Deep Learning-based Channel Estimation and Tracking for Millimeter-wave Vehicular Communications

Sangmi Moon, Hyunsung Kim, and Intae Hwang

Abstract: **The application of millimeter-wave (mmWave) frequencies is a potential technology for satisfying the continuously increasing need for handling data traffic in highly advanced wireless communications. A substantial challenge presented in mmWave communications is the high path loss. mmWave systems adopt beamforming techniques to overcome this issue. These require robust channel estimation and tracking algorithm for maintenance of an adequate quality of service. In this study, we propose a deep learning-based channel estimation and tracking algorithm for vehicular mmWave communications. More specifically, a deep neural network is leveraged to learn the mapping function between the received omni-beam patterns and mmWave channel with negligible overhead. Following the channel estimation, long short-term memory is leveraged to track the channel. The simulation results demonstrate that the proposed algorithm estimates and tracks the mmWave channel efficiently with negligible training overhead.**

Index Terms: **Channel estimation, channel tracking, deep learning, deep neural network, long short-term memory, mmWave.**

I. INTRODUCTION

THE use of millimeter-wave (mmWave) frequencies is a potential technology for supporting high data rates for highly advanced wireless communications [1], [2]. However, mmWave communications exhibit shortcomings such as signal attenuation and reduced transmission distance owing to their short wavelengths (high frequencies) [3], [4]. However, mmWaves are suitable for use in massive multiple-input–multiple-output (MIMO) systems, wherein multiple antennas are installed within a small space. Based on these features, many studies have been performed to overcome the large path losses encountered in mmWave bands via a highly directional beamforming technique [5]–[7]. To perform high directional beamforming, it is necessary to estimate and track channels for all the transmitter and receiver antenna pairs. In this study, we develop a novel algorithm for channel estimation and tracking by leveraging a deep learning tool for mmWave communications.

A. Prior Work

Beam training based on grid-of-beams is the de facto approach for configuring transmitted and received beams. Its variations are used in IEEE 802.11ad systems [8], [9] and 5G [10]. However, the dependence of its performance on the grid resolution results in high complexity, significant training overhead, and access delays. To reduce the training overhead, Berger [11] leveraged the sparse nature of the mmWave channels and developed compressed sensing-based channel estimation. Although the training overhead that these techniques generally incur is less than that incurred by exhaustive search solutions, it is still large for massive array systems. Moreover, it scales with the number of antennas. Furthermore, compressed sensing-based estimation techniques generally make difficult assumptions on the exact sparsity of the channels. This decreases their practical feasibility. For fast-changing environments such as moving vehicles, following channel estimation, fast beam-tracking methods are required to prolong the duration of communication between the transmitter and receiver. The classical Kalman filter can be employed to track the non-line-of-sight (NLOS) paths by first eliminating their influence [12]. In [13], the concept of a Kalman filter was also exploited while designing algorithms for angle-tracking and abrupt change detection. In [14], the extended Kalman filter was used to track a channel's angles of departure (AoDs) and angles of arrival (AoAs) via the measurement of only one beam pair. However, the angle-tracking algorithms developed in [12]–[14] depended on specific modeling of the geometric relationship between the base stations (BSs) and user equipment (UE) and the angle variations.

For a long time, machine learning techniques have been known to be highly effective tools for classification and regression (prediction) problems. More recently, deep learning has emerged with more advanced tools capable of constructing universal classifiers and/or approximate general functions. Typical problems/scenarios where machine learning methods have been successfully applied include, but are not limited to, image restoration and identification, natural language processing, network security, customer segmentation, and predictive maintenance (e.g., for machinery in industrial plants). Over the past two decades, the application of machine learning/deep learning techniques to communication problems has been, to a large extent, confined to the field of wireless communication systems. In [15], a deep learning-based coordinated beamforming was proposed. Here, orthogonal frequency-division multi-

Manuscript received December 20, 2019; revised March 30, 2020; approved for publication by Chan Byoung Chae, Guest Editor, May 24, 2020.

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2020-2016-0-00314) supervised by the Institute for Information & communications Technology Planning & Evaluation (IITP). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: Ministry of Science and ICT) (2018R1A2B6002255).

S. Moon, H. Kim, and I.-T. Hwang are with the Department of Electronic Engineering, Chonnam National University, email: msm0804@jnu.ac.kr, hsk940127@naver.com, hit@jnu.ac.kr.

I.-T. Hwang is the corresponding author.

Digital Object Identifier: 10.1109/JCN.2020.000012

plexing (OFDM) symbols received from multiple BSs constitute the input to a deep neural network (DNN). In [16], deep learning-based selection of mmWave beam by using the channel state information (CSI) of a sub-6-GHz channel was proposed. DNN-based beam selection using power delay profile (PDP) was proposed in [17]. An artificial neural network-based channel modeling was proposed for molecular MIMO communication in [18]. In [19], deep learning was used successfully in joint channel estimation and signal detection of OFDM systems with interference and nonlinear distortions. In [20], the authors proposed a deep learning-based scheme for achieving super-resolution direction-of-arrival (DOA) estimation and channel estimation in a massive MIMO system. In [21], to reduce the CSI feedback overhead of a frequency duplex division (FDD) massive MIMO system, deep learning has been employed to compress the channel into a low-dimensional codeword and then achieve recovery with high accuracy. The authors of [22] constructed the prediction model based on a long short-term memory (LSTM) structure to track the channel in a vehicular scenario. Importantly, the advantages of the deep learning-based communications solutions are demonstrated briefly in the aforementioned work.

