

Joint Timing and Frequency Synchronization Using Convolutional Neural Network in WLAN Systems

Eui-Soo Lee¹, Eui-Rim Jeong^{*2}

¹MS student, Department of Mobile Convergence and Engineering, Hanbat National University, Daejeon, 34158, Republic of Korea

^{*2}Professor, Department of Information and Communication Engineering, Hanbat National University, Daejeon, 34158, Republic of Korea

^{*}Corresponding author. E-mail: erjeong@hanbat.ac.kr

Article History: Received: 11 November 2020; Accepted: 27 December 2020; Published online: 05 April 2021

Abstract : In wireless communication systems, the performance of the receiver is very sensitive to time and frequency offsets. In particular, orthogonal frequency division multiplexing (OFDM) systems are highly vulnerable to those offsets due to inter-carrier interference (ICI) and inter-symbol interference (ISI). To solve this problem, wireless local area network (WLAN) systems transmit a preamble for synchronization. In this paper, we propose a joint time and frequency offsets estimation technique based on convolutional neural network (CNN) for WLAN systems. In the proposed technique, the correlation between the received signal and the transmitted preamble is performed first. Then the frequency offset is coarsely compensated by several hypothesized offsets. The compensated signals are inputted to the proposed CNN and the CNN predicts the time and frequency offsets. The estimation performance is examined through computer simulation. According to the results, the proposed time offset estimator shows 3 dB to 6 dB performance gain, and the frequency offset estimator shows much lower root mean square error (RMSE) performance than the conventional technique at low SNRs.

Keywords: WLAN, OFDM, Timing offset, Carrier frequency offset, Convolutional neural network, Synchronization.

1. Introduction

Orthogonal Frequency Division Multiplexing (OFDM)-based communication systems such as Wireless LAN (WLAN) are very sensitive to synchronization errors such as timing and frequency offsets. Due to the clock offset between transmitter and receiver, and the propagation delay due to the distance between them, the exact arrival time of the packet is not known at the receiver. In addition, carrier sense multiple access (CSMA) systems have a random transmission time. In this case, the timing offset (TO) is random and should be estimated at the receiver. Especially, in OFDM systems, the receiver must find the start position of the OFDM symbol to avoid inter OFDM symbol interference. Carrier frequency offset (CFO) is the frequency difference between the carriers of the transmitter and the receiver. CFO is due to the oscillator tolerance and Doppler frequency. If synchronization is not perfect, the orthogonality of OFDM is broken, and accordingly, inter-symbol interference (ISI) or inter-carrier Interference (ICI) may happen [1]. Hence, synchronization is a crucial task of the receiver. Many studies have been conducted for synchronization [2-7]. In the IEEE 802.11 standard, short training field (*STF*) and long training field (*LTF*) are transmitted ahead of the packet for receiver synchronization. Conventional synchronization such as [2] and [3] proceed in the following procedure. First, the *STF* is used to detect whether a valid packet has arrived, and coarse CFO is compensated. Next, fine TO and CFO correction is performed using *LTF*.

In this paper, we propose a new synchronization technique using convolutional neural network (CNN) for WLAN systems. CNN was first introduced by LeNet-5 as one of the deep learning techniques and has recently been widely used in image processing tasks [8-10]. First, the proposed technique compensates the CFO at a specific frequency interval within the maximum possible CFO range allowed in the standard. Since multiple CFO candidates are compensated with the received signal, multiple signal branches are made after CFO compensation. After that, we perform cross-correlation between each compensated signal and *STF*. The multiple correlation outputs are stacked to form a two-dimensional matrix. This matrix is the input of the proposed CNN and the CNN predicts both the TO and CFO.

We compare the performance of the proposed technique and the conventional technique through computer simulation. In the simulation, the channel environments defined in the standard are used. According to the results, the proposed technique shows 3 dB ~ 6 dB gain for false detection probability (FDP) performance than the conventional technique in TO estimation, and the proposed CFO estimator shows better root mean square errors (RMSEs) than the conventional technique at low to medium signal to noise ratios (SNRs).

^{*}Corresponding author: Eui-Rim Jeong

Professor, Department of Information and Communication Engineering, Hanbat National University, Daejeon, 34158, Republic of Korea

E-mail: erjeong@hanbat.ac.kr

2. System model

In WLAN standards (IEEE 802.11ac or IEEE 802.11n), a legacy preamble precedes the transmission packet for synchronization and backward compatibility. The legacy preamble consists of STF repeated 10 times, LTF repeated 2 times, and signal field as shown in Fig. 1.

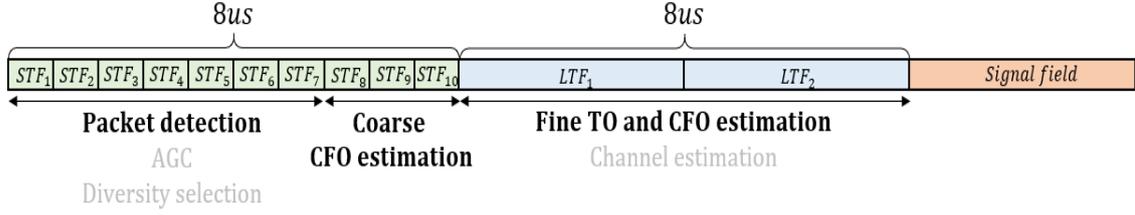


Fig. 1. Legacy preamble structure

In [2] and [3], the packet is detected using $STF_{1\sim7}$, the CFO is coarsely estimated using $STF_{8\sim10}$, and fine TO and CFO estimations are done using $LTF_{1\sim2}$. The synchronization procedure is shown in Fig. 2.

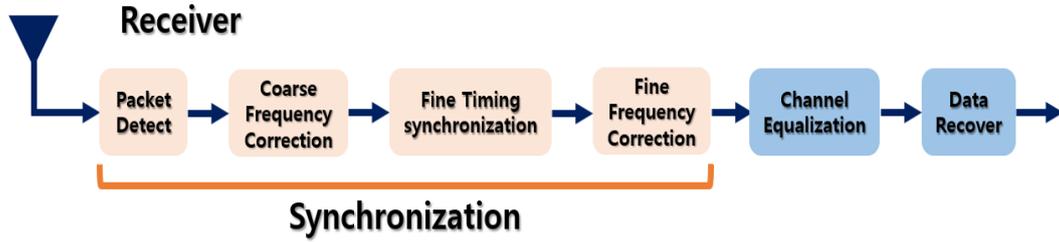


Fig. 2. WLAN synchronization flow chart

In this paper, it is assumed that a packet arrival is detected using the time $STF_{1\sim7}$ in advance. The problem to solve here is to estimate the TO and CFO afterward.

2.1 Conventional CFO estimation technique

In the conventional technique, CFO is estimated using both STF and LTF . The CFO was estimated by using the autocorrelation of the received repetitive signal, and in this case, the CFO can be estimated without being affected even when the TO and multipath channels exist. The following equation is the conventional CFO estimation technique.

$$r(n) = s(n)e^{j2\pi(f_{tx}-f_{rx})nT_s} = s(n)e^{j2\pi f_{\Delta}nT_s} \quad (1)$$

$$z(n) = \sum_{n=0}^{L-1} s(n)e^{j2\pi f_{\Delta}nT_s} (s(n+D)e^{j2\pi f_{\Delta}(n+D)T_s})^* = e^{-j2\pi f_{\Delta}DT_s} \sum_{n=0}^{L-1} |s(n)|^2 \quad (2)$$

$$\hat{f}_{\Delta} = -\frac{1}{2\pi DT_s} \arg(z) \quad (3)$$

where s is transmitter signal, f_{tx} is carrier frequency of the transmitter, f_{rx} is carrier frequency of the receiver, f_{Δ} is the CFO, r is the received signal, L is the training signal length, D is the repetition period of the training field, z is autocorrelation output, and \arg is phase angle.

2.2 Conventional TO estimation technique

In the conventional technique, TO is estimated using LTF . TO is estimated by using the cross-correlation between the received signal and the LTF. Autocorrelation technique can be used as well, but the cross-correlation shows superior performance to the autocorrelation [11]. The following equation is a conventional cross-correlation based TO estimator.

$$\hat{d}_{\Delta} = \underset{n}{\operatorname{argmax}} |\sum_{k=0}^{L-1} r(n+k)t(k)^*|^2 \quad (4)$$

where $t(k)$ is LTF and d_{Δ} is the TO.

2.3 System parameter

The parameters of WLAN systems used in this paper are shown in Table 1.

Table 1. Parameters of WLAN system

Parameter	Details
Frequency band	5.3 GHz
Bandwidth	40 MHz
FFT size	128

The carrier frequency is 5.3 GHz, the bandwidth is 40 MHz, and the FFT point is 128. At this time, the length of STF is 32 samples and the length of LTF is 160 samples. Therefore, the range of packet detection is in $STF_1 \sim STF_7$, that is, 1~224 samples, so it is necessary to find the exact arrival time of packet. In addition, the oscillator precision tolerance is specified to be less than ± 20 Parts Per Million (ppm), so that CFO is in the range from -40 ppm to +40 ppm. when the carrier frequency is 5.3 GHz, the range of CFO is -212 kHz to 212 kHz.

3. Proposed technique

In this paper, we propose a deep learning based TO and CFO estimation technique using only STF , assuming that packet arrival is already detected in advance. First, the received signal is compensated with a constant frequency interval. If the frequency interval is f_0 , the number of branches is K , and $f_0 \times K = 212$ kHz. The equation is as follows.

$$x_i(n) = r e^{j2\pi(-212 \text{ kHz} + if_0)nT_s} \text{ for } i = 0, \dots, K - 1 \quad (5)$$

where $x_i(n)$ is the i -th CFO compensated signal. We set f_0 at 42.4 kHz. Next, cross-correlation between x_i and $STF_{1 \sim 10}$ are performed. The equation is as follows.

$$y_i(n) = \sum_{k=0}^{M-1} x_i(n+k)s(k)^* \quad (6)$$

$$y_i = [y_i(0), \dots, y_i(B-1)] \quad (7)$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_{K-1} \end{bmatrix} \quad (8)$$

where $s(k)$ is STF , M is length of the STF , $y_i(n)$ is the correlator output, B is the window size of the correlator, and \mathbf{Y} is a matrix that collects the correlator output of each branch. B is fixed to 864 in this paper because the size that can express the correlation characteristics of all STF is 640 and the maximum TO is 224. Overall structure is as shown in the following figure.

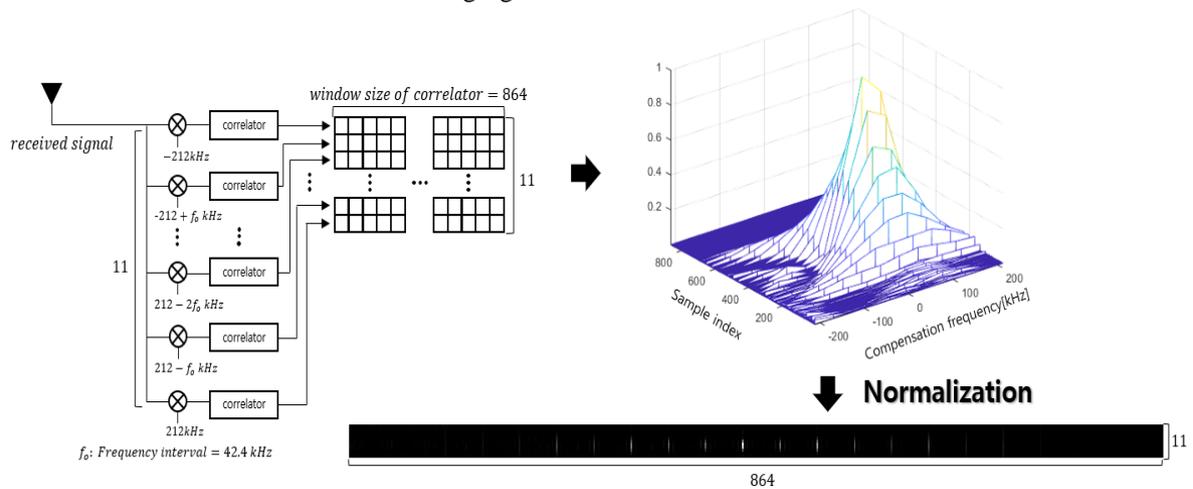


Fig. 3. Procedure for generation of proposed CNN input

In other words, First the multiple CFOs are compensated, and then the cross-correlation is conducted at each branch. Total 864 correlator output is collected. So, the final signal becomes 11 by 864 matrix. This matrix performs normalization and is used as input to two proposed CNNs. As shown in Fig. 4, two CNNs are designed for estimation of TO and CFO. The two CNNs predict the TO and CFO, respectively from the same 11 x 864

input. Table 2 summarizes the parameters of the proposed CNNs.

Table 2. The parameters of the proposed CNNs

CNN of TO estimation			CNN of CFO estimation		
Input Layer	Input size	$11 \times 864 \times 1$	Input Layer	Input size	$11 \times 864 \times 1$
Conv. Layer 1	Num. of filters	32	Conv. Layer 1	Num. of filters	32
	Filter size	1×49		Filter size	1×49
	Activation function	ReLU		Activation function	ReLU
	Max Pooling	size: 2×2 stride: 2, 2		Max Pooling	size: 2×2 stride: 2, 2
	Num. of parameters	1,600		Num. of parameters	1,600
Conv. Layer 2	Num. of filters	64	Conv. Layer 2	Num. of filters	64
	Filter size	1×49		Filter size	1×49
	Activation function	ReLU		Activation function	ReLU
	Max Pooling	size: 2×2 stride: 2, 2		Max Pooling	size: 2×2 stride: 2, 2
	Num. of parameters	3,200		Num. of parameters	3,200
Conv. Layer 3	Num. of filters	128	Conv. Layer 3	Num. of filters	128
	Filter size	1×49		Filter size	1×49
	Activation function	ReLU		Activation function	ReLU
	Max Pooling	size: 2×2 stride: 2, 2		Max Pooling	size: 2×2 stride: 2, 2
	Num. of parameters	6,400		Num. of parameters	6,400
Conv. Layer 4	Num. of filters	256	Conv. Layer 4	Num. of filters	256
	Filter size	1×49		Filter size	1×49
	Activation function	ReLU		Activation function	ReLU
	Max Pooling	size: 2×2 stride: 2, 2		Max Pooling	size: 2×2 stride: 2, 2
	Num. of parameters	12,800		Num. of parameters	12,800
Conv. Layer 5	Num. of filters	512	Conv. Layer 5	Num. of filters	512
	Filter size and stride	1×49		Filter size and stride	1×49
	Activation function	ReLU		Activation function	ReLU
	Max Pooling	size: 2×2 stride: 2, 2		Max Pooling	size: 2×2 stride: 2, 2
	Num. of parameters	256,000		Num. of parameters	256,000
Skip-connection 1	Num. of filters	128	Skip-connection 1	Num. of filters	128
	Filter size and stride	$1 \times 1, (2, 2)$		Filter size and stride	$1 \times 1, (2, 2)$
	Activation function	ReLU		Activation function	ReLU
	Num. of parameters	4,224		Num. of parameters	4,224
Skip-connection 2	Num. of filters	512	Skip-connection 2	Num. of filters	512
	Filter size and stride	$1 \times 1, (2, 2)$		Filter size and stride	$1 \times 1, (2, 2)$
	Activation function	ReLU		Activation function	ReLU
	Num. of parameters	66,048		Num. of parameters	66,048
Fully-connected layer	Output size	224	Fully-connected layer	Output size	1
	Activation function	Soft-max		Activation function	.
	Num. of parameters	3,096,576		Num. of parameters	13,824
Num. of total parameters		3,446,848	Num. of total parameters		364,096

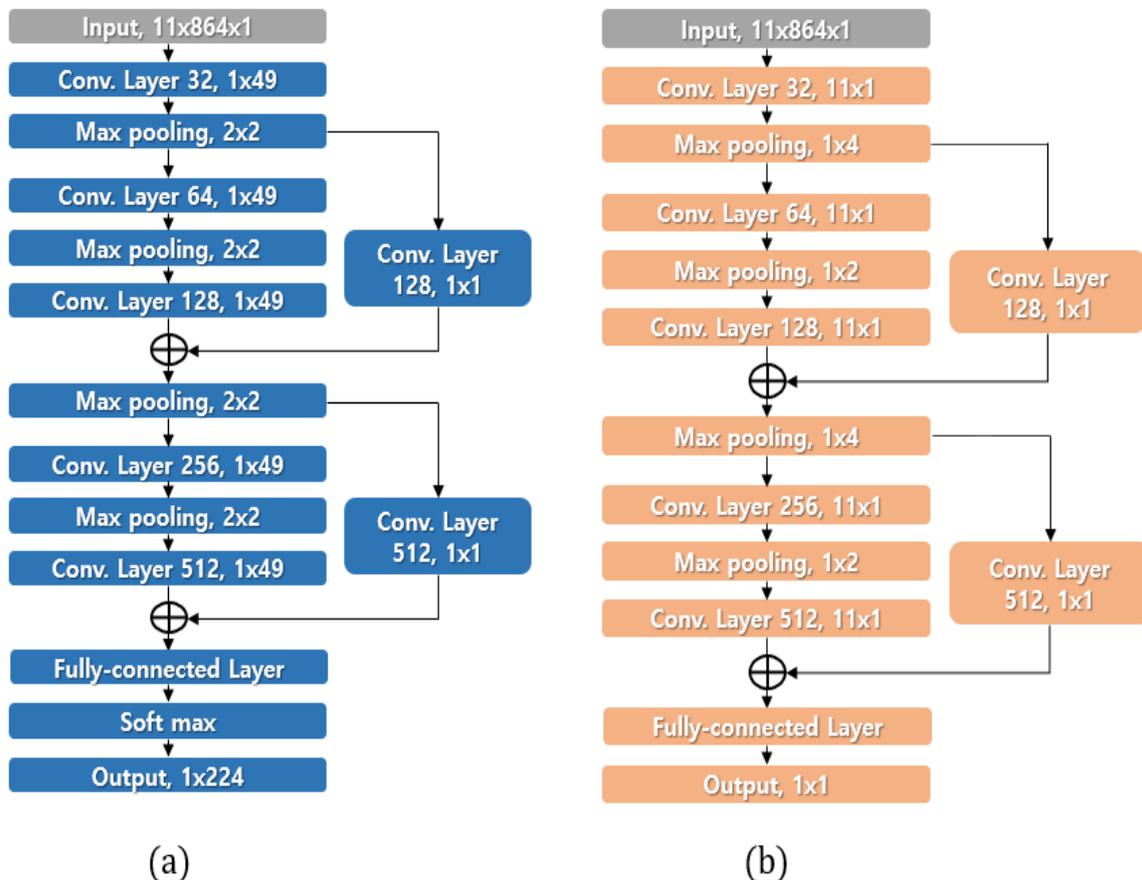


Fig. 4. Proposed CNN structures for (a) TO estimation and (b) CFO estimation

Both the networks consist of 5 convolutional layers, 2 skip-connections, and one fully-connected layer. The TO estimator is a classification network and the CFO estimator is a regression network.

4. Simulation result

Computer simulation was performed to compare the performance of the proposed technique and the conventional technique. The channel environment used in the simulation used the channel models in the standard [12]. The following table is the parameters of the channel models. Model A is AWGN channel, Model C and Model E are multipath fading channels.

Table 3. Parameters of channel models

Model	Environment	1 st break point	RMS Spreading delay	Max delay	# tap	# cluster	Doppler
A	AWGN	-	0nsec	0nsec	1	-	1.2km/h all tap
C	Small office	5m	30nsec	200nsec	14	2	1.2km/h all tap
E	Indoor hotspot	20m	100nsec	730nsec	18	4	1.2km/h all tap

4.1 CNN training

To train the proposed CNN, 100,000 training sets were used. At this time, TO and CFO were randomly generated, and SNR was also randomly selected in the range of -10dB to 30dB. The Table 4 summarizes the optimization techniques for TO and CFO training.

Table 4. Optimization techniques for training

CNN of TO estimation		CNN of CFO estimation	
Optimization algorithm	Adam	Optimization algorithm	Adam
Learning rate	0.001	Learning rate	0.001

