

Research Article

Real-Time Emulation of Nonstationary Channels in Safety-Relevant Vehicular Scenarios

Golsa Ghiaasi ¹, Thomas Blazek ², Mehdi Ashury,²
Rute Ramalho Santos,³ and Christoph Mecklenbräuer ²

¹Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway

²Institute of Telecommunications, TU Wien, Vienna, Austria

³Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

Correspondence should be addressed to Thomas Blazek; thomas.blazek@tuwien.ac.at

Received 29 September 2017; Accepted 18 February 2018; Published 8 May 2018

Academic Editor: Marceau Coupechoux

Copyright © 2018 Golsa Ghiaasi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes and discusses the architecture for a real-time vehicular channel emulator capable of reproducing the input/output behavior of nonstationary time-variant radio propagation channels in safety-relevant vehicular scenarios. The vehicular channel emulator architecture aims at a hardware implementation which requires minimal hardware complexity for emulating channels with the varying delay-Doppler characteristics of safety-relevant vehicular scenarios. The varying delay-Doppler characteristics require real-time updates to the multipath propagation model for each local stationarity region. The vehicular channel emulator is used for benchmarking the packet error performance of commercial off-the-shelf (COTS) vehicular IEEE 802.11p modems and a fully software-defined radio-based IEEE 802.11p modem stack. The packet error ratio (PER) estimated from *temporal averaging* over a *single* virtual drive and the packet error probability (PEP) estimated from *ensemble averaging* over *repeated* virtual drives are evaluated and compared for the same vehicular scenario. The proposed architecture is realized as a virtual instrument on National Instruments™ LabVIEW. The National Instrument universal software radio peripheral with reconfigurable input-output (USRP-Rio) 2953R is used as the software-defined radio platform for implementation; however, the results and considerations reported are of general purpose and can be applied to other platforms. Finally, we discuss the PER performance of the modem for two categories of vehicular channel models: a vehicular nonstationary channel model derived for urban single lane street crossing scenario of the DRIVEWAY'09 measurement campaign and the stationary ETSI models.

1. Introduction

The key characteristics of vehicle-to-anything (V2X) propagation channels are their nonstationarity, shadowing by other vehicles, and high Doppler shifts [1].

On a small scale in time and space, the channel characteristics in safety-relevant vehicular scenarios exhibit significant delay-Doppler spreads and the channel's spreading is varying over time and frequency. These are due to several factors: major Doppler shifts are caused by moving transmitters, receivers, and mobile interacting objects in the multipath propagation channel. Many other details shape the local delay spread and Doppler spectral characteristics of V2X channels, for example, antenna patterns and the angular statistics of individual multipath components (MPCs) [2].

On a larger spatial and temporal scale, the *statistics* of the channel characteristics due to the surrounding environment, link and shadowing geometry, and velocities of physical objects evolve. On this coarser scale, the V2X multipath propagation channel exhibits nonstationarity [3, 4] and such channels do not belong to the well-known class of wide-sense stationary uncorrelated scattering (WSSUS) models. Both the small- and the large-scale effects have significant influence on the reliability and latency of V2X packet transmission.

Aiming at standardized conformance tests of V2X radio modems, both IEEE and ETSI have proposed channel models [5, 6] which reproduce the short-term delay-Doppler characteristics in various vehicular scenarios [2, 7]. These channel models belong to the class of WSSUS models because neither

the channel's tap delays nor their Doppler spectrum evolves in time. As a consequence, the local scattering function [3, 4] of these channel models becomes independent of time and frequency. The channel models in [6] will be referred to as ETSI models in this work for simplicity.

Recent research on real-world channel sounder data acquired in vehicular scenarios [3] proved that the WSSUS channel assumptions are violated for V2X transmission durations longer than 40 ms and bandwidths larger than 40 MHz. Thus, the ETSI models are not necessarily relevant to vehicular scenarios of longer duration and/or larger bandwidth.

In order to reproduce the varying Doppler and delay statistics of V2X channels on a larger scale, it is essential to allow non-WSSUS channel models featuring nonstationary fading on individual MPCs. For emulating such V2X channels, finite regions in the time-frequency plane are defined in which the process is assumed to be locally stationary [4]. Within such local stationarity region, the V2X channel is modeled adequately by means of the well-known tapped-delay line architecture [8]. In this contribution, we report on a V2X channel emulator for transmissions occupying up to 20 MHz of bandwidth and scenario durations much longer than 40 ms. For the computation of the channel output signal in real time, this means that the gain coefficients of a tapped-delay line representation need to be updated whenever the signal leaves the local stationarity region in time.

To reproduce the V2X channel characteristics in a lab setting, a channel emulator is placed between the transmitter (TX) and the receiver (RX) and connected by RF coaxial lines. This facilitates reproducible tests and measurements and has proved to be a cost-effective means for benchmarking of the V2X radio modem performance before heading for costly real-world drive tests. For safety-relevant applications, the emulated channel models for modem benchmarking shall represent somewhat harsher than typical V2X channel conditions. This approach is more conservative than aiming at accurately representing the V2X channel.

There are several commercial solutions available for real-time channel emulation [9, 10]. They offer adequate solutions for the emulation of WSSUS channels and their specifications are targeted for cellular network applications. In [11], researchers have reported a field-programmable gate array- (FPGA-) based emulator for stationary vehicular environments based on the Ingram-Acosta models [12] which belong to the WSSUS class. To the best of our knowledge, all these lack the capability to emulate non-WSSUS channels as observed in safety-related vehicular scenarios [3]. In [13], a channel emulator for vehicular modem stress tests is reported. For the stress test, an idealized channel with just two taps of equal gain is emulated. The first tap models the LOS component with zero delay and zero Doppler and the second tap models a specular reflection with delay τ and Doppler ν . Reference [13] served as a starting point for our design.

In this work, we present the design and implementation of an emulator on a software-defined radio (SDR) platform. FPGAs are suitable options which offer flexibility in design process along with shorter time needed for prototyping; however, their finite resources, such as gate count and data and clock rates, impose limitation on the design of

the emulator. Therefore the main challenge is to optimize the trade-off between reconfigurability of the system while keeping the computational complexity as low as possible. The chosen platform is National Instruments universal software radio peripheral with reconfigurable input-output (USRP-Rio) 2953R which operates at frequencies up to 6 GHz and processes signals with up to 40 MHz bandwidth. It features two multiple-input multiple-output (MIMO) RF chains along with a Xilinx Kintex-7 FPGA which is programmable with the National Instruments LabVIEW FPGA Module [14]. The emulator documented in this work can either playback *recorded* channel impulse responses or emulate stochastic channel models [2, 6]. In doing so, the real-time emulator supports taps with fixed and variable delays and Doppler shifts and facilitates the Monte Carlo realizations of path-loss and phase variables.

We use the emulator to evaluate the benchmarking experiments in terms of the packet error ratio (PER) and the packet error probability (PEP) for several vehicular commercial off-the-shelf (COTS) modems and two sets of fading channels: (1) ETSI channel models and (2) a low complexity nonstationary channel model derived from the V2V urban crossing scenario channel impulses measured in the DRIVEWAY'09 campaign [15].

This manuscript is organized as follows: firstly, we present the overview of a continuous time delay- and time-variant channel as an input-output system. In Section 3, we discuss the structure of the emulator with a focus on generating sampled traces for accurate representation of the fading channels while abiding to the physical restrictions of the hardware. The design of the real-time emulator is presented in Section 4. The testbed configuration for the benchmarking tests is explained in Section 5. The description of the tests along with the experimental results is presented in Section 6. The measurements are conducted by analyzing the emulation performance for both presented model types, and then benchmark measurements of two types of modems are conducted.

2. Channel Emulation

The small-scale fading characteristics of the V2X channel within a stationarity region are modeled by complex-valued stochastic processes representing the gains and phase shifts on individual MPCs. The continuous time delay- and time-variant impulse response $h(t, \tau)$ of the V2X channel is represented as a superposition of MPCs:

$$h(t, \tau) = \sum_{p=1}^P h_p(t) \delta(\tau - \tau_p(t)), \quad (1)$$

where P is the number of the relevant MPCs defined by significant received power level, and $h_p(t)$ is

$$h_p(t) = \eta_p \hat{\eta}_p(t) e^{j(2\pi\nu_p t + \hat{\phi}_p(t))}, \quad (2)$$

in which η_p is a positive constant attenuation and ν_p is a real number representing the Doppler shift of the p th

MPC, respectively. Furthermore, the small-scale fading of the p th MPC is modeled by the pair of stochastic processes $\hat{\eta}_p(t)$ and $\hat{\phi}_p(t)$. The delays $\tau_p(t)$ of the MPCs themselves are also time-dependent. To carry out the calculation of the time-discrete output signal sample stream $r[m]$ representing the continuous-time function $r(mT_s)$ sampled at a sampling period of T_s , the V2X channel emulator implements a discrete-time convolution of the time-varying channel impulse response (1) and the input signal sample stream $s[m] = s(mT_s)$, that is,

$$r(t) = \int_{-\infty}^{\infty} h(t, t - \tau) s(\tau) d\tau = \sum_{p=1}^P h_p(t) s(t - \tau_p). \quad (3)$$

The output $r(t)$ needs only be calculated for $t = mT_s$ with $m = 1, 2, \dots$:

$$r(mT_s) = \sum_{p=1}^P h_p(mT_s) s(mT_s - \tau_p). \quad (4)$$

3. Emulator Structure

In our previous work in [17], we presented a general systematic approach to emulation of a geometry-based stochastic channel model. This structure or its subsets can serve as a platform for other classes of channel models, such as stochastic models and models formed by recorded impulse response. Equation (4) indicates that in order to perform a real-time emulation, it is required to generate a sampled trace which realizes the fading properties of the channel and then convolve the trace with samples of the transmitted signal at the emulator. Since the data exchange and sampling rates in the emulator are limited, we would need to choose the parameters of the trace in an effective manner, so as to maximize the accuracy of the model while optimizing the implementation complexity and usage of hardware resources. To accurately compute the vehicular channel models, it is required to evaluate a large number of MPC parameters (namely, attenuation, path delays, and angles of arrival and departure) and to sum them up in real-time. Therefore, we apply a complexity reduction method aiming at determining the minimum value for P which still achieves the accuracy desired for the channel models, hence reducing the number of gates needed on the FPGA for real-time computation. For instance, in [16], LASSO regression has been applied to sets of recorded impulse responses in DRIVEWAY'09 vehicular measurement campaign [3, 7], resulting in a nonstationary parsimonious model which has been benchmarking using the emulator in Section 6. Similarly, the model reduction deployed in [18, 19] utilizes clustering technique in which the paths with similar properties in terms of delay and Doppler spread are grouped and represented as a cluster. For instance one cluster could represent different points of a single moving object which obviously exhibit similar delay and Doppler spreads. This model was used in [17] for benchmