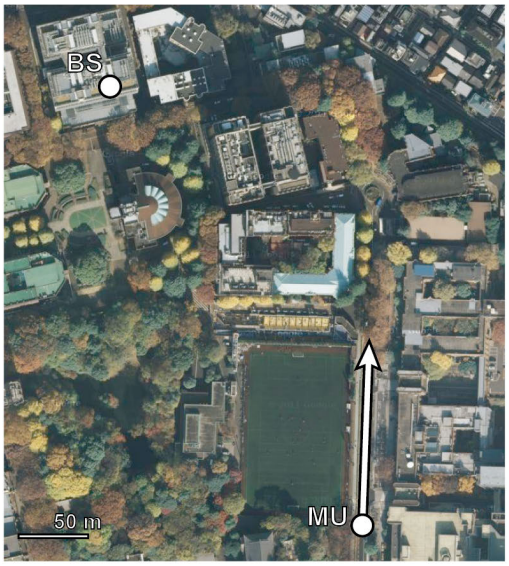
conventional network with $\alpha = 0$ kept almost all weight connections active regardless of the initial configurations. Table 3 also depicts that the proposed penalized methods with a larger hidden layer ($K_{\mathrm{ML}} = 30$ or 60) can slightly improve the prediction performance (prediction-phase error) compared to those with a restricted hidden layer $K_{\mathrm{ML}} = 5$, most likely due to a larger degree of freedom for realizing different internal weight connections. Together, these results demonstrate that the proposed prediction methods with an appropriate α can automatically prune redundant connections in its network to achieve higher prediction accuracy even in prediction conditions that are difficult for the conventional method.

VI. EXPERIMENTS IN ACTUAL COMMUNICATION ENVIRONMENT

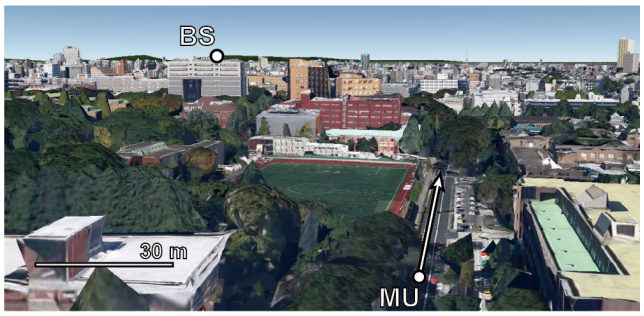
In this section, we further demonstrate adaptability of the proposed methods in prediction with actually observed fading channels. We experimentally observed fading channels in a communication situation shown in Fig. 10. There are an MU as a transmitter and a BS as a receiver in the experimental site with a lot of obstacles, such as buildings, trees and other automobiles, providing a typical mobile communication environment in an urban area. The MU moves in the direction of the arrows shown in Figs. 10(a) and (b) with a velocity around 12.5 m/s and transmits 1.297 GHz non-modulated waves from a monopole antenna, whereas the BS receives the wave

TABLE 3. Mean prediction-phase error (prediction performance) and non-zero weight ratios (network size) of the L_1-norm and $L_{2,1}$-norm penalized ML-CVNN prediction methods with various initial network configurations. The prediction methods were evaluated with three different hidden-neuron numbers ($K_{\mathrm{ML}} = 5$, 30 and 60), whereas the same input terminals ($I_{\mathrm{ML}} = 30$) were used in this evaluation in order to exclude effect of different inputs in the networks.

Measure	Method	$K_{\mathrm{ML}} = 5$	$K_{\mathrm{ML}} = 30$	$K_{\mathrm{ML}} = 60$
Mean prediction-phase error ($\times 10^4$ rad/frame)	$\alpha = 0$	1.1735	1.1038	1.0633
	L_1-norm, $\alpha = 5 \times 10^{-4}$	0.9071	0.8844	0.8846
	$L_{2,1}$-norm, $\alpha = 5 \times 10^{-4}$	0.9135	0.8846	0.8795
Mean non-zero weight (%)	$\alpha = 0$	99.00	99.05	98.87
	L_1-norm, $\alpha = 5 \times 10^{-4}$	94.97	82.70	67.74
	$L_{2,1}$-norm, $\alpha = 5 \times 10^{-4}$	96.65	88.49	71.05



(a)



(b)

FIGURE 10. Geometrical setup of the experiment illustrated as (a) two-dimensional top view (Google Maps, modified) and (b) three-dimensional side view (Google Earth, modified) which includes a fixed base station (BS), a moving mobile user (MU) and other obstacles.

by using another monopole antenna. Note that, depending on communication instants, a line-of-sight path may or may not present due to obstacles. The received channel signal was mixed with a 1.287 GHz local oscillator wave after an amplifier, and then extracted as a signal at an intermediate frequency of 10 MHz. After passing it to another amplifier and a band-pass filter with 2 MHz bandwidth, we sample the channel information at 30 M Sample/s. The channel was further down-sampled to 500 kHz for reduction of computational requirements in the CZT estimation. An observed channel change has already been shown in Fig. 1 as an example of the fading captured in this communication situation. It is time-sequential data showing irregular rotation of the channel in the complex plane received at the BS. The channel state gives roughly 2 distinct main paths over the communication period. The channel changes in a TDD frame are predicted based on preceding channel states by using CZT and ML-CVNNs in the same way as in Section V.

First, we evaluate the time variation of the ML-CVNN size to demonstrate the online dynamics. Fig. 11 shows sequential changes of a fading channel and the neural-network size in a prediction process. The actually received signal power, phase values and change rates of the phase are shown in Figs. 11(a-c), respectively. Non-zero connection weights are counted by using the same scheme as (14) in Section V, and plotted against time in Fig. 11(d). The non-zero weight ratios shown in this plot represent network sizes after updating CVNNs using the channel states up to the same time points in Fig. 11(a,b). In order to demonstrate the impact of the penalty functions on the network size change, $\alpha = 5 \times 10^{-4}$ has been used based on the discussion in the previous section. For a comparison, the operation without any penalty constraint is also characterized as the conventional method.

Fig. 11(c) shows that the channel does not always change in the same manner but there are sporadic fast changes among relatively stable states. The fast changes cause difficulty in channel prediction and degrade its performance. In Fig. 11(d), we observed large network changes at the beginning part of the network update (up to 0.4 s). We attribute these changes to self-optimization of the networks with the penalties for adapting themselves to follow the basic trends of channel changes. Since our methods do not require any pre-training, we always observe similar dynamically changing periods right after starting the network update. However, once the network has experienced enough channel changes, the proposed methods tend to keep relatively small structures and increase their effective network sizes only in response to serious fading. On the other hand, the conventional method without any penalty, namely CVNN, does not change its network size in any part of the update procedure, and no correlation with the channel changes is observed.

For further discussion, we focus on a prediction period containing three fast phase jumps. Figs. 11(e-g) present the

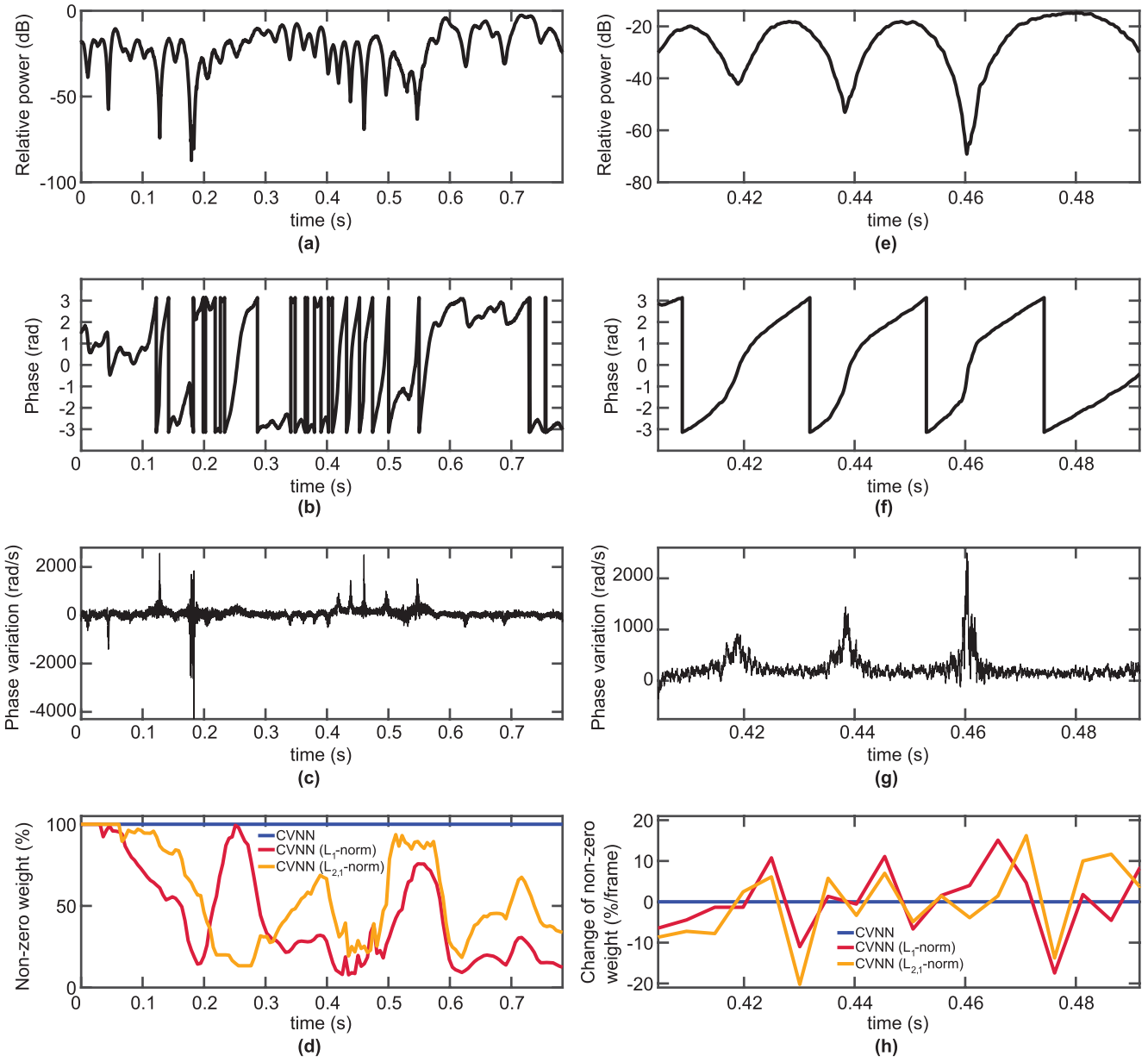


FIGURE 11. Actual propagation-experiment results showing (a) observed relative power, (b) observed phase value, (c) change rate of the phase in every 2×10^{-6} ms, and (d) non-zero weight ratios of CVNNs with L_1-norm (red) and $L_{2,1}$-norm (orange) penalties as well as without any penalty (blue). (e-g) zoomed-in version of a-c from 0.405 to 0.492 ms. (h) change rates of non-zero weight ratios of the CVNNs in this period. $\alpha = 5 \times 10^{-4}$ is assigned as the penalty degree for the proposed methods.

channel power, phase and phase changes during the period from 405 ms to 492 ms in Figs. 11(a-c). Fig. 11(h) shows the change rates of the non-zero weight ratio in the period calculated by taking the difference of the weight ratios at two consecutive TDD frames. It is obvious that the CVNNs with the regularization increase their non-zero weight connections during their update process synchronously with the large channel changes in order to adapt the network structures to those difficult prediction parts and decrease the connections after them. In contrast, the conventional method is not sensitive to the channel changes. These results demonstrate that the CVNNs with the penalty functions accommodates themselves to such large and irregular channel changes by increasing their weight connections while they reduce the connections when the channel changes steadily. In other words, the proposed method has the ability to change the network structure dynamically and adaptively online according to the degree of difficulty in the channel prediction. Note that we observed a lag between the actual channel changes (Fig. 11(g)) and the structure changes (Fig. 11(h)). This lag, however, could be mitigated by updating the online networks more frequently, e.g., with

5 TDD sub-frame steps or less, at the expense of calculation costs.

Finally, we compare prediction accuracy in various channel prediction methods in the actual communications. In this test, respective methods predict fading channel states in each TDD frame using channel information prior to the prediction periods in the same way as above. The predicted channel states are used for compensating the true fading in a communication situation with the CP-OFDM system described is Section V. A randomly generated binary sequence has been converted into QPSK symbols and modulated into transmission signals based on the parameters shown in Table 1. The signals are assumed to be transmitted through a communication environment with the observed fading channel and different levels of additive white Gaussian noise. Before demodulation of OFDM and QPSK, the received signals are compensated by the predicted channel states using multiple prediction methods, namely, methods based on a linear prediction directly in the time domain, an AR model using channel characteristics estimated by CZT, a LSTM network [53], [54] (see Appendix B), the conventional CVNN, and the proposed CVNN with the L_1-norm and $L_{2,1}$-norm penalties. We independently performed this process 101 times on different periods of the observed fading channels. For demonstrating the performance of the proposed methods, we also evaluate the conventional CVNN-based method with a smaller network size, namely CVNN (small), consisting of the same input terminals $I_{\mathrm{ML}} = 30$ but with smaller hidden-neuron number $K_{\mathrm{ML}} = 5$ in addition to the network structure listed in Table 2.

Fig. 12 shows the bit-error-rate (BER) curves against bit-energy to noise-power-density ratio E_b/N_0. Here, CVNN (large) shows the result of the conventional CVNN-based method with the network structure listed in Table 2, whereas CVNN (L_1-norm) and CVNN ($L_{2,1}$-norm) present that of the proposed methods with the same structure size and a penalty coefficient $\alpha = 5 \times 10^{-4}$. Communication without any prediction method, that is channel compensation using channel states in the most recent TDD frame, is also performed for a comparison (No prediction). Communication bit errors with respective prediction methods are obtained in each process and accumulated over all iterations for the final BER calculation. The black solid curve in the plot represents BER if the true future channel is perfectly known, thus showing the lower bound of the BER with the considered CP-OFDM setup. We can see the difficulty of the channel prediction on the actual fading in the large deviation of the BER curves corresponding to the linear, AR-model-based, LSTM and the conventional CVNN-based (for both large and small) prediction methods. The larger error rates by some of the methods, that is linear, AR model and LSTM, compared to the no-prediction method implies that the observed rapid and irregular changes of the channel states cause failure of channel compensation by those methods. Note that the conventional CVNN-based methods with the large and small network configurations show roughly the same error-rate performance in this evaluation. This is because these fixed network structures exhibit high prediction performance in some prediction cases but provide poor results in the other situations over the 101 individual trials. In contrast, the proposed methods with the regularization, CVNN (L_1-norm) and CVNN ($L_{2,1}$-norm), achieve an accurate prediction even in the difficult communication situations. Those methods exhibit 2.5 dB and 5.5 dB improvement of BER at 10^{-3} and 5×10^{-4}, and allow a mobile communication with 10^{-5} BER at $E_b/N_0 = 23 - 24$ dB. While other channel prediction methods have error floors in the BER evaluation and cannot provide more precise communications than 10^{-4} BER, the proposed CVNN methods show low BER without any error floors up to 10^{-5} BER. The results show that the proposed online adaptive CVNNs with the regularization provides higher channel prediction performance. The clear improvement of the proposed methods compared to CVNN (large) and CVNN (small) shows the benefit of the online regularization methods over the stationary networks in the channel prediction.

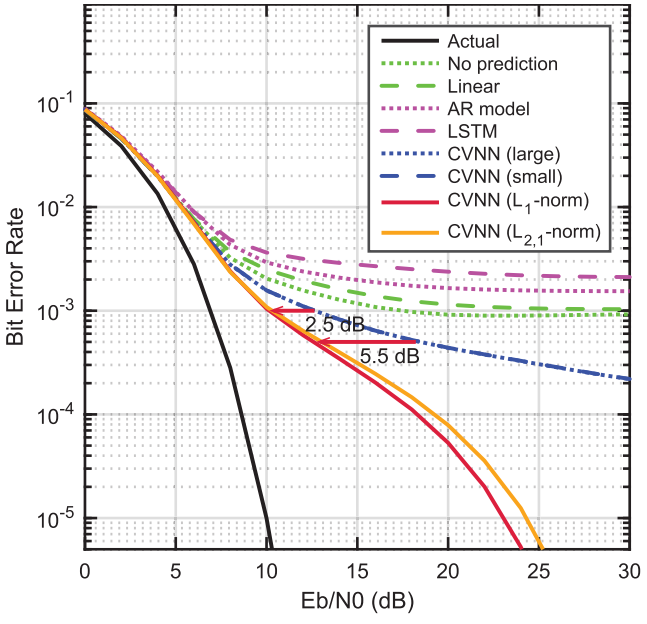


FIGURE 12. BER curves obtained for different channel prediction methods in communications with fading channel measured in actual environment. Red arrows indicate improvement of BER at 10^{-3} and 5×10^{-4} by the proposed methods.

VII. CONCLUSION

In this article, we proposed online adaptive channel prediction methods based on ML-CVNNs with self-optimizing dynamic structures. One of the main challenges in channel prediction with fast fading is its randomly time-varying channel states. For example, even a slight environment change due to movement of mobile users and/or obstacles during signal transmissions may significantly alter received channel states at communication ends. Hence, it is extremely difficult to construct a universal learning model covering such wide ranges of changing communication scenarios for pre-training

of networks [19]. Here, we believe that the combination of the shallow network and the regularization-based weight update methods with the online learning-prediction scheme can overcome these fundamental difficulties of the fast fading prediction and provide practical and computationally non-expensive prediction with high prediction accuracy. Our simulations and experiments demonstrated that the proposed CVNNs automatically change their effective connection numbers depending on the channel variation so that they keep appropriate network sizes to achieve accurate channel prediction without prior knowledge of the communication environments. The results presented in the experiment section for actually observed channels showed that the proposed method can provide more accurate prediction even in situations that are difficult for conventional methods including the time-domain linear, the AR-model-based, the LSTM-based, and the conventional CVNN-based predictions.

APPENDIX A
DERIVATION OF COMPLEX-VALUED (CV) STEEPEST DESCENT METHODS WITH L_1-NORM AND $L_{2,1}$-NORM PENALTIES

We consider the complex-valued steepest descent methods with the sparse constraints (L_1- and $L_{2,1}$-norm penalties) in a multi-layered network structure. Here, we consider weight updates at connections from layer l to layer $(l+1)$. The connection weight w_{lkj} to kth output of jth neuron/input terminal in layer l is expressed by its amplitude $|w_{lkj}|$ and phase θ_{lkj}. Input signals to neurons in layer $(l+1)$ are output signals from layer l:

$$z_{lj} = |z_{lj}|e^{i\theta_{lj}} \tag{15}$$

The internal state $u_{(l+1)k} = |u_{(l+1)k}|e^{i\theta_{(l+1)k}}$ of kth neuron in $(l+1)$th layer is obtained as the summation of the inputs $\boldsymbol{z}_l = [z_{lj}]$ weighted by $\boldsymbol{w}_{lk} = [w_{lkj}]$, i.e.,

$$u_{(l+1)k} \equiv \sum_j w_{lkj}z_{lj} = \sum_j |w_{lkj}||z_{lj}|e^{i(\theta_{lkj}+\theta_{lj})} \tag{16}$$

The output $z_{(l+1)k}$ of kth neuron in $(l+1)$th layer is, then, given by adopting an amplitude-phase-type activation function $f_{\rm ap}$ to $u_{(l+1)k}$ as

$$\begin{aligned} z_{(l+1)k} &\equiv f_{\rm ap}(u_{(l+1)k}) \\ &= \tanh(|u_{(l+1)k}|)e^{i\arg(u_{(l+1)k})} \\ &= \tanh(|u_{(l+1)k}|)e^{i\theta_{(l+1)k}} \end{aligned} \tag{17}$$

Here, we use $dw_{lkj'}/dt$ to represent the descent direction of a weight $w_{lkj'}$ in $\boldsymbol{w}_{lk}$. Then, the change direction of the internal state $u_{(l+1)k}$ due to the weight change can be expressed as

$$\frac{du_{(l+1)k}}{dt} = \left(\frac{d(|u_{(l+1)k}|)}{dt} + i|u_{(l+1)k}|\frac{d\theta_{(l+1)k}}{dt}\right)e^{i\theta_{(l+1)k}} \tag{18}$$

If we newly define the descent direction of the weight as

$$\frac{dw_{lkj'}}{dt} \equiv \left(\frac{dw^{\rm a}_{lkj'}}{dt} + i\frac{dw^{\rm p}_{lkj'}}{dt}\right)e^{i(\theta_{(l+1)k}-\theta_{lj'})} \tag{19}$$

by introducing two weight-change fractions in the directions of the weight amplitude and phase, $dw^{\rm a}_{lkj'}/dt$ and $w^{\rm p}_{lkj'}/dt$, on the complex plane, the change of the internal state $u_{(l+1)k}$ can also be represented as:

$$\frac{du_{(l+1)k}}{dt} = \frac{dw_{lkj'}}{dt}z_{lj'} = \left(\frac{dw^{\rm a}_{lkj'}}{dt} + i\frac{dw^{\rm p}_{lkj'}}{dt}\right)|z_{lj'}|e^{i\theta_{(l+1)k}} \tag{20}$$

Accordingly, we can obtain the change of the internal state $u_{(l+1)k}$ in terms of the weight-change fractions, for generalized $j = j'$, as

$$\frac{d|u_{(l+1)k}|}{dw^{\rm a}_{lkj}} = |z_{lj}| \tag{21}$$

$$\frac{d\theta_{(l+1)k}}{dw^{\rm p}_{lkj}} = \frac{|z_{lj}|}{|u_{(l+1)k}|} \tag{22}$$

Here, we obtain the relationship between the change of the weight $dw_{lkj'}/dt$ and the two fractions $dw^{\rm a}_{lkj'}/dt$ and $dw^{\rm p}_{lkj'}/dt$ by dividing the both sides of (19) by $e^{i\theta_{lkj'}}$ as

$$\begin{aligned} \frac{1}{e^{i\theta_{lkj'}}}\frac{dw_{lkj'}}{dt} &= \left(\frac{dw^{\rm a}_{lkj'}}{dt} + i\frac{dw^{\rm p}_{lkj'}}{dt}\right)e^{i(\theta_{(l+1)k}-\theta_{lj'}-\theta_{lkj'})} \\ &= \cos\theta^{\rm rot}_{lkj'}\frac{dw^{\rm a}_{lkj'}}{dt} - \sin\theta^{\rm rot}_{lkj'}\frac{dw^{\rm p}_{lkj'}}{dt} \\ &\quad + i\left(\cos\theta^{\rm rot}_{lkj'}\frac{dw^{\rm p}_{lkj'}}{dt} + \sin\theta^{\rm rot}_{lkj'}\frac{dw^{\rm a}_{lkj'}}{dt}\right) \end{aligned} \tag{23}$$

where $\theta^{\rm rot}_{lkj'} \equiv \theta_{(l+1)k} - \theta_{lj'} - \theta_{lkj'}$. On the other hand, similarly as (18), we can write $dw_{lkj'}/dt$ as

$$\frac{dw_{lkj'}}{dt} = \left(\frac{d(|w_{lkj'}|)}{dt} + i|w_{lkj'}|\frac{d\theta_{lkj'}}{dt}\right)e^{i\theta_{lkj'}} \tag{24}$$

Hence,

$$\frac{1}{e^{i\theta_{lkj'}}}\frac{dw_{lkj'}}{dt} = \left(\frac{d(|w_{lkj'}|)}{dt} + i|w_{lkj'}|\frac{d\theta_{lkj'}}{dt}\right) \tag{25}$$

By (23) and (25), we derive a relationship, for general $j = j'$, expressed as

$$\begin{bmatrix} \frac{d(|w_{lkj}|)}{dt} \\ |w_{lkj}|\frac{d\theta_{lkj}}{dt} \end{bmatrix} = \begin{bmatrix} \cos\theta^{\rm rot}_{lkj} & -\sin\theta^{\rm rot}_{lkj} \\ \sin\theta^{\rm rot}_{lkj} & \cos\theta^{\rm rot}_{lkj} \end{bmatrix} \begin{bmatrix} \frac{dw^{\rm a}_{lkj}}{dt} \\ \frac{dw^{\rm p}_{lkj}}{dt} \end{bmatrix} \tag{26}$$

where

$$\theta^{\rm rot}_{lkj} \equiv \theta_{(l+1)k} - \theta_{lj} - \theta_{lkj} \tag{27}$$

The explicit expression for the two change fractions can be written as

$$\begin{bmatrix} \frac{dw^{\rm a}_{lkj}}{dt} \\ \frac{dw^{\rm p}_{lkj}}{dt} \end{bmatrix} = \begin{bmatrix} \cos\theta^{\rm rot}_{lkj} & \sin\theta^{\rm rot}_{lkj} \\ -\sin\theta^{\rm rot}_{lkj} & \cos\theta^{\rm rot}_{lkj} \end{bmatrix} \begin{bmatrix} \frac{d(|w_{lkj}|)}{dt} \\ |w_{lkj}|\frac{d\theta_{lkj}}{dt} \end{bmatrix} \tag{28}$$

This reads the following rule of the weight changes:

$$\frac{d|w_{lkj}|}{dt} = -\frac{\partial E_{(l+1)}}{\partial|w_{lkj}|}$$

$$= -\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\frac{\partial w^{\mathrm{a}}_{lkj}}{\partial |w_{lkj}|} + \frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\frac{\partial w^{\mathrm{p}}_{lkj}}{\partial |w_{lkj}|}\Big)$$
$$= -\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\cos\theta^{\mathrm{rot}}_{lkj} - \frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\sin\theta^{\mathrm{rot}}_{lkj}\Big) \quad (29)$$

$$\frac{d\theta_{lkj}}{dt} = -\frac{1}{|w_{lkj}|}\frac{\partial E_{(l+1)}}{\partial \theta_{lkj}}$$
$$= -\frac{1}{|w_{lkj}|}\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\frac{\partial w^{\mathrm{a}}_{lkj}}{\partial \theta_{lkj}} + \frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\frac{\partial w^{\mathrm{p}}_{lkj}}{\partial \theta_{lkj}}\Big)$$
$$= -\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\sin\theta^{\mathrm{rot}}_{lkj} + \frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\cos\theta^{\mathrm{rot}}_{lkj}\Big) \quad (30)$$

For the conventional objective function (6) [21], we get

$$\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}} = \frac{\partial E_{(l+1)}}{\partial(|u_{(l+1)k}|)}\frac{d(|u_{(l+1)k}|)}{dw^{\mathrm{a}}_{lkj}}$$
$$= (1-|z_{(l+1)k}|^2)$$
$$\times\Big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\Big)|z_{lj}| \quad (31)$$

$$\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}} = \frac{\partial E_{(l+1)}}{\partial \theta_{(l+1)k}}\frac{d\theta_{(l+1)k}}{dw^{\mathrm{p}}_{lkj}}$$
$$= |z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\frac{|z_{lj}|}{|u_{(l+1)k}|} \quad (32)$$

since we have the relationships (21) and (22) as well as

$$E_{(l+1)} \equiv \frac{1}{2}|\boldsymbol{z}_{(l+1)} - \hat{\boldsymbol{z}}_{(l+1)}|^2$$
$$= \frac{1}{2}\sum_k |z_{(l+1)k} - \hat{z}_{(l+1)k}|^2$$
$$= \frac{1}{2}\sum_k \Big(\tanh^2(|u_{(l+1)k}|) + \tanh^2(|\hat{u}_{(l+1)k}|)$$
$$- 2\tanh(|u_{(l+1)k}|)\tanh(|\hat{u}_{(l+1)k}|)$$
$$\times\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\Big) \quad (33)$$

where $\hat{\boldsymbol{z}}_{(l+1)} = [\hat{z}_{(l+1)k}]$ are the teacher signals of the outputs $\boldsymbol{z}_{(l+1)} = [z_{(l+1)k}]$ and $\hat{\boldsymbol{u}}_{(l+1)} = [|\hat{u}_{(l+1)k}|e^{i\hat{\theta}_{(l+1)k}}]$ are the equivalent internal state corresponding to the teacher signals.

The objective function with the L_1-norm penalty is

$$E^{\mathrm{S}}_{(l+1)} = \frac{1}{2}|\boldsymbol{z}_{(l+1)} - \hat{\boldsymbol{z}}_{(l+1)}|^2 + \alpha\|\boldsymbol{W}_l\|_1$$
$$= E_{(l+1)} + \alpha\sum_k\sum_j |w_{lkj}| \quad (34)$$

where α is a penalty coefficient. Since $\partial(|w_{lkj}|)/\partial w^{\mathrm{a}}_{lkj} = \cos\theta^{\mathrm{rot}}_{lkj}$ and $\partial(|w_{lkj}|)/\partial w^{\mathrm{p}}_{lkj} = -\sin\theta^{\mathrm{rot}}_{lkj}$ by (26), the descent direction of the steepest descent with $E^{\mathrm{S}}_{(l+1)}$ can be written as

$$\frac{d|w_{lkj}|}{dt} = -\frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial |w_{lkj}|}$$
$$= -\Big(\frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\cos\theta^{\mathrm{rot}}_{lkj} - \frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\sin\theta^{\mathrm{rot}}_{lkj}\Big)$$
$$= -\Big\{\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}} + \alpha\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{a}}_{lkj}}\Big)\cos\theta^{\mathrm{rot}}_{lkj}$$
$$-\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}} + \alpha\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{p}}_{lkj}}\Big)\sin\theta^{\mathrm{rot}}_{lkj}\Big\}$$
$$= -\Big\{(1-|z_{(l+1)k}|^2)$$
$$\times\Big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\Big)$$
$$\times|z_{lj}|\cos\theta^{\mathrm{rot}}_{lkj}$$
$$-|z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})$$
$$\times\frac{|z_{lj}|}{|u_{(l+1)k}|}\sin\theta^{\mathrm{rot}}_{lkj}$$
$$+\alpha\big(\cos^2\theta^{\mathrm{rot}}_{lkj} + \sin^2\theta^{\mathrm{rot}}_{lkj}\big)\Big\} \quad (35)$$

$$\frac{d\theta_{lkj}}{dt} = -\frac{1}{|w_{lkj}|}\frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial \theta_{lkj}}$$
$$= -\Big(\frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\sin\theta^{\mathrm{rot}}_{lkj} + \frac{\partial E^{\mathrm{S}}_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\cos\theta^{\mathrm{rot}}_{lkj}\Big)$$
$$= -\Big\{\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}} + \alpha\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{a}}_{lkj}}\Big)\sin\theta^{\mathrm{rot}}_{lkj}$$
$$+\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}} + \alpha\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{p}}_{lkj}}\Big)\cos\theta^{\mathrm{rot}}_{lkj}\Big\}$$
$$= -\Big\{(1-|z_{(l+1)k}|^2)$$
$$\times\Big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})\Big)$$
$$\times|z_{lj}|\sin\theta^{\mathrm{rot}}_{lkj}$$
$$+|z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k} - \hat{\theta}_{(l+1)k})$$
$$\times\frac{|z_{lj}|}{|u_{(l+1)k}|}\cos\theta^{\mathrm{rot}}_{lkj}$$
$$+\alpha\big(\cos\theta^{\mathrm{rot}}_{lkj}\sin\theta^{\mathrm{rot}}_{lkj} - \cos\theta^{\mathrm{rot}}_{lkj}\sin\theta^{\mathrm{rot}}_{lkj}\big)\Big\} \quad (36)$$

Therefore, we obtain (8) and (9).

For the objective function with the $L_{2,1}$-norm penalty,

$$E^{\mathrm{GS}}_{(l+1)} = \frac{1}{2}|\boldsymbol{z}_{(l+1)} - \hat{\boldsymbol{z}}_{(l+1)}|^2 + \alpha\|\boldsymbol{W}_l\|_{2,1}$$
$$= E_{(l+1)} + \alpha\sum_j\Big(\sqrt{|\boldsymbol{w}_{lj}|}\sqrt{\sum_k |w_{lkj}|^2}\Big) \quad (37)$$

where $\boldsymbol{w}_{lj} = [w_{lkj}]$ and $|\boldsymbol{w}_{lj}|$ denotes the dimensionality of the vector $\boldsymbol{w}_{lj}$. Based on (26) and $w_{lkj} = |w_{lkj}|e^{i\theta_{lkj}}$, we can get the following relationship:

$$\frac{\partial|w_{lkj}|^2}{\partial w^{\mathrm{a}}_{lkj}} = \frac{\partial w_{lkj}}{\partial w^{\mathrm{a}}_{lkj}}w^*_{lkj} + w_{lkj}\frac{\partial w^*_{lkj}}{\partial w^{\mathrm{a}}_{lkj}}$$
$$= \Big(\frac{\partial w_{lkj}}{\partial|w_{lkj}|}\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{a}}_{lkj}} + \frac{\partial w_{lkj}}{\partial\theta_{lkj}}\frac{\partial\theta_{lkj}}{\partial w^{\mathrm{a}}_{lkj}}\Big)w^*_{lkj}$$
$$+ w_{lkj}\Big(\frac{\partial w^*_{lkj}}{\partial|w_{lkj}|}\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{a}}_{lkj}} + \frac{\partial w^*_{lkj}}{\partial\theta_{lkj}}\frac{\partial\theta_{lkj}}{\partial w^{\mathrm{a}}_{lkj}}\Big)$$
$$= 2|w_{lkj}|\cos\theta^{\mathrm{rot}}_{lkj} \quad (38)$$

$$\begin{aligned}\frac{\partial|w_{lkj}|^2}{\partial w^{\mathrm{p}}_{lkj}} &= \frac{\partial w_{lkj}}{\partial w^{\mathrm{p}}_{lkj}} w^*_{lkj} + w_{lkj}\frac{\partial w^*_{lkj}}{\partial w^{\mathrm{p}}_{lkj}} \\ &= \Big(\frac{\partial w_{lkj}}{\partial|w_{lkj}|}\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{p}}_{lkj}} + \frac{\partial w_{lkj}}{\partial\theta_{lkj}}\frac{\partial\theta_{lkj}}{\partial w^{\mathrm{p}}_{lkj}}\Big) w^*_{lkj} \\ &\quad + w_{lkj}\Big(\frac{\partial w^*_{lkj}}{\partial|w_{lkj}|}\frac{\partial|w_{lkj}|}{\partial w^{\mathrm{p}}_{lkj}} + \frac{\partial w^*_{lkj}}{\partial\theta_{lkj}}\frac{\partial\theta_{lkj}}{\partial w^{\mathrm{p}}_{lkj}}\Big) \\ &= -2|w_{lkj}|\sin\theta^{\mathrm{rot}}_{lkj} \end{aligned} \tag{39}$$

By (38) and (39), the descent direction of the steepest descent with $E^{\mathrm{GS}}_{(l+1)}$ can be written as

$$\begin{aligned}\frac{d|w_{lkj}|}{dt} &= -\frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial|w_{lkj}|} \\ &= -\Big(\frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\cos\theta^{\mathrm{rot}}_{lkj} - \frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\sin\theta^{\mathrm{rot}}_{lkj}\Big) \\ &= -\Big\{\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}} + \alpha\frac{\sqrt{|\boldsymbol{w}_{lj}|}}{2\|\boldsymbol{w}_{lj}\|_2}\frac{\partial|w_{lkj}|^2}{\partial w^{\mathrm{a}}_{lkj}}\Big)\cos\theta^{\mathrm{rot}}_{lkj} \\ &\quad -\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}} + \alpha\frac{\sqrt{|\boldsymbol{w}_{lj}|}}{2\|\boldsymbol{w}_{lj}\|_2}\frac{\partial|w_{lkj}|^2}{\partial w^{\mathrm{p}}_{lkj}}\Big)\sin\theta^{\mathrm{rot}}_{lkj}\Big\} \\ &= -\Big\{(1-|z_{(l+1)k}|^2) \\ &\quad \times\Big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k}-\hat{\theta}_{(l+1)k})\Big) \\ &\quad \times|z_{lj}|\cos\theta^{\mathrm{rot}}_{lkj} \\ &\quad -|z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k}-\hat{\theta}_{(l+1)k}) \\ &\quad \times\frac{|z_{lj}|}{|u_{(l+1)k}|}\sin\theta^{\mathrm{rot}}_{lkj} \\ &\quad +\alpha\sqrt{|\boldsymbol{w}_{lj}|}\frac{|w_{lkj}|}{\|\boldsymbol{w}_{lj}\|_2}\big(\cos^2\theta^{\mathrm{rot}}_{lkj}+\sin^2\theta^{\mathrm{rot}}_{lkj}\big)\Big\} \end{aligned} \tag{40}$$

$$\begin{aligned}\frac{d\theta_{lkj}}{dt} &= -\frac{1}{|w_{lkj}|}\frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial\theta_{lkj}} \\ &= -\Big(\frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}}\sin\theta^{\mathrm{rot}}_{lkj} + \frac{\partial E^{\mathrm{GS}}_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}}\cos\theta^{\mathrm{rot}}_{lkj}\Big) \\ &= -\Big\{\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{a}}_{lkj}} + \alpha\frac{\sqrt{|\boldsymbol{w}_{lj}|}}{2\|\boldsymbol{w}_{lj}\|_2}\frac{\partial|w_{lkj}|^2}{\partial w^{\mathrm{a}}_{lkj}}\Big)\sin\theta^{\mathrm{rot}}_{lkj} \\ &\quad +\Big(\frac{\partial E_{(l+1)}}{\partial w^{\mathrm{p}}_{lkj}} + \alpha\frac{\sqrt{|\boldsymbol{w}_{lj}|}}{2\|\boldsymbol{w}_{lj}\|_2}\frac{\partial|w_{lkj}|^2}{\partial w^{\mathrm{p}}_{lkj}}\Big)\cos\theta^{\mathrm{rot}}_{lkj}\Big\} \\ &= -\Big\{(1-|z_{(l+1)k}|^2) \\ &\quad \times\Big(|z_{(l+1)k}| - |\hat{z}_{(l+1)k}|\cos(\theta_{(l+1)k}-\hat{\theta}_{(l+1)k})\Big) \\ &\quad \times|z_{lj}|\sin\theta^{\mathrm{rot}}_{lkj} \\ &\quad +|z_{(l+1)k}||\hat{z}_{(l+1)k}|\sin(\theta_{(l+1)k}-\hat{\theta}_{(l+1)k}) \\ &\quad \times\frac{|z_{lj}|}{|u_{(l+1)k}|}\cos\theta^{\mathrm{rot}}_{lkj} \\ &\quad +\alpha\sqrt{|\boldsymbol{w}_{lj}|}\frac{|w_{lkj}|}{\|\boldsymbol{w}_{lj}\|_2}\big(\cos\theta^{\mathrm{rot}}_{lkj}\sin\theta^{\mathrm{rot}}_{lkj} \\ &\quad -\cos\theta^{\mathrm{rot}}_{lkj}\sin\theta^{\mathrm{rot}}_{lkj}\big)\Big\} \end{aligned} \tag{41}$$

Therefore, we obtain (11) and (12).

Note that we, in this work, use the steepest descent on the regularization problems, which may only guarantee significantly small weights instead of providing exact-zero amplitude for redundant weight connections. Although significantly small versus exactly zero weights have limited difference on the performance of the channel prediction, that is prediction accuracy, one may want exact-zero weights or completely pruned neurons for reduction of power consumption in practical mobile communications. A possible extension of the proposed methods by utilizing other proximal algorithms such as iterative shrinkage-thresholding algorithm (ISTA) in the optimization of the regularization [55], [56] is an interesting topic but we leave this to future investigation because this is out of the scope of this article.

APPENDIX B
STRUCTURE OF LONG SHORT-TERM MEMORY NETWORK

As a performance comparision of channel prediction, a simple long short-term memory (LSTM) network [54], a variation of real-valued recurrent neural networks, is used in this work. The mathematical description of the LSTM network is as follows:

$$\boldsymbol{y}_t = \sigma(\boldsymbol{w}_y\boldsymbol{x}_t + \boldsymbol{U}_y\boldsymbol{h}_{(t-1)}) \tag{42}$$
$$\boldsymbol{r}_t = \sigma(\boldsymbol{w}_r\boldsymbol{x}_t + \boldsymbol{U}_r\boldsymbol{h}_{(t-1)}) \tag{43}$$
$$\tilde{\boldsymbol{h}}_t = \tanh\big(\boldsymbol{w}\boldsymbol{x}_t + \boldsymbol{U}(\boldsymbol{r}_t\odot\boldsymbol{h}_{(t-1)})\big) \tag{44}$$
$$\boldsymbol{h}_t = \boldsymbol{y}_t\odot\boldsymbol{h}_{(t-1)} + (1-\boldsymbol{y}_t)\odot\tilde{\boldsymbol{h}}_t \tag{45}$$

where $\boldsymbol{x}_t$, $\boldsymbol{h}_{(t-1)}$, $\boldsymbol{y}_t$, $\boldsymbol{r}_t$, $\tilde{\boldsymbol{h}}_t$ and $\boldsymbol{h}_t$ are inputs, previous hidden states, update gates, reset gates, intermediate hidden states and present hidden states, respectively, $\boldsymbol{w}_y$, $\boldsymbol{U}_y$, $\boldsymbol{w}_r$, $\boldsymbol{U}_r$, $\boldsymbol{w}$ and $\boldsymbol{U}$ are weight matrices which are learned, and $\odot$ denotes the Hadamard product of vectors and $\sigma(x) = 1/(1+e^{-x})$. A set of estimated past path characteristics $\hat{c}_m(t-1), \ldots, \hat{c}_m(t-I_{\mathrm{ML}})$ is converted into the input vector $\boldsymbol{x}_t$ by splitting real and imaginary parts of the complex value. The weight matrices are updated by using the steepest descent method with the standard error-backpropagation so that they minimize the difference

$$E^{\mathrm{lstm}} \equiv \frac{1}{2}|\boldsymbol{h}_t - \hat{\boldsymbol{h}}_t|^2 \tag{46}$$

where $\hat{\boldsymbol{h}}_t$ denotes the teacher signals of $\boldsymbol{h}_t$ consists of real and imaginary parts of $\hat{c}_m(t)$. The learned weights are kept internally in the network and updated in the following time points until the latest channel component is used. The future channel states are predicted by the most up-to-dated weights.

REFERENCES

[1] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications With MATLAB*. Chichester, U.K.: Wiley, Aug. 2010.

[2] C. D. Ho, H. Q. Ngo, M. Matthaiou, and T. Q. Duong, "On the performance of zero-forcing processing in multi-way massive MIMO relay networks," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 849–852, Apr. 2017.

[3] E. Eraslan, B. Daneshrad, and C.-Y. Lou, "Performance indicator for MIMO MMSE receivers in the presence of channel estimation error," *IEEE Wireless Commun. Lett.*, vol. 2, no. 2, pp. 211–214, Apr. 2013.

[4] X. Ren, J. Wu, K. H. Johansson, G. Shi, and L. Shi, "Infinite horizon optimal transmission power control for remote state estimation over fading channels," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 85–100, Jan. 2018.

[5] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proc. IEEE*, vol. 95, no. 12, pp. 2299–2313, Dec. 2007.

[6] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1790–1821, 3rd Quart., 2017.

[7] F. Maehara, F. Sasamori, and F. Tkahata, "Linear predictive maximal ratio combining transmitter diversity for OFDM-TDMA/TDD systems," *IEICE Trans. Commun.*, vol. E86-B, no. 1, pp. 221–229, 2003.

[8] H. P. Bui, Y. Ogawa, T. Nishimura, and T. Ohgane, "Performance evaluation of a multi-user MIMO system with prediction of time-varying indoor channels," *IEEE Trans. Antennas Propag.*, vol. 61, no. 1, pp. 371–379, Jan. 2013.

[9] T. Eyceoz, A. Duel-Hallen, and H. Hallen, "Deterministic channel modeling and long range prediction of fast fading mobile radio channels," *IEEE Commun. Lett.*, vol. 2, no. 9, pp. 254–256, Sep. 1998.

[10] A. Arredondo, K. R. Dandekar, and G. Xu, "Vector channel modeling and prediction for the improvement of downlink received power," *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1121–1129, Jul. 2002.

[11] A. Duel-Hallen, H. Hallen, and T.-S. Yang, "Long range prediction and reduced feedback for mobile radio adaptive OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 5, no. 10, pp. 2723–2733, Oct. 2006.

[12] P. Sharma and K. Chandra, "Prediction of state transitions in Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 416–425, Mar. 2007.

[13] T. Ding and A. Hirose, "Fading channel prediction based on combination of complex-valued neural networks and chirp Z-Transform," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1686–1695, Sep. 2014.

[14] Y. Zhao, H. Gao, N. C. Beaulieu, Z. Chen, and H. Ji, "Echo state network for fast channel prediction in ricean fading scenarios," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 672–675, Mar. 2017.

[15] Y. Sui, W. Yu, and Q. Luo, "Jointly optimized extreme learning machine for short-term prediction of fading channel," *IEEE Access*, vol. 6, pp. 49029–49039, 2018.

[16] W. Liu, L.-L. Yang, and L. Hanzo, "Recurrent neural network based narrowband channel prediction," in *Proc. IEEE 63rd Veh. Technol. Conf.*, vol. 5, May 2006, pp. 2173–2177.

[17] C. Potter, G. K. Venayagamoorthy, and K. Kosbar, "RNN based MIMO channel prediction," *Signal Process.*, vol. 90, no. 2, pp. 440–450, Feb. 2010.

[18] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan. 2020.

[19] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[20] J. Joo, M. C. Park, D. S. Han, and V. Pejovic, "Deep learning-based channel prediction in realistic vehicular communications," *IEEE Access*, vol. 7, pp. 27846–27858, 2019.