B. Contribution

In this study, we propose a novel deep learning-based algorithm for channel estimation and tracking for mmWave vehicular communications. In accordance with [15], the developed channel estimation would require the UE to transmit only one uplink training sequence that is received jointly by multiple BSs using omni-directional beam patterns, i.e., with negligible training overhead. These received training signals represent the radio-frequency (RF) signature of both the environment and transmitter/receiver locations. A DNN is then leveraged to learn the implicit mapping function between the received training signals and mmWave channel. After the channel estimation, LSTM is leveraged to track the channel. The main contributions of this study can be summarized as follows:

- As is established, the conventional channel estimation techniques such as beam training and compressed sensing incur a large training overhead for massive MIMO systems. Furthermore, the overhead scales with the number of antennas. Therefore, in this study, the signals received at the coordinating BSs with only omni-beam pattern are considered, so that the proposed algorithm requires negligible time overhead for estimating the channel.
- We propose a method algorithm integrating deep learning and channel estimation/tracking, and develop its deep learning modeling. Here, the DNN can obtain the estimated channel using an omni-beam pattern, with negligible overhead. Then, we track the channel using LSTM, which employs the past channel to promote the prediction of the user's channel.
- We conduct a performance analysis of the proposed deep learning algorithm of massive MIMO in vehicular communications. Specifically, we simulate the normalized mean square error (NMSE) for assessing the accuracy of the channel estimation and tracking. In addition, the effective achievable rate demonstrated the efficiency of the proposed algorithm with negligible training overhead, rendering it a potential enabling-solution for fast-changing environments.

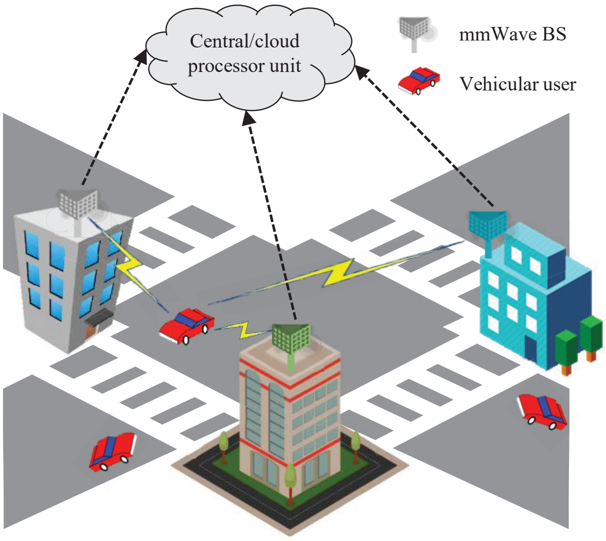

Fig. 1. Illustration of the considered coordinated mmWave system, where N BSs serve a vehicular UE. Each BS is equipped with M antennas and one RF chain, and applies analog-only beamforming/combining during the downlink/uplink transmission. The UE has only one antenna.

The remainder of this paper is organized as follows: In section II, we introduce the system and channel models for mmWave vehicular communications. In section III, we describe the proposed channel estimation and tracking. The simulation results are presented in section IV. Finally, the paper is concluded in section V.

II. SYSTEM AND CHANNEL MODEL

In this section, we describe the adopted coordinated mmWave system and channel models.

A. System Model

Because of the large path loss in mmWave communications, the service range of mmWave BSs is smaller than that of 4G BSs. This results in the dense coverage of mmWave BSs. In similar cases, a solution for enhancing the coverage of dense mmWave systems is to coordinate the transmission between multiple BSs to serve the same UE simultaneously [23], [24]. We consider a coordinated mmWave communication system, where N BSs serve a vehicular UE simultaneously, as illustrated in Fig. 1. Each BS is equipped with $M(= M_h \times M_v)$ antennas, whereby a uniform planar array (UPA) is formed. The UE has only one antenna. The BSs are assumed to be connected to each other so that they can share the uplink training signals received from the mobile user. For simplicity, we assume that each BS has only one radio-frequency (RF) chain and applies analog-only beamforming/combining through a network of phase shifters during the downlink/uplink transmission [5]. Extensions of this system to more sophisticated mmWave beamforming architectures at the BSs, such as hybrid beamforming [6], [7] are also topics of interest for future research. The results of this study can be straightforwardly extended to the case of multi-antenna users.

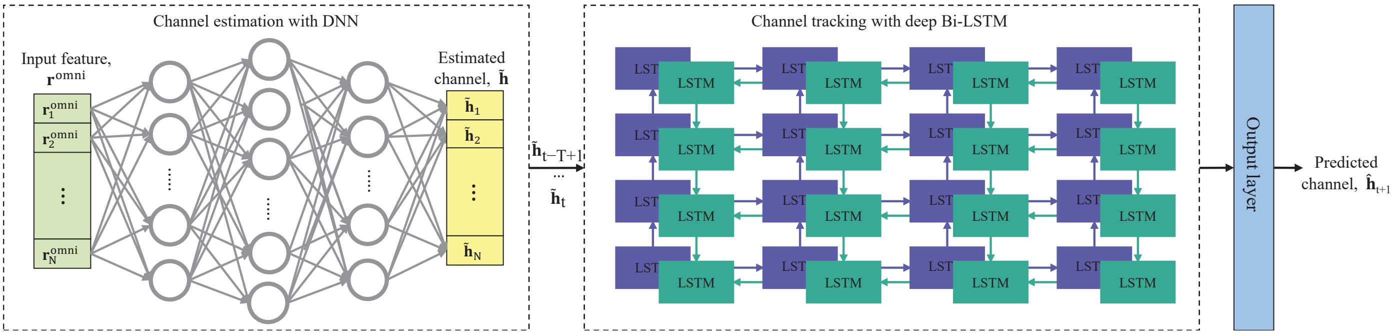
Channel estimation with DNN
Input feature, $\mathbf{r}^{\text{omni}}$
$\mathbf{r}_1^{\text{omni}}$
$\mathbf{r}_2^{\text{omni}}$
⋮
$\mathbf{r}_N^{\text{omni}}$
Estimated channel, $\tilde{\mathbf{h}}$
$\tilde{\mathbf{h}}_1$
$\tilde{\mathbf{h}}_2$
⋮
$\tilde{\mathbf{h}}_N$
$\tilde{\mathbf{h}}_{t-T+1}$
...
$\tilde{\mathbf{h}}_t$
Channel tracking with deep Bi-LSTM
LSTM
Output layer
Predicted channel, $\hat{\mathbf{h}}_{t+1}$

Fig. 2. The overall structure of the proposed channel estimation and tracking. We first estimate the channel using DNN. Then, the channel is tracked by LSTM using the estimated channel that is outputted by the DNN.

For channel estimation and tracking with uplink beam training, a vehicular UE transmits the known symbol $s_{\text{pilot}} = 1$, and each BS estimates the channel using the received signal. Denoting the channel vector $\mathbf{h}_n \in \mathbb{C}^{M\times 1}$ between the vehicular UE and nth BS, the post-combining received signal at nth BS can then be expressed as

$$r_n = \mathbf{f}_n^T \mathbf{h}_n s_{\text{pilot}} + \mathbf{f}_n^T \mathbf{v}_n, \tag{1}$$

where $\mathbf{f}_n \in \mathbb{C}^{M\times 1}$ is the analog beamforming vector and $\mathbf{v}_n \sim \mathcal{CN}(0, \sigma^2)$ is the receiver noise at the nth BS. In the downlink transmission, the data symbol $s_{\text{data}} \in \mathbb{C}$ is pre-coded using beamforming vector $\mathbf{f}_n$ at the nth BS. The received signal at the vehicular UE can be expressed as

$$y = \sum_{n=1}^{N} \mathbf{h}_n^T \mathbf{f}_n s_{\text{data}} + v, \tag{2}$$

where $v \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$ is the receiver noise at the vehicular UE.

B. Channel Model

A wideband geometric channel model with L clusters is adopted for our mmWave system. In this model, each of the clusters contributes a ray that has a time delay, $\tau_{n,\ell}$, and an AoA, $\theta_{n,\ell}$. Denotes the pulse-shaping function by $p(t)$, the delay-d channel vector between the user and nth BS can be expressed as

$$\mathbf{h}_{d,n} = \sqrt{\frac{M}{\rho}} \sum_{\ell=1}^{L} g_{n,\ell} p(dT_s - \tau_{n,\ell}) \mathbf{a}_n(\phi_\ell, \theta_\ell), \tag{3}$$

where ρ denotes the path loss between nth BS and the user. $g_{n,l}$ is the complex gain for the lth path. $\mathbf{a}_n(\theta_\ell, \phi_\ell)$ is the array response vector of the nth BS for the lth path and is defined as

$$\mathbf{a}_n(\phi_\ell, \theta_\ell) = \mathbf{a}_v(\theta_\ell) \otimes \mathbf{a}_h(\phi_\ell, \theta_\ell), \tag{4}$$

where $\mathbf{a}_v(\cdot)$ and $\mathbf{a}_h(\cdot)$ are the BS array response vectors in the vertical and horizontal directions, respectively. These are represented as

$$\mathbf{a}_v(\theta) = \frac{1}{\sqrt{M_v}} [1, e^{j\frac{2\pi}{\lambda} d \sin(\theta)}, \cdots, e^{j(M_V-1)\frac{2\pi}{\lambda} d \sin(\theta)}]^T, \tag{5}$$

$$\begin{aligned} \mathbf{a}_h(\phi, \theta) = \frac{1}{\sqrt{M_h}} [1, e^{j\frac{2\pi}{\lambda} d \sin(\phi)\cos(\theta)}, \cdots, \\ e^{j(M_H-1)\frac{2\pi}{\lambda} d \sin(\phi)\cos(\theta)}]^T. \end{aligned} \tag{6}$$

III. DEEP LEARNING-BASED CHANNEL ESTIMATION AND TRACKING

In this section, we present our deep learning-based mmWave channel estimation and tracking algorithm, as illustrated in Fig. 2. We first estimate the channel using DNN. Then, the channel is tracked by LSTM using the estimated channel that is outputted by the DNN.

A. DNN-based Channel Estimation

The key challenge in estimating and tracking a channel in highly mobile mmWave applications is the large training overhead in terms of time. This time-wise overhead is caused by the large number of antennas at the transmitters and receivers. Prior research in mmWave channel estimation and tracking repeated the estimation process each time the channel varied. In addition, the system did not utilize the previous observations of this estimation process. However, the channels are perceptibly functions of the various elements of the environment, including the transmitter/receiver locations and scatterer positions. The challenge is that these functions are difficult to characterize analytically as they generally involve many physical interactions and are unique to each environmental setup. Therefore, we leveraged the exceptional capability of deep learning models to learn this mapping function and to enable the prediction of an mmWave channel that can be conveniently estimated with low training overhead.

In [15], the authors demonstrated that when the uplink training pilots are received simultaneously by multiple distributed BSs using omni-directional antenna patterns, these omni-received signals draw a defining signature for the UE location and its interaction with the surrounding environment. This is highly noteworthy as no beam training is required to acquire these omni-received signals. This reduces the training overhead significantly. Inspired by this observation, we adopt a model in which the uplink training pilots are received only via omni-directional patterns. Furthermore, we train the deep learning model to learn the mapping between these omni-received signals and estimated channel. The omni-received signal at the nth BS can be expressed as

$$r_n^{\text{omni}} = \mathbf{f}_n^T \mathbf{h}_n s_{\text{pilot}} + \mathbf{f}_n^T \mathbf{v}_n, \tag{7}$$

where the beamforming vector is set to $\mathbf{f}_n = [1, 0, \cdots, 0]$, $\forall n$,

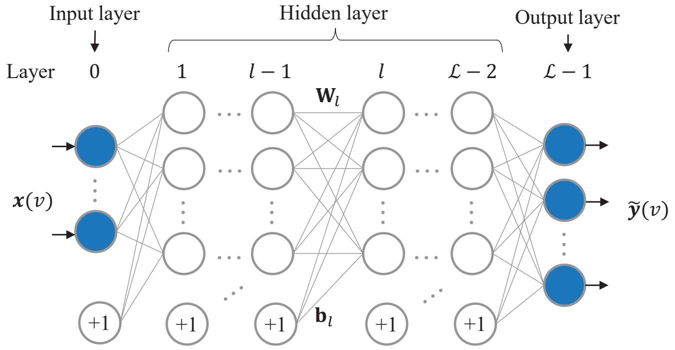

Fig. 3. Structure of the DNN. $\mathbf{W}_l$ is the $n_l \times n_{l-1}$ weight matrix associated with the $(l-1)$th and lth layers, and $\mathbf{c}_l$ is the bias vector for the lth layer. $\boldsymbol{x}(\nu)$ and $\boldsymbol{y}(\nu)$ represent the input and labels, respectively, at v.

i.e., by activating only the first receiving antenna element at each BS.

For the DNN-based channel estimation, we need to explain DNN structure. DNN is an artificial neural network with multiple hidden layers between the input and output layers [25]. Each hidden layer is equipped with multiple neurons. The output is the weighted sum of these neurons with a nonlinear function. The DNN is processed by activation to realize a recognition and representation operation. In general, the sigmoid function and rectified linear unit (ReLU) function are used almost universally in the nonlinear operation. These can be expressed as $f_S(x) = 1/(1+e^{-x})$ and $f_R(x) = \max\,(0, x)$, respectively. In the proposed DNN-based channel estimation, we choose no activation function for neurons in the output layer and ReLU functions for neurons in the rest of layers.

The proposed DNN architecture for channel estimation is illustrated in Fig. 3. We adopt a fully connected feedforward DNN with $\mathcal{L}$ layers: an input layer, $\mathcal{L}-2$ hidden layers, and an output layer. $\mathbf{W}_l$ is the $n_l \times n_{l-1}$ weight matrix associated with the $(l-1)$th and lth layers, and $\mathbf{b}_l$ is the bias vector for the lth layer. Because a single execution of the deep learning algorithm is based on a batch of data, we denote V and $v(0 \leq \nu \leq V-1)$ as the batch size and serial index, respectively. Let $\boldsymbol{x}(\nu)$ and $\boldsymbol{y}(\nu)$ represent the input and labels, respectively, of the DNN at v. The output of the DNN is the estimate of $\boldsymbol{y}(\nu)$, which can be mathematically expressed as

$$\tilde{\boldsymbol{y}}(\nu) = \boldsymbol{g}_{\mathcal{L}-1}\left(\cdots \boldsymbol{g}_1\left(\boldsymbol{x}(\nu); \boldsymbol{\theta}_1\right); \boldsymbol{\theta}_{\mathcal{L}-1}\right), \tag{8}$$

where $\boldsymbol{\theta}_\ell \triangleq \{\boldsymbol{W}_\ell, \boldsymbol{b}_\ell\}$ represents the parameters of the lth layer. The input number corresponds to the number of BSs. The omni-received signal, $\mathbf{r}^{\text{omni}}$, is the input feature and is defined as

$$\mathbf{r}^{\text{omni}} = [r_1^{\text{omni}}, \cdots, r_N^{\text{omni}}]. \tag{9}$$

It is collected from all the coordinating BSs. Thus, the estimated mmWave channel collected from all the coordinating BSs, $\mathbf{h} = [\mathbf{h}_1, \cdots, \mathbf{h}_N]$ can be obtained.

For simplicity, we define $\boldsymbol{\theta} \triangleq \{\boldsymbol{\theta}_\ell\}_{\ell=1}^{\mathcal{L}-1}$ as the set of parameters to be optimized. The optimal $\boldsymbol{\theta}$ can be obtained by minimizing the loss function Loss $(\boldsymbol{\theta})$ through training. Loss $(\boldsymbol{\theta})$ can be expressed as

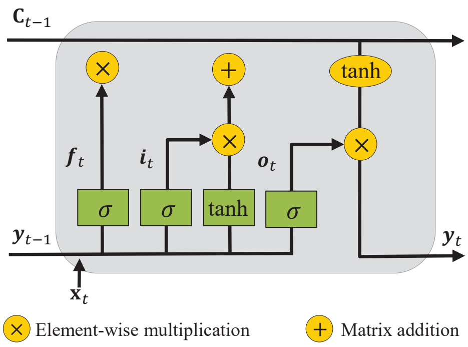

Fig. 4. Structure of the LSTM. The LSTM consists of the forget gate, $\mathbf{f}_t$, input gate, $\mathbf{i}_t$, output gate, $\mathbf{o}_t$, and memory cell, $\mathbf{c}_t$. $\mathbf{x}_t$ and $\mathbf{y}_t$ are the input and output of the current moment, respectively.

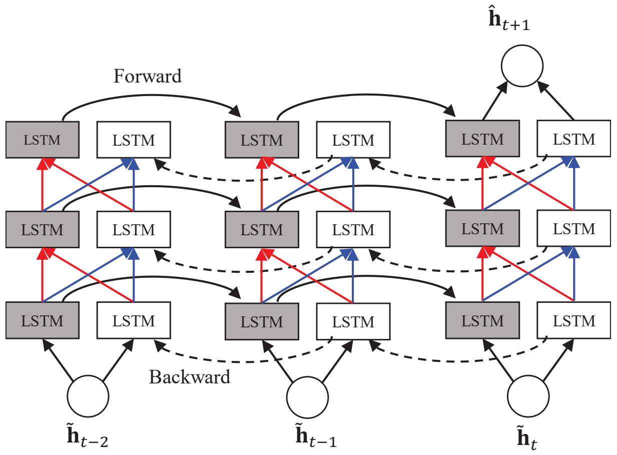

Fig. 5. Structure of the Bi-LSTM with three hidden layers and three time slots. The Bi-LSTM contains a forward LSTM layer and a backward LSTM layer.

$$\text{Loss}\,(\boldsymbol{\theta}) = \frac{1}{VL_y}\sum_{\nu=0}^{V-1}(\tilde{\boldsymbol{y}}(\nu) - \boldsymbol{y}(\nu))^2, \tag{10}$$

where L_y is the length of the vector $\boldsymbol{y}(\nu)$.

