Implementation and Evaluation of Channel Estimation and Phase Tracking for Vehicular Networks

Yongjiu Du, Dinesh Rajan, and Joseph Camp
Department of Electrical Engineering, Southern Methodist University (SMU), Dallas, TX
{ydu, rajand, camp}@smu.edu

***Abstract*—The high mobility of vehicular networks makes channel estimation and phase tracking challenging and an important problem for OFDM receiver design. In IEEE 802.11p vehicular networks, the channel estimation based on a long preamble in the PHY frame performs poorly in fast-fading channels. Moreover, the pilot-based phase tracking usually suffers from large residual phase errors. In this paper, we propose and implement a novel phase tracking algorithm, leveraging the decoded data. We also provide a detailed hardware design for a decoder-based channel estimation algorithm with a pipeline structure on an FPGA-based platform. Through diverse experiments via an advanced channel emulator, our results show that the proposed algorithm could significantly improve the system performance in terms of packet error rate, while adding less than 3% of additional hardware resources. We also jointly evaluate the packet error rate versus the bitwidth of the data in the FPGA, which is important to achieve a good balance between the hardware cost and the system performance.**

Keywords*—IEEE 802.11p; Vehicular Networks; Channel Estimation; Phase Tracking; OFDM; FPGA; Implementation;

I. Introduction

Vehicular networks could potentially provide safety, entertainment, roadway information, and navigation for drivers. However, due to the high velocity of vehicles, wireless channels change rapidly with time, providing much greater challenges for vehicular networks than other commercial communication systems (e.g., WLANs). IEEE 802.11p, a vehicular networks standard, has been released in recent years [1], which has the same PHY-layer frame structure as the IEEE 802.11a/g standard. However, it uses a bandwidth of 10 MHz and operates at the 5.9-GHz band. The narrower bandwidth and the higher carrier frequency make vehicular networks even more vulnerable to fast-fading channels. For example, if the lowest rate is used to transmit a packet of 2047 bytes, the frame duration is 5.5 ms. However, for a speed of 80 MPH, the channel coherence time is about 0.6 ms according to the coherence-time calculation in [2], which means that the channel is likely to change a number of times during the course of a packet.

Using short packets in fast-fading environments may help to reduce the influence of channel inconsistency within one frame slot, but small payloads usually lead to low spectrum efficiency due to the large overhead in both the PHY frame and the handshaking between the transmitter and receiver [3]. Channel estimation is one of the most important aspects of receiver designs, especially when targeting applications in time-varying multi-path wireless channels. In vehicular environments, the receiver needs to continually track the fast-fading channel in order to recover distorted signals. There exist several methods to estimate the channel characteristics for OFDM systems. Channel estimation usually leverages the long preamble sequence in IEEE 802.11a/g [4], [5], which is the common solution for off-the-shelf chipsets. However, in vehicular environments, the channel varies so frequently that the channel estimate at the beginning of the frame provides very little information for the channel state at the end of the same frame. Experimental results have shown that this channel estimation method can only be used for short packets (e.g., less than 400 bytes) with small delay spreads and high levels of Doppler shift or large delay spreads with low levels of Doppler shift [6]. Known sequences inserted between symbols could be used for channel estimation for time-varying channels [7], [8], but this technique is not compatible with the current IEEE 802.11p standard. Pilot sub-carriers in each OFDM symbol could provide some information for time-varying channels [9]. Nevertheless, in a multi-path channel with large delay spread, the number of pilots in IEEE 802.11p standard is too small to achieve an accurate channel estimate for each sub-carrier in the OFDM symbols. Decision-aided channel tracking techniques have been proposed to estimate the channel on each sub-carrier, which works well in high signal-to-noise ratio (SNR) scenarios [10]–[12]. Correspondingly, the decision become erroneous when the SNR is low, leading to poor performance of the channel estimation. Leveraging the decoded information to track the fast-fading channels has been shown to achieve more robustness due to the error correction gains from the decoder [13], [14], which is the state-of-the-art method via simulation. However, the decoder delay introduced by the buffer time in convolutional decoders usually makes the channel estimate stale. As a result, there is usually residual phase error for the OFDM symbols due to the residual carrier frequency offset, oscillator phase error, and channel variations. This residual phase error usually degrades the system performance significantly. Thus, leveraging the small number of pilots to recover the phase error usually performs poorly.

In this paper, we provide a novel phase tracking algorithm for the IEEE 802.11p vehicular networks standard. Moreover,

This research was supported by NSF grants CNS-0958436, CNS-1040429, and CNS-1150215.

978-1-4673-2480-9/13/$31.00 © 2013 IEEE

we design and implement a pipeline architecture for the decoder-based channel estimation method and the proposed phase tracking algorithm. We also present a detailed hardware design solution as well as the hardware resource usage on an FPGA-based platform. Through diverse experiments with an advanced wireless channel emulator, we show that the proposed algorithm outperforms existing methods significantly in terms of packet error rate. We evaluate the different packet error rates versus different bitwidths used in the signal processing in hardware, providing an important reference to achieve a good balance between the hardware cost and the system performance.

The main contributions of our work are as follows:

1) We propose a phase tracking algorithm based on the decoded data in IEEE 802.11p systems.
2) We provide detailed hardware design for the decoder-based channel estimation method and the proposed phase tracking algorithm, as well as a hardware cost comparison between different methods on an FPGA-based platform.
3) We evaluate the performance of different channel estimation and phase tracking mechanisms in a multi-path channel emulated over an advanced wireless channel emulator. For the proposed algorithm, we also present the diverse system performances versus different bitwidths to represent the signals in the FPGA.

This paper is organized as follows. Section II introduces the channel estimation and phase tracking compensation algorithms for the IEEE 802.11p standard. Next, we present the detailed hardware architecture and circuit design for the decoder-based channel estimation, the proposed phase tracking algorithm in Section III. In Section IV, we compare and evaluate the performance of the existing algorithms and the proposed method in terms of packet error rate. We also provide the different packet error rates versus different hardware costs. Finally, in Section V, we summarize our work and discuss the possible extensions of this work.

II. Channel Estimation and Common Phase Error Compensation

A. IEEE 802.11p PHY-Layer Frame Architecture

We use a PHY-layer frame structure as described in the IEEE 802.11p standard. One frame is composed of a short preamble, a long preamble, a header symbol, and several data symbols [15], as shown in Figure 1. The short preamble is used for frame detection, automatic gain control, and carrier frequency offset estimation. Traditional channel estimation usually takes advantage of the long preamble. The control information is coded and modulated in the header. The packet from higher layers is then coded, interleaved, and modulated in the data symbols of the frame.

According to the modulation process described in the IEEE 802.11p PHY-layer standard, the signal processing steps in the receiver could be summarized as follows:

1) Frame detection, automatic gain control, coarse carrier frequency estimation, and compensation based on the short preamble.

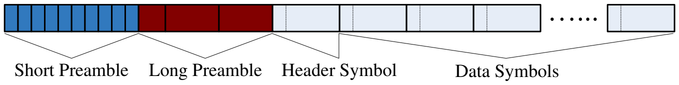

Fig. 1. IEEE 802.11p PHY frame structure, which is used in the analysis and evaluation in this work.

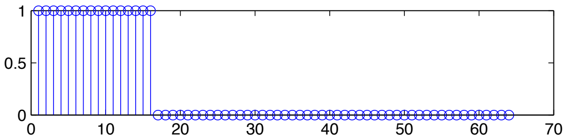

Fig. 2. Time domain impulse response of the complex-factor low-pass filter.

2) Symbol timing synchronization, channel estimation, and fine frequency estimation based on the long preamble.
3) Demodulation and equalization.
4) Signal de-mapping, de-interleaving, and decoding.

B. Channel Estimation

As mentioned before, channel estimation purely based on the long preamble is usually unable to track the fast-fading channels in a vehicular environment. In this section, we introduce a channel estimation scheme jointly using the long preamble and the decoded data symbols.