Epoch	20	Epoch	80
Cost function	Cross-entropy	Cost function	Mean square error

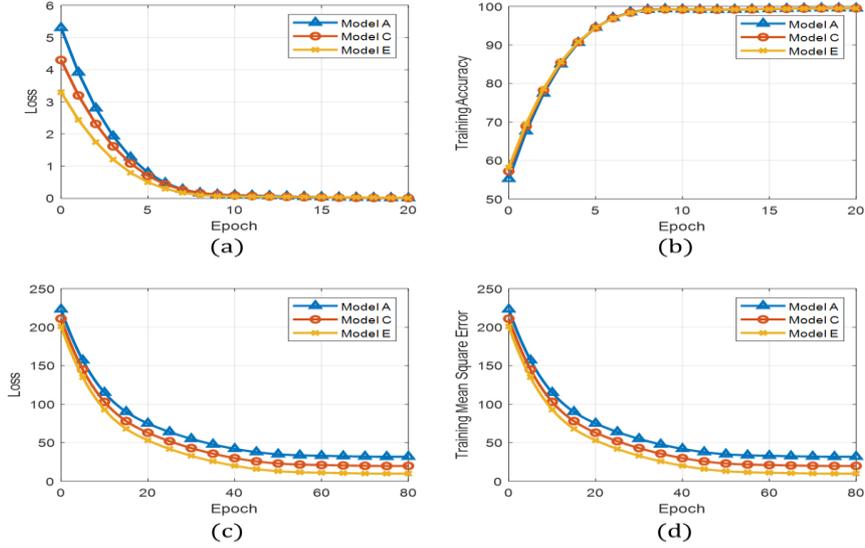


Fig. 5. Learning curves for TO CNN((a), (b)) and for CFO CNN((c), (d))

4.2 TO estimation performance

To verify the performance of the TO estimation CNN, 50,000 test sets were generated each at 1dB interval in the range of -10dB to 15dB SNRs. The performance evaluation measure is false detection probability. For channel Model A, correct TO estimation means the exact TO for the first STF arrival. For Model C and Model E, the channel has long delay spreads by the channels, and there are some ambiguity for defining the exact TO. In this paper, correct TO estimation means that the estimated TO is within $-L_{CP}/2$ to $+L_{CP}/2$ from the arrival time of the maximum multipath components. Fig. 6 shows the FDR performances under channels: Model A, Model C and Model E. According to the results, the proposed technique shows performance gain of about 6 dB over the conventional technique in Model A, and about 3dB in Model C and Model E.

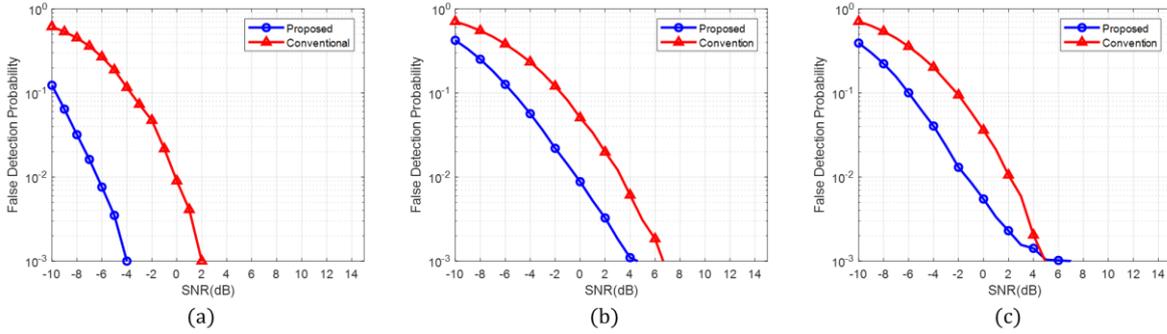


Fig. 6. TO estimation performances for (a) Model A, (b) Model C and (c) Model E

4.3 CFO estimation performance

Since the TO estimation CNN and CFO estimation CNN use the same input signal, the test set for TO estimation is the same as TO estimation. The performance evaluation measure of CFO estimation is RMSE. Fig. 7 shows the RMSE performance for Model A, Model C, and Model E. According to the results, the proposed technique shows much lower RMSE than that of the conventional technique at low to medium SNRs. However, at high SNRs, the conventional technique becomes similar or better [13,14].