[21] A. Hirose, *Complex-Valued Neural Networks*, 2nd ed. New York, NY, USA: Springer-Verlag, 2012, vol. 400.

[22] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 96–108, Nov. 2017.

[23] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[24] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York, NY, USA: Springer, 2010.

[25] T. Ding and A. Hirose, "Fading channel prediction based on self-optimizing neural networks," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 8834. Cham, Switzerland: Springer, 2014, pp. 175–182.

[26] A. Aghasi, A. Abdi, N. Nguyen, and J. Romberg, "Net-trim: Convex pruning of deep neural networks with performance guarantee," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 3178–3187.

[27] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini, "Group sparse regularization for deep neural networks," *Neurocomputing*, vol. 241, pp. 81–89, Jun. 2017.

[28] B. N. G. Koneru and V. Vasudevan, "Sparse artificial neural networks using a novel smoothed LASSO penalization," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 5, pp. 848–852, May 2019.

[29] A. Hirose and R. Eckmiller, "Coherent optical neural networks that have optical-frequency-controlled behavior and generalization ability in the frequency domain," *Appl. Opt.*, vol. 35, no. 5, p. 836, Feb. 1996.

[30] T. Ding and A. Hirose, "Proposal of online regularization for dynamical structure optimization in complex-valued neural networks," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 11954. Cham, Switzerland: Springer, 2019, pp. 407–418.

[31] W. Jakes, *Microwave Mobile Communications*, 2nd ed. New York, NY, USA: Wiley, 1994.

[32] S. Tan and A. Hirose, "Low-calculation-cost fading channel prediction using chirp Z-transform," *Electron. Lett.*, vol. 45, no. 8, p. 418, 2009.

[33] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.

[34] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *Proc. ICLR*, May 2018, pp. 1–19.

[35] T. Hara and A. Hirose, "Plastic mine detecting radar system using complex-valued self-organizing map that deals with multiple-frequency interferometric images," *Neural Netw.*, vol. 17, nos. 8–9, pp. 1201–1210, Oct. 2004.

[36] S. Kawata and A. Hirose, "Frequency-multiplexed logic circuit based on a coherent optical neural network," *Appl. Opt.*, vol. 44, no. 19, p. 4053, Jul. 2005.

[37] M. E. Valle, "Complex-valued recurrent correlation neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1600–1612, Sep. 2014.

[38] Y. Arima and A. Hirose, "Performance dependence on system parameters in millimeter-wave active imaging based on complex-valued neural networks to classify complex texture," *IEEE Access*, vol. 5, pp. 22927–22939, 2017.

[39] A. Hirose, "Applications of complex-valued neural networks to coherent optical computing using phase-sensitive detection scheme," *Inf. Sci.-Appl.*, vol. 2, no. 2, pp. 103–117, Sep. 1994.

[40] M. Ishikawa, "Structural learning with forgetting," *Neural Netw.*, vol. 9, no. 3, pp. 509–521, Apr. 1996.

[41] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Apr. 2005.

[42] T.-C. Lu, G.-R. Yu, and J.-C. Juang, "Quantum-based algorithm for optimizing artificial neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1266–1278, Aug. 2013.

[43] E. D. Karnin, "A simple procedure for pruning back-propagation trained neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 2, pp. 239–242, Jun. 1990.

[44] R. Reed, "Pruning algorithms—A survey," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, Sep. 1993.

[45] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, Jul. 1993.

[46] M. Barakat, F. Druaux, D. Lefebvre, M. Khalil, and O. Mustapha, "Self adaptive growing neural network classifier for faults detection and diagnosis," *Neurocomputing*, vol. 74, no. 18, pp. 3865–3876, Nov. 2011.

[47] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.

[48] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[49] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[50] D. L. Donoho and J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lowers dimension," *J. Amer. Math. Soc.*, vol. 22, no. 1, pp. 1–53, Jul. 2008.

[51] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 68, no. 1, pp. 49–67, Feb. 2006.
[52] J. Wang, C. Xu, X. Yang, and J. M. Zurada, "A novel pruning algorithm for smoothing feedforward neural networks based on group lasso method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 2012–2024, May 2018.
[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[54] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, vol. 28, no. 4, pp. 1724–1734.
[55] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
[56] Z. Li, S. Ding, and Y. Li, "A fast algorithm for learning overcomplete dictionary for sparse representation based on proximal operators," *Neural Comput.*, vol. 27, no. 9, pp. 1951–1982, Sep. 2015.

TIANBEN DING received the B.Eng. degree in physics from Yokohama National University, in 2012, and the M.Eng. degree in electrical engineering and information systems from The University of Tokyo, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering with the Washington University in St. Louis. His current research interests include neural networks, optical imaging systems, and single-molecule fluorescence.

AKIRA HIROSE (Fellow, IEEE) received the Ph.D. degree in electronic engineering from The University of Tokyo, in 1991.

In 1987, he joined the Research Center for Advanced Science and Technology (RCAST), The University of Tokyo, as a Research Associate. In 1991, he was appointed as an Instructor at RCAST. From 1993 to 1995, on leave of absence from The University of Tokyo, he joined the Institute for Neuroinformatics, University of Bonn, Bonn, Germany. He is currently a Professor with the Department of Electrical Engineering and Information Systems, The University of Tokyo. In the fields, he published several books such as *Complex-Valued Neural Networks* (Second Edition, Springer, 2012). His research interests include wireless electronics and neural networks. He is a Fellow of IEICE and a member of JNNS and APNNS. He currently serves as a member of the IEEE Computational Intelligence Society (CIS) and the Neural Networks Technical Committee (NNTC), since 2009. He has been a Founding Chair of the NNTC Complex-Valued Neural Network Task Force, since 2010, and a Governing Board Member of APNNA/APNNS, since 2006. He served previously as the President of Asia-Pacific Neural Network Society (APNNS), in 2016, the President of Japanese Neural Network Society (JNNS), from 2013 to 2015, the Vice President of the IEICE Electronics Society (ES), from 2013 to 2015, the Editor-in-Chief of the *IEICE Transactions on Electronics*, from 2011 to 2012, an Associate Editor of journals such as the IEEE TRANSACTIONS ON NEURAL NETWORKS, from 2009 to 2011, the IEEE GEOSCIENCE AND REMOTE SENSING NEWSLETTER, from 2009 to 2012, the IEEE GRSS All Japan Chapter Chair, from 2013 to 2015, the IEEE CIS All Japan Chapter Chair, from 2017 to 2018, and also as a General Chair of Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), in 2013 Tsukuba, the International Conference on Neural Information Processing (ICONIP), Kyoto, in 2016, and the International Geoscience and Remote Sensing Symposium (IGARSS), Yokohama, in 2019.