Initial Channel Estimation. The initial channel estimation leverages a long preamble. Since the channel estimation is usually designed in the frequency domain for OFDM systems, we use frequency domain expressions in the following analysis. Assume the two transmitted long-preamble symbols are represented by $X_{n,L1}$, $X_{n,L2}$, and the channel response by $H_{n,L1}$, $H_{n,L2}$, respectively. Here, n, $L1$ ($L2$) denote the sub-carrier index and the symbol index in the long preamble, respectively. Then, the two received distorted long-preamble symbols can be expressed as:

$$Y_{n,L1} = H_{n,L1}X_{n,L1} + W_{n,L1} \tag{1}$$

$$Y_{n,L2} = H_{n,L2}X_{n,L2} + W_{n,L2} \tag{2}$$

where $W_{n,L1}$ and $W_{n,L2}$ are the additive white Gaussian noise (AWGN).

Consequently, with $X_{n,L1}$ and $X_{n,L2}$ known at the receiver, the estimated channel response in the frequency domain can be estimated as:

$$\hat{H}_{n,L1} = \frac{Y_{n,L1}}{X_{n,L1}} \tag{3}$$

$$\hat{H}_{n,L2} = \frac{Y_{n,L2}}{X_{n,L2}} \tag{4}$$

Assuming a negligible channel difference between the two adjacent symbols in the long preamble, we can average the two estimated channel responses to obtain the least-squares (LS) estimate [9]:

$$\hat{H}_{n,I} = \frac{\hat{H}_{n,L1} + \hat{H}_{n,L2}}{2} \tag{5}$$

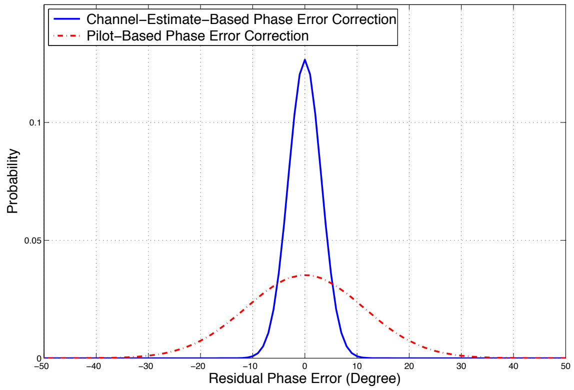

Fig. 3. Residual phase errors.

The LS estimate is usually susceptible to AWGN [16]. The minimum mean-square error (MMSE) estimate has been shown to be better than the LS estimate for channel estimation in OFDM systems [17]. However, both obtaining the channel statistics and the complex matrix operations are challenging for hardware implementations of MMSE estimation. Thus, we apply a frequency-domain complex-factor low-pass filter on the LS channel estimate over all sub-carriers to remove the noise beyond the maximum delay spread. Assuming that the maximum delay spread in a multi-path channel is within the guard interval (1.6 μs) between the OFDM symbols in IEEE 802.11p, the time domain impulse response of the low-pass filter we used is shown in Figure 2.

Decoder-Based Channel Estimation. Before the first data symbol being decoded, equalization is performed with the initial channel estimate. Next, the initially-equalized signal could be de-mapped, de-interleaved, and decoded. Then, we could re-encode, re-interleave, and re-map the decoded data to assist further channel estimation. Assuming the reconstructed signal for symbol i is $\tilde{X}_{n,i}$, we obtain the updated channel estimate according to:

$$\hat{H}_{n,i} = \frac{Y_{n,i}}{\tilde{X}_{n,i}} \tag{6}$$

We also apply the low-pass filter described in Figure 2 on the new channel estimate over all sub-carriers. Then, we apply an exponentially weighted moving average to further remove the noise effect. Finally, we obtain the updated channel estimate as:

$$\hat{H}_n = \alpha \hat{H}_n + (1-\alpha)\hat{H}_{n,i} \tag{7}$$

where α is a weighting factor, which could be configured in our hardware system.

C. Channel-Estimate-Based Phase Tracking

For convolutional decoding, both the BCJR decoder and Viterbi decoder usually keep a certain amount of data in a buffer and jointly decode all the data in the buffer. Due to the buffering delay, the channel estimation method we described above could not provide an up-to-date phase estimate for the symbols to be equalized. Pilot-based phase tracking methods for IEEE 802.11 systems have been discussed in [18], [19]. However, the traditional pilot-aided phase tracking has poor performance due to the small number of pilot sub-carriers in IEEE 802.11p systems. The phase error may degrade the system performance severely, especially for higher-order constellation modes. In this section, we propose a phase tracking algorithm, which could improve the system performance significantly in addition to the channel estimation method. The phase difference indicator between the current channel estimate and the previous channel estimate on sub-carrier n for symbol i is:

TABLE I. TRANSMISSION RATE PARAMETERS

Rate Index	Constellation	Code Rate	$R_n(bit)$
0	BPSK	1/2	24
1	BPSK	3/4	36
2	QPSK	1/2	48
3	QPSK	3/4	72
4	16QAM	1/2	96
5	16QAM	3/4	144
6	64QAM	2/3	192
7	64QAM	3/4	216

TABLE II. FEEDBACK DELAYS FOR DIFFERENT RATES

Rate Index	0	1	2	3	4	5	6	7
Delay (symbol)	6	5	5	4	4	4	4	4

$$\hat{P}_{n,i} = \frac{\hat{H}_{n,i}}{\hat{H}_{n,i-1}} \tag{8}$$

Then, we average the phase error indicators over all N sub-carriers:

$$\bar{P}_i = \frac{1}{N}\sum_{n=1}^{N} \hat{P}_{n,i} \tag{9}$$

Finally, we normalize $\bar{P}_i$ to get the estimated phase error indicator to further recover the distorted signals.

$$\hat{P}_i = \frac{\bar{P}_i}{|\bar{P}_i|} \tag{10}$$

The residual phase error distributions for traditional pilot-based phase-error removal and the proposed channel-estimate-based phase-error correction are shown in Figure 3. We can see that the proposed method could achieve a much more accurate phase error correction.

D. Signal Equalization

After achieving the updated channel estimate and the normalized phase error indicators between two adjacent symbols, we can equalize the data symbols as:

$$\hat{X}_{n,k} = \frac{\hat{Y}_{n,k}}{\hat{H}_n \hat{P}^{k-i}} \tag{11}$$

Here, $(k-i)$ is the channel tracking symbol delay, which can be found in Table II for each rate.

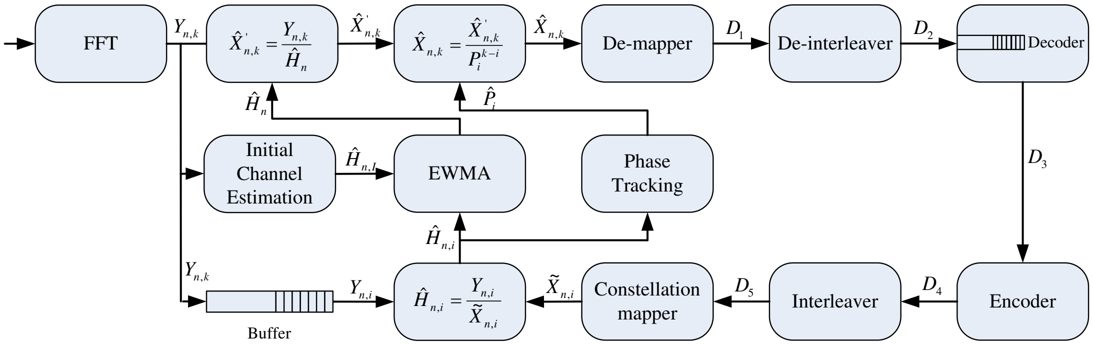

Fig. 4. Pipe-line structure of the implemented channel estimation and phase tracking algorithm.

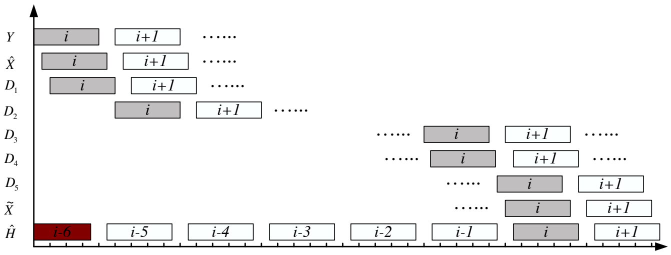

Fig. 5. Gantt chart of the signals from different components in the receiver.

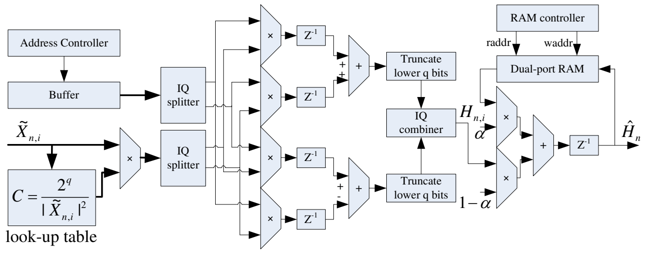

Fig. 6. Channel estimation circuit architecture.

III. Hardware Implementation

In this section, we introduce a detailed pipeline structure of the channel estimation and the proposed phase tracking algorithms, as shown in Figure 4. In our design, to decode the noisy signals, we use a Viterbi decoder based on a trace-back structure with a trace-back depth of 64. Let R_n denote the number of data bits that can be transmitted in one OFDM symbol at rate n, as shown in Table I [15]. The number of symbols delayed by the decoder is $\lceil 64/R_n \rceil$. The total delay between the reconstructed OFDM symbol from the decoded data and the current symbol to be equalized is shown in Table II, which includes the delays introduced by de-mapping, de-interleaving, decoding, re-encoding, re-interleaving, and re-mapping. The Gantt chart describing the system timing for rate 0 is shown in Figure 5, where we can see the signal delays in each step. The definition of some of the signals in Figure 5 can be found in Figure 4.

Some of the components in Figure 4, such as FFT, de-mapper, de-interleaver, Viterbi decoder, convolutional encoder, interleaver, constellation mapper, and initial channel estimator, have been well discussed for OFDM receivers [20]. Thus, we only introduce the hardware architecture of the channel estimation and phase tracking design in this work.

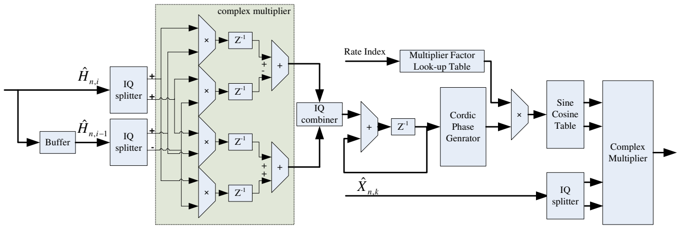

Fig. 7. Channel-estimate-based phase tracking circuit architecture.

Channel Estimation Circuit Design. Due to the processing delay of the reconstructed OFDM symbols $\tilde{X}_{n,i}$, we use a buffer to delay the received signal $Y_{n,i}$ and make it synchronized with $\tilde{X}_{n,i}$. When the constellation mapper module outputs $\tilde{X}_{n,i}$, the buffer controller reads the buffered $Y_{n,i}$ and performs a complex division to obtain $\hat{H}_{n,i}$. We first put $\hat{H}_{n,I}$ in a dual-port RAM. When $\hat{H}_{n,1}$ is calculated, the dual-port RAM controller reads the buffered $\hat{H}_{n,I}$ from one port of the dual-port RAM and performs the computation as described in (7). Then, the RAM controller writes $\hat{H}_n$ back to the dual-port RAM via the other port. The architecture is shown in Figure 6.

Phase Tracking Circuit Design. We leverage the channel estimate from every two adjacent symbols to estimate the common phase error, as shown in Figure 7. We first use a buffer to delay the channel estimates for one symbol duration. Then we use a complex multiplier to get the phase difference indicator between the two channel estimates from adjacent symbols of each sub-carrier. An accumulator is used to average the phase difference indicators over all sub-carriers. We then use a Cordic phase generator to calculate the phase error from the average phase difference indicator. We then predict the phase rotation for the symbol to be decoded, considering a different symbol delay for each rate. Finally, we use a sine-cosine look-up table to calculate the rotation vector to remove the common phase error.

IV. Experimental Evaluation

The FPGA-based platform we use for our implementation and experimental evaluation is the Wireless Open-Access Research Platform (WARP). WARP is a useful platform supporting a fully customized cross-layer design [21]. Mainly, the

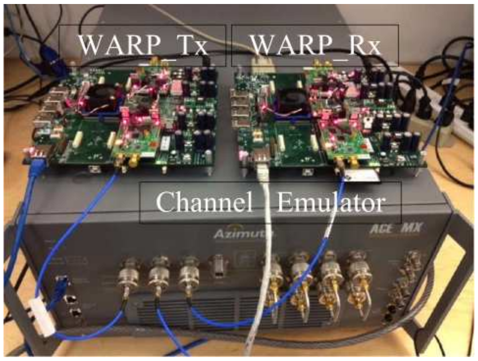

Fig. 8. Performance emulation system using WARP and Azimuth channel emulator.

TABLE III. TOTAL HARDWARE RESOURCES IN XC4VFX100-11

Resource Name	Flip-Flop	LUT	Slice	RAMB16
Amount	84352	84352	42176	376

PHY layer is implemented in the FPGA fabric, and the higher layers exist as C code on an embedded PowerPC [20]. In contrast to the commonly used reference design for WARP which heavily leverages Xilinx System Generator for the physical layer implementation, we use Verilog HDL to design and implement a full OFDM transceiver according to the IEEE 802.11p standard. We implement complete real-time signal processing, synchronization and control systems in the fabric of the FPGA. The PowerPC hosts the MAC controller implementation in this system. The FPGA on WARP is a Xilinx Virtex IV XC4VFX100-11, which is a large-capacity device. The total amount of the main hardware resources in that FPGA is shown in Table III. To directly evaluate different algorithms over repeatable and controlled channels, we use the Azimuth ACE-MX channel emulator to generate diverse fast-fading channel effects, which provides approximately the same effects as complex over-the-air channels. The hardware and performance emulation system is shown in Figure 8.

In our experiment, we set the carrier frequency to 5620 MHz, which is very close to the IEEE 802.11p band (WARP can not operate at the band of 5.9 GHz). Thus, our experiment results could approximate those from true IEEE 802.11p systems. We use a 10-MHz bandwidth which is the same as that in the IEEE 802.11p standard. A packet size of 1600 bytes is used throughout our experiments. During our channel-emulator-based test, we use a 2-tap Rayleigh fast-fading channel, with 0-db relative power attenuation and 0.5 μs delay spread, which is similar to the highway channel model described in [22]. We set the emulated vehicle speed to 40 MPH and 80 MPH, which is typical for local and highway traffic, respectively.

We use Xilinx XPS to compile our PHY-layer logic design and MAC-layer software control and run it in the FPGA on WARP. The hardware cost for different channel estimation schemes are shown in Table IV. The first scheme is a channel estimation scheme purely based on the long preamble with pilot-based phase tracking (PP), which is currently used in off-the-shelf chipsets [6]. The second scheme is a decoder-based channel estimation with pilot-based phase tracking (DP), which is the state-of-the-art algorithm published based on simulation [13]. The third scheme is a decoder-based channel estimation with the proposed channel-estimate-based phase tracking in this work (DC). We can see that, the additional hardware used for the proposed algorithm is very little. However, the packet error rate (PER) is dramatically improved, as shown in Figure 9 and 10. We have shown the emulation results for QPSK with rate-1/2 convolutional code, 16QAM with rate-1/2 convolutional code, and 64QAM with rate-2/3 convolutional code, although similar improvements are also obtained for other rates listed in Table I. Moreover, the maximum operating frequency for the FPGA nearly remains the same for different schemes.

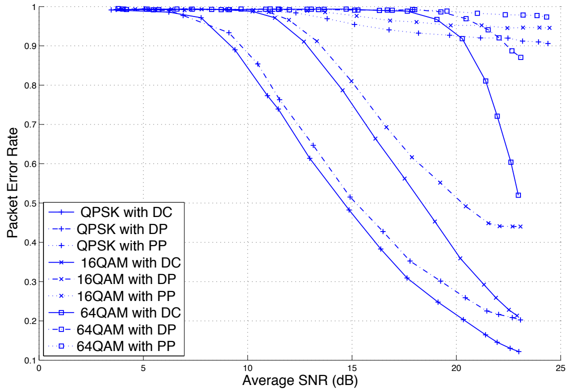

Fig. 9. Packet error rate with different average SNR in a Rayleigh fading channel at a speed of 40 MPH.

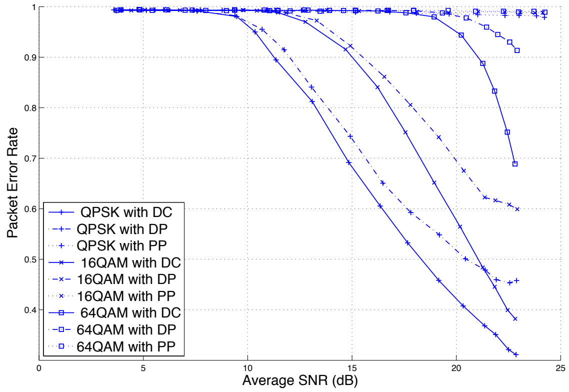

Fig. 10. Packet error rate with different average SNR in a Rayleigh fading channel at a speed of 80 MPH.

TABLE IV. FPGA-BASED HARDWARE COST FOR DIFFERENT SCHEMES

Scheme	Flip-Flop	LUT	Slice	RAMB16
PP	19543(23%)	29324(34%)	18910(44%)	25(6%)
DP	20062(24%)	34618(41%)	21794(52%)	27(7%)
DC	20658(24%)	37170(44%)	23362(55%)	27(7%)

TABLE V. MAXIMUM OPERATING FREQUENCY

Scheme	S0	S1	S2
Frequency (MHz)	101.34	101.23	101.19

In order to investigate the performance with diverse bitwidths during the digital signal processing, we emulate the

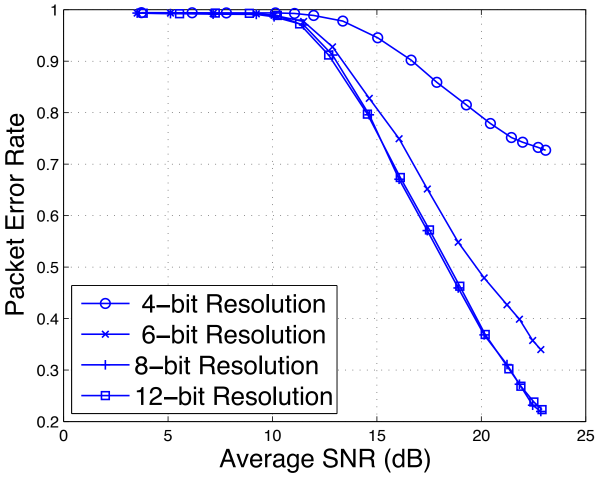

Fig. 11. Packet error rate with different bitwidths and different average SNR in a Rayleigh fading channel.

TABLE VI. FPGA-BASED HARDWARE COST FOR DIFFERENT BITWIDTHS

bit-width	Flip-Flop	LUT	Slice	RAMB16
4	19914(24%)	31939(38%)	20394(48%)	26(7%)
6	20115(24%)	33412(40%)	21197(50%)	26(7%)
8	20285 (24%)	34554(41%)	21878(52%)	26(7%)
12	20658(24%)	37170(44%)	23362(55%)	27(7%)

PER performance with different bitwidths for the implementation of the channel estimation and phase tracking function (DC), as shown in Figure 11. In this experiment, we set the vehicle speed to 40 MPH. We also provide the hardware cost for each bitwidth listed in Table VI. We can see that the performance with 8 bits and 12 bits have similar packet error rates. However, using 4 bits and 6 bits in the hardware results in a worse performance. We conclude that an appropriate bitwidth is 8 in our design, which saves hardware resources while maintaining the PER performance.