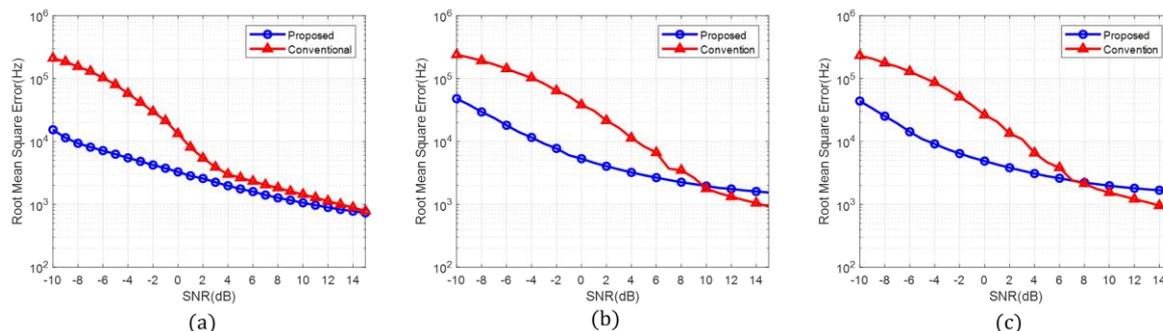


Fig. 7. CFO estimation performances for (a) Model A, (b) Model C and (c) Model E

5. Conclusions

This paper proposed a new deep learning based joint TO and CFO estimation technique for WLAN systems over multipath fading channels. The proposed technique estimates TO and CFO using only *STF* among legacy preambles for synchronization. According to computer simulation results, the proposed TO estimator show about 3 dB ~ 6 dB better than the conventional technique, and the proposed CFO estimator is better at low to medium SNRs. Therefore, if the proposed technique is applied to WLAN receivers, the receiver performance can be improved at low to medium SNRs.

Acknowledgements

This work was supported by the research fund of Hanbat National University in 2020

References

1. Sklar, B. (2001) *Digital communications: fundamentals and applications*. 2ndEd. Prentice Hall.
2. Terry, J. and Heiskala, J. (2002) *OFDM wireless LANs: A theoretical and practical guide*. Sams publishing.
3. Cho, Y. S., Kim, J. K., Yang, W. Y. and Kang, C. G. (2010) *MIMO-OFDM wireless communications with MATLAB*. John Wiley & Sons.
4. Son, Y. W., Kim, S. W., Byeon, S. H. and Choi, S. H. (2018) Symbol Timing Synchronization for Uplink Multi-User Transmission in IEEE 802.11ax WLAN. *IEEE Access* 6, 72962-72977.
5. Lee, E.-S. and Jeong, E.-R. (2019) Frame Synchronization by Using Convolutional Neural Network. *International Journal of Advanced Science and Technology* 28(4), 201-206.
6. Zhang, Z., Zhao, M., Zhou, H., Liu, Y. and Gao, J. (2004) Frequency offset estimation with fast acquisition in OFDM system. *IEEE Communications Letters* 8(3), 171-173.
7. Li, J., Liu, G. and Giannakis, G. B. (2001) Carrier frequency offset estimation for OFDM-based WLANs. *IEEE signal Processing Letters* 8(3), 80-82.
8. Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. and Vandergheynst, P. (2017) Geometric Deep Learning: Going beyond Euclidean data. *IEEE signal Processing Magazine* 34(4), 18-42.
9. Zhang, M., Diao, M. and Guo, L. (2017) Convolutional Neural Networks for Automatic Cognitive Radio Waveform Recognition. *IEEE Access* 5, 11074-11082.
10. Joung, J., Jung, S., Chung, S. and Jeong, E.-R. (2019) CNN-based Tx-Rx distance estimation for UWB system localization. *Electronics Letters* 55(17), 938-940.
11. Fort, A., Weijers, J.-W., Derudder, V., Eberle, W. and Bourdoux, A. (2003) A performance and complexity comparison of auto-correlation and cross-correlation for OFDM burst synchronization. In *2003 IEEE International Conference on Acoustics, Speech, and signal Processing, 2003. Proceedings. (ICASSP'03)* 2, 341-344.
12. Erceg, V., Schumacher, L., Kyritsi, P., Molisch, A. Baum, D. S., Gorokhov, A. Y., et al. TGN Channel Models. Version 4. *IEEE 802.11-03/940r4*.
13. Satapathy, S. K., Mishra, S., Mallick, P. K., Badiginchala, L., Gudur, R. R., & Guttha, S. C. (2019). Classification of Features for detecting Phishing Web Sites based on Machine Learning Techniques. *International Journal of Innovative Technology and Exploring Engineering*, volume8 (8S2), 425-430.
14. Mishra, S., Mallick, P. K., Tripathy, H. K., Bhoi, A. K., & González-Briones, A. (2020). Performance Evaluation of a Proposed Machine Learning Model for Chronic Disease Datasets Using an Integrated Attribute Evaluator and an Improved Decision Tree Classifier. *Applied Sciences*, 10(22), 8137.