B. LSTM-based Channel Tracking

LSTM is an artificial recurrent neural network (RNN) architecture that effectively overcomes the vanishing gradient issue in a naively designed RNN [26]. The LSTM cell has the input layer, $\mathbf{x}_t$, and the output layer, $\mathbf{y}_t$, during time slot t. An LSTM is composed of a memory cell, an input gate, an output gate and a forget gate. The cell stores values over arbitrary time intervals. The three gates regulate the flow of information into and out of the cell. The architecture of the LSTM model is illustrated in Fig. 4. The forget gate, $\mathbf{f}_t$, input gate, $\mathbf{i}_t$, and output gate, $\mathbf{o}_t$, are calculated as

$$\mathbf{f}_t = \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f), \tag{11}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{12}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o). \tag{13}$$

Based on the results of the above equations, the cell state, $\mathbf{c}_t$, and output, $\mathbf{y}_t$, are updated by the following equation

$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \tanh(\mathbf{W}_{cx}\mathbf{d}_t + \mathbf{W}_{ch}\mathbf{y}_{t-1} + \mathbf{b}_c), \quad (14)$$

$$\mathbf{y}_t = \mathbf{o}_t \otimes \tanh(\mathbf{c}_t), \quad (15)$$

where $\mathbf{W}$ denotes a weight matrix and $\mathbf{b}$ denotes a bias vector.

A bidirectional LSTM (Bi-LSTM) has two hidden layers by forward and backward processes, which then feed forward to the same output layer [27]. The function of this hidden layer can be defined as follows [28]:

$$\mathbf{y}_t = \sigma(\underset{h}{\overset{t}{\rightarrow}} \oplus \underset{h}{\overset{t}{\leftarrow}}). \quad (16)$$

Note that notations $\rightarrow$ and $\leftarrow$ denote the forward and backward processes, respectively. Both the forward and backward layer outputs are calculated using the standard LSTM updating equations: Eqs. (11)–(15). The Bi-LSTM layer generates an output vector in which each element is calculated by Eq. (16).

For the channel tracking system considered by us, the sequence of the most recent T channel estimation results, $\tilde{\mathbf{h}}_{t-T+1}, \cdots, \tilde{\mathbf{h}}_t$, is the input of the Bi-LSTM. Furthermore, the next time slot-estimated channel $\tilde{\mathbf{h}}_{t+1}$ is the desired output, which correspond to $\mathbf{x}_t$ and the desired output $\mathbf{y}_t$, respectively, in the Bi-LSTM model. In the Bi-LSTM training procedure, the prediction results are improved continuously based on advanced memory, by discarding some of the ineffective information from the past. The predicted channel vector $\hat{\mathbf{h}}_{(t+1)}$ after the training is the output of the Bi-LSTM. The difference between this vector and the actual channel vector at the next time $\tilde{\mathbf{h}}_{(t+1)}$ is negligible. Fig. 5 illustrates an example of the Bi-LSTM structure with three hidden layers and three time slots for the estimated channel sequence.

C. Implementation of the deep learning-based Channel Estimation and Tracking

The proposed deep learning-based algorithm has two phases: the deep learning training phase and deployment phase.

- Deep learning training phase: In this phase, the mmWave channel is estimated using the omni-received signal and a conventional algorithm such as an exhaustive beam training or compressed sensing. Then, a new data point $\mathbf{r}^{\text{omni}}$ is added to the deep learning dataset. We collect a large number of data points and then use this dataset to train the deep learning model. This is described in detail in Section III-A.
- Deep learning deployment phase: Once the deep learning model is trained, the BS uses it to directly estimate and track the mmWave channel using the omni-received signal. More specifically, this phase requires the user to send only an uplink pilot to estimate the mmWave channels. This channel is passed to the deep learning model. This saves the training overhead associated with the mmWave exhaustive beam training or compressed sensing process.

It is important to note that the dataset collection process and deep learning training are performed without affecting the classical mmWave system operation. Hence, it is feasible to collect a large dataset for capturing the dynamics in the environment because it does not interfere with the classical system operation.

Table 1. The Deep MIMO dataset parameters.

Parameter	Values
Carrier frequency	60 GHz
System bandwidth	500 MHz
Active BS	5, 6, 7, 8
Active users	From row R1100 to R2000
Number of BS antennas	$M_x = 1, M_y = 8, M_z = 4$
Number of user antennas	$M_x = 1, M_y = 1, M_z = 1$
Antenna spacing (in wavelength)	0.5
Number of paths	5

D. Complexity Analysis

In this subsection, we analyze the computational complexity of the proposed deep learning-based channel estimation and tracking approach in the testing stage. For the proposed approach, the computational complexity originates from the DNN processing for channel estimation and the LSTM processing for channel tracking. Since the fully connected layer for channel estimation is implemented as a matrix-matrix multiplication and addition, the computational complexity of the DNN for channel estimation, ignoring the biases, is $C_{\text{DNN}} \sim \mathrm{O}\left(\sum_{l=1}^{\mathcal{L}-1} n_{l-1}n_l\right)$. The total number of parameters N in the of LSTM for channel tracking with one cell, ignoring the biases, can be calculated as follows:

$$N = n_c \times n_c \times 4 + n_i \times n_c \times 4 + n_c \times n_o + n_c \times 3, \quad (17)$$

where n_c is the number of memory cell, n_i is the number of input units, and n_o is the number of output units. The learning time for a network with a relatively small number of inputs is dominated by the the $n_c \times (n_c + n_o)$ factor [29]. Therefore, the complexity of the deep learning-based algorithm can be expressed as

$$C_{\text{DL-based}} \sim \mathrm{O}\left(\sum_{\ell=1}^{\mathcal{L}-1} n_{\ell-1}n_\ell\right) + \mathrm{O}\left(n_c \times (n_c + n_o)\right). \quad (18)$$

IV. SIMULATION RESULTS

In this section, we describe the simulation setup in detail (including the channel models and dataset generation) and present the simulation results.

A. Simulation Setup

The simulation setup was based on the publicly-available generic DeepMIMO [30] dataset with the parameters described in Table 1. These parameters are constructed using the 3D ray-tracing software Wireless Insite [31], which captures the channel dependence on the frequency. The development of the system model and channel model is described in Section II. The channel vector was constructed by using parameters such as AoA, AoD, and path loss. More specifically, we set the frequency of the mmWave at 60 GHz. In addition, the four BSs were distributed on top of a building with a height of 50 m. Each BS was equipped with a UPA antenna with $M = 8 \times 4$ antennas. The user was equipped with one antenna. To predict the channel vector of vehicular mobile users, we constructed a few random routes with moving rates ranging from 10 m/s to 30 m/s.

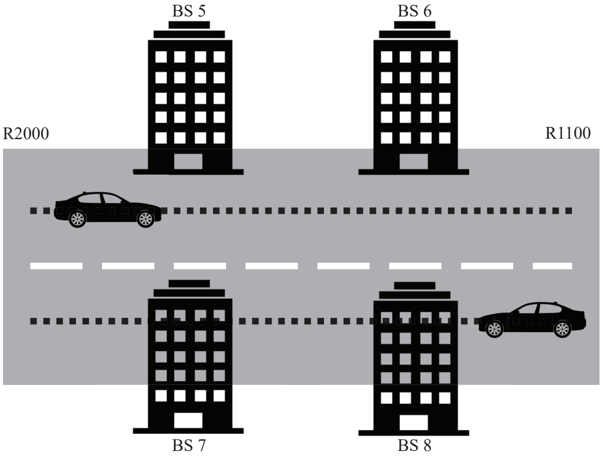

Fig. 6. Simulation environment. The users are the black dots, which are randomly distributed to simulate the movement of vehicle. The four BSs are coordinately designed and built on different buildings.

Table 2. DNN training hyper-parameters.

Parameter	Values
Optimizer	Adam
Learning rate	0.001
Dropout	0.9
Regularization	l_2
Max. number of epochs	100
Data size	50,000
Dataset split	80:20

The specific simulation environment is illustrated in Fig. 6. The figure shows that the four BSs were placed on different buildings. They covered all the user's movements. The location of the vehicular UE is selected randomly from a uniform x–y grid of candidate locations. For the channel tracking, the dots represent the movement of the vehicular UE. Two tracks are apparent in the figure. Each BS received an omni-directional signal, as described in Section III, which was sent to the same cloud as the dataset for the deep learning model. In the cloud, the omni-directional signals of all the BSs were combined, and the final input was $\mathbf{r}^{\text{omni}}$. The nth BS was equipped with a UPA antenna array with $M = 32$ antennas. Therefore, with $N = 4$ BSs serving the same user simultaneously, the dimension of the integrated omni-directional signal $\mathbf{r}^{\text{omni}}$ equaled 128×1. Before training the neural network, $\mathbf{r}^{\text{omni}}$ was normalized by the maximum and minimum values of the vector. In the deep learning simulation, we adopt the DNN described in Section III-A, with $\mathcal{L}= 3, 4, \cdots, 7$ and $n_l = 2048$ neurons per layer. This DNN is trained using the datasets for the channel estimation. The other hyper-parameters are summarized in Table 2. In the LSTM, the learning rate is set to 0.001, and the batch size is 30. We construct our DNN and LSTM network in Keras [32] with a tensorFlow [33] backend. The rest of the simulation is implemented on MATLAB.

B. Simulation Results

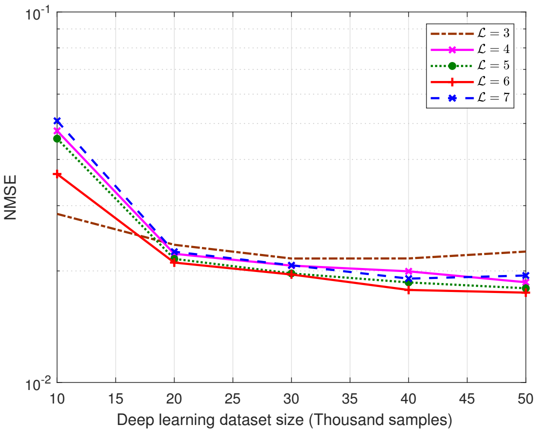

Fig. 7. The NMSE performance of the DNN-based channel estimation. This shows that the performance first improves and then degrades as the number of layers $\mathcal{L}$ increases. The optimal number of layers is $\mathcal{L} = 6$, which is the default value of $\mathcal{L}$ in our simulations.

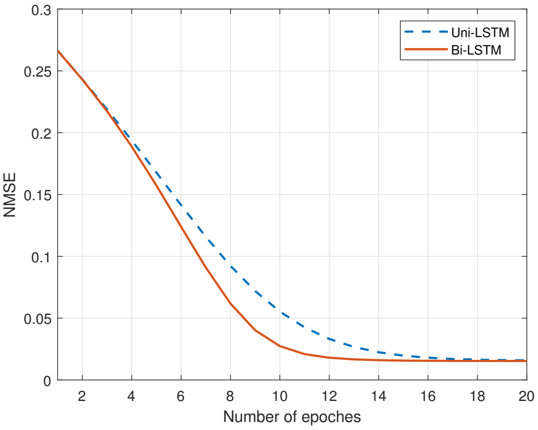

Fig. 8. The NMSE performance of the Uni-LSTM and Bi-LSTM with three hidden layer and three time slots. The results show that Bi-LSTM causes the LSTM to converge faster.

We adopt the NMSE to test the difference between the estimated channel vector and predicted channel vector and thereby evaluated the performance of the proposed machine learning system. It is defined as

$$\text{NMSE} = \frac{\|\hat{\mathbf{h}} - \tilde{\mathbf{h}}\|^2}{\|\tilde{\mathbf{h}}\|^2}, \tag{19}$$

where $\hat{\mathbf{h}}$ is the predicted channel vector and $\tilde{\mathbf{h}}$ is the actual channel vector.

In Fig. 7, we investigate the performance of the DNN-based channel estimation with different number of layers. The performance first improves and then degrades as the number of layers $\mathcal{L}$ increases. Fig. 6 shows that the optimal number of layers is $\mathcal{L}$=6, which is the default value of $\mathcal{L}$ in our simulations. Theoretically, the learning capability of the DNN improves as the number of layers increases. Owing to the vanishing gradient and pathology degradation, the training of the DNN becomes more challenging as the network deepens [34].

Fig. 8 illustrates our investigation of the performance of the LSTM-based channel tracking with a different type of LSTM (with three hidden layers and three time slots) for the

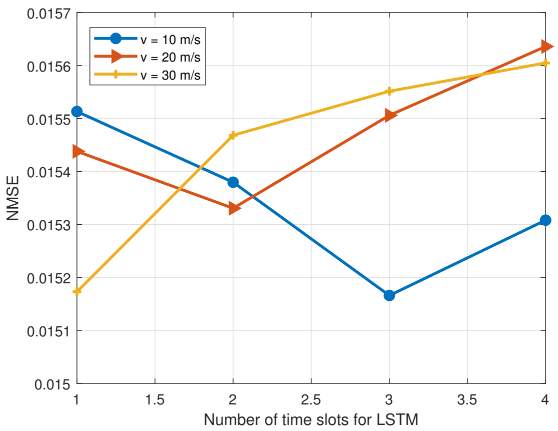

Fig. 9. The NMSE performance of Bi-LSTM with different number of time slots. This figure shows that the time slots of the Bi-LSTM input shorten as the vehicular UE speeds up. Furthermore, the performance degrades as the time slot of Bi-LSTM increases, even for high vehicular UE speed.

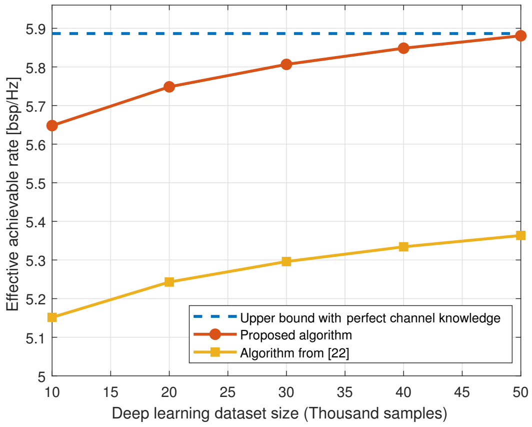

Fig. 10. The effective achievable rate performance. This clearly illustrates the capability of the proposed deep learning-based algorithm in supporting highly-mobile mmWave applications with negligible training overhead.

estimated channel sequence. Based on Fig. 8, we adopt Bi-LSTM rather than unidirectional LSTM (Uni-LSTM) because Bi-LSTM causes the LSTM to converge faster.

Fig. 9 shows the NMSE performance of Bi-LSTM with different numbers of time slots. The time slots of the Bi-LSTM input shorten as the vehicular UE speeds up. Furthermore, the performance degrades as the time slot of Bi-LSTM increases, even for high vehicular UE speed. This is because the estimated channel was outdated and the long-predictions inaccurate. Based on Fig. 9, we adopt the numbers of time slots according to the vehicular environment. For example, Bi-LSTM adopts one time slot in a high-speed environment such as a freeway, and three time slots in a dense urban environment.

To demonstrate that our algorithm can reduce the pilot overhead, we introduce the beam coherence time and effective achievable rate, which is a recent concept in mmWave communications to represent the average beam training time [15]. The effective achievable rate can be characterized as

$$R_{\text{eff}} = \left(1 - \frac{N_{\text{tr}} T_{\text{p}}}{T_{\text{B}}}\right) \log_2 \left(1 + \frac{\left|\sum_{n=1}^{N} \tilde{\mathbf{h}}_n^T \mathbf{f_n}\right|^2}{\sigma^2}\right), \tag{20}$$

where N_{tr}, T_{p}, and T_{B} are the number of training pilot, beam training pilot sequence time, and beam coherence time, respectively. We compare our algorithm to a prior work [22]. The algorithm in [22] estimates the channel vectors using the traditional method and then designs the beamformer in the first beam coherence time. Rather than estimating the channel vectors, the BSs design the beamformer using our proposed system in the second beam coherence time. They reduced the overhead of two beam coherence times to half of the original value. Fig. 10 shows the achievable rate. The algorithm in [22], which incurs a higher overhead, has a lower effective achievable rate than that of our algorithm. When the number of training pilots is increased, the performance difference increases. This clearly illustrates the capability of the proposed deep learning-based algorithm in supporting highly-mobile mmWave applications with negligible training overhead.

V. CONCLUSION

In this study, we proposed a novel method integrating deep learning and channel estimation/tracking, and develop its deep learning modeling for vehicular mmWave communications. More specifically, a DNN was leveraged to learn the mapping function between an omni-beam pattern and mmWave channel, with negligible overhead. Following the channel estimation, Bi-LSTM was leveraged to track the channel. Bi-LSTM employes the past channel to promote the prediction of the user's channel. We use accurate 3D ray-tracing to analyze a performance of the proposed deep learning algorithm of massive MIMO in vehicular communications. The simulation results demonstrated that the proposed algorithm estimated and tracked the mmWave channel efficiently, incurring a negligible training overhead.

REFERENCES

[1] M. Xiao *et al.*, "Millimeter wave communications for future mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1909–1935, Sept. 2017.

[2] S. K. Yong and C.-C. Chong, "An overview of multigigabit wireless through millimeter wave technology: Potentials and technical challenges," *EURASIP J. Wireless Commun. Netw.*, vol. 2007, no. 1, p. 78907, Dec. 2006.

[3] M. Marcus and B. Pattan, "Millimeter wave propagation: Spectrum management implications," *IEEE Microw. Mag.*, vol. 6, no. 2, pp. 54–62, June 2005.

[4] T. S. Rappaport *et al.*, "Overview of millimeter wave communications for fifth-generation (5g) wireless networks—with a focus on propagation models," *IEEE Trans. Antennas Propag.*, vol. 65, no. 12, pp. 6213–6230, Dec. 2017.

[5] R. W. Heath, N. Gonzlez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.

[6] O. El Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. Heath, "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499–1513, Mar. 2014.

[7] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic, and R. Heath, "MIMO precoding and combining solutions for millimeter-wave systems," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 122–131, Dec. 2014.

[8] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 4: Enhancements for Very High Throughput in the 60 GHz Band, IEEE Standard P802.11ad/D9.0, 2012.

[9] J. Wang *et al*., "Beam codebook based beamforming protocol for multi-Gbps millimeter-wave WPAN systems," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1390–1399, Oct. 2009.

[10] Overview of NR Initial Access, document R1-1611272, TSG RAN WG1 Meeting #87, 3GPP, Nov. 2016.

[11] C. R. Berger, Z. Wang, J. Huang and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, Nov. 2010.

[12] L. Dai and X. Gao, "Priori-aided channel tracking for millimeter-wave beamspace massive MIMO systems," in *Proc. IEEE URSI AP-RASC*, 2016, pp. 1493–149.

[13] X. Gao, L. Dai, T. Xie, X. Dai, and Z. Wang, "Fast channel tracking for terahertz beamspace massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 5689–5696, July 2017.

[14] Y. Zhou, P. C. Yip, and H. Leung, "Tracking the direction-of-arrival of multiple moving targets by passive arrays: Asymptotic performance analysis," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2644–2654, Oct. 1999.

[15] A. Alkhateeb *et al*., "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, June 2018.

[16] M. S. Sim, Y. Lim, S. H. Park, L. Dai and C. Chae, "Deep learning-based mmWave beam selection for 5G NR/6G with sub-6 GHz channel information: Algorithms and prototype validation," *IEEE Access*, vol. 8, pp. 51634–51646, Mar. 2020.

[17] Y.G. Lim, Y.J. Cho, Y. Kim, C.B. Chae and R. Valenzuela, "Map-based millimeter-wave channel models: An overview, data for B5G evaluation and machine learning," *IEEE Wireless Commun.*, 2020.

[18] C. Lee, H. B. Yilmaz, C. Chae, N. Farsad and A. Goldsmith, "Machine learning based channel modeling for molecular MIMO communications," in *Proc. IEEE SPAWC*, 2017, pp. 1–5.

[19] H. Ye, G. Y. Li and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[20] H. Huang, J. Yang, H. Huang, Y. Song and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sept. 2018.

[21] C. Wen, W. Shih and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.

[22] Y. Guo, Z. Wang, M. Li and Q. Liu, "Machine learning based mmWave channel tracking in vehicular scenario," in *Proc. IEEE ICC Wksps.*, 2019, pp. 1–6.

[23] G. R. MacCartney, T. S. Rappaport, and A. Ghosh, "Base station diversity propagation measurements at 73 GHz millimeter-wave for 5G coordinated multipoint (CoMP) analysis," in *Proc. IEEE Globecom Wksps.*, 2017.

[24] D. Maamari, N. Devroye, and D. Tuninetti, "Coverage in mmWave cellular networks with base station co-operation," *IEEE Trans. on Wireless Commun.*, vol. 15, no. 4, pp. 2981–2994, Apr. 2016.

[25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, Jan. 2015.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–80, Dec. 1997.

[27] A. Graves, J. Schmidnuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5-6, pp. 602-610, 2005.

[28] M. Schuster, K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673-2681, 1997.

[29] W. Pedrycz and S.M. Chen, "Deep Learning: Concepts and Architectures," *Springer Nature*, 2019.

[30] A. Alkhateeb, "DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications," in *Proc. IEEE ITA*, Feb. 2019.

[31] Remcom, Wireless insite, [Online]. Available: http://www.remcom.com/wireless-insite.

[32] F. R. F. Branchaud-Charron and T. Lee, Accessed: Jun. 27, 2018. [Online]. Available: https://github.com/keras-team.

[33] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," arXiv preprint arXiv:1609.08144, 2016.

[34] D. Duvenaud, O. Rippel, R. Adams, and Z. Ghahramani, "Avoiding pathologies in very deep networks," in *Proc. AIS-TATS*, 2014.

Sangmi Moon received the B.S., M.S., and Ph.D. degrees in Electronics & Computer Engineering from Chonnam National University, Gwangju, Korea, in 2012, 2014, and 2017 respectively. She was a Visiting Scholar in the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, from September 2017 to February 2019. She is currently a Postdoctoral Researcher at Chonnam National University, Gwangju, Korea from 2018. Her research interests include 3D-MIMO, 3D-Beamforming, V2X, deep learning, and Next generation wireless communication systems.

Hyeonsung Kim received his B.S. degree in Electronics & Computer Engineering from Chonnam National University, Gwangju, Korea in 2020. He is currently a Master's Student in the Department of Electronics & Computer Engineering at Chonnam National University, Gwangju, Korea from 2020. His research interests include mobile and next generation wireless communication systems; MIMO, OFDM and deep learning.

Intae Hwang received a B.S. degree in Electronics Engineering from Chonnam National University, Gwangju, Korea in 1990 and a M.S. degree in Electronics Engineering from Yonsei University, Seoul, Korea in 1992, and a Ph.D. degree in Electrical & Electronics Engineering from Yonsei University, Seoul, Korea in 2004. He was a Senior Engineer at LG Electronics from 1992 to 2005. He is currently a Professor in the Department of Electronic Engineering at Chonnam National University, Gwangju, Korea from 2006. His current research activities are in digital & wireless communication systems, mobile terminal system for next generation applications; physical layer software for mobile terminals, efficient algorithms for MIMO-OFDM, Relay, ICIM, CoMP, D2D, SCE, MTC, V2X, IoE, and NR MIMO schemes for wireless communication.