V. CONCLUSION

In this paper, we proposed and implemented a novel phase tracking algorithm. We also implemented a decoder-based channel estimation algorithm with a pipeline structure. We provided detailed hardware design and resource usage on an FPGA-based platform. Moreover, through our emulation over a wireless channel emulator, we showed that our proposed algorithm significantly improves the system performance in terms of packet error rate, while adding very little hardware cost compared to the total implementation resources used by the receiver. We jointly showed the system performance versus the bitwidth used, which is important for device developers to achieve a good balance between the cost and the performance. In future works, we will apply our channel tracking and common phase error compensation algorithm to other advanced systems (e.g., IEEE 802.11n, 802.11ac).

REFERENCES

[1] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, 2010.

[2] T. Rappaport, *Wireless Communications Principles*. Upper Saddle River, NJ:Prentice-Hall, 1999.

[3] P. Mary, M. Dohler, J. Gorce, and G. Villemaud, "Packet error outage for coded systems experiencing fast fading and shadowing," *Wireless Communications, IEEE Transactions on*, vol. PP, no. 99, 2013.

[4] Z. Zheng, "Novel channel estimation for the ofdm-based wireless local area network systems," in *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on*, april 2010.

[5] M. Belotserkovsky, "An equalizer initialization algorithm for ieee802.11a and hiperlan/2 receivers," *Consumer Electronics, IEEE Transactions on*, vol. 48, no. 4, pp. 1051 – 1055, nov 2002.

[6] C. Mecklenbrauker, A. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, "Vehicular channel characterization and its implications for wireless system design and performance," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189 –1212, july 2011.

[7] C.-S. Lin, C.-K. Sun, J.-C. Lin, and B.-C. Chen, "Performance evaluations of channel estimations in ieee 802.11p environments," in *Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on*, oct. 2009.

[8] C.-S. Lin and J.-C. Lin, "Improved time-domain channel estimation techniques in ieee 802.11p environments," in *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, july 2010.

[9] J. Fernandez, D. Stancil, and F. Bai, "Dynamic channel equalization for ieee 802.11p waveforms in the vehicle-to-vehicle channel," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 29 2010-oct. 1 2010.

[10] A. Bourdoux, H. Cappelle, and A. Dejonghe, "Channel tracking for fast time-varying channels in ieee802.11p systems," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, dec. 2011.

[11] Z.-S. Lin, T.-L. Hong, and D.-C. Chang, "Design of an ofdm system with long frame by the decision-aided channel tracking technique," in *Electro/information Technology, 2006 IEEE International Conference on*, may 2006.

[12] T. Kella, "Decision-directed channel estimation for supporting higher terminal velocities in ofdm based wlans," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, dec. 2003.

[13] T. Zemen, L. Bernado, N. Czink, and A. Molisch, "Iterative time-variant channel estimation for 802.11p using generalized discrete prolate spheroidal sequences," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 3, pp. 1222 –1233, march 2012.

[14] H.-W. Kim, C.-H. Lim, and D.-S. Han, "Viterbi decoder aided equalization and sampling clock tracking for ofdm wlan," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, sept. 2004.

[15] *IEEE Std 802.11-2007*, 2007.

[16] P. Pradhan, O. Faust, and B. Patra, S. andChua, "Channel estimation algorithms for ofdm systems," in *International Conference on Electronics Systems*, Jan 2011.

[17] J.-J. van de Beek, O. Edfors, M. Sandell, S. Wilson, and P. Borjesson, "On channel estimation in ofdm systems," in *Vehicular Technology Conference, 1995 IEEE 45th*, jul 1995.

[18] K. Akita, R. Sakata, and K. Sato, "A phase compensation scheme using feedback control for ieee 802.11a receiver," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, sept. 2004.

[19] R. Sakata, K. Akita, and K. Sato, "Real-time phase tracking method for ieee 802.11a/g/n receiver under phase noise condition," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, may 2006.

[20] P. O. Murphy, "Design, implementation and characterization of a cooperative communications system," Ph.D. dissertation, Rice University, TX, USA, 2010.

[21] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," in *ACM MobiCom*, San Francisco, CA, Sep. 2008.

[22] G. Acosta-Marum and M. Ingram, "Six time- and frequency-selective empirical channel models for vehicular wireless lans," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, 30 2007-oct. 3 2007, pp. 2134 –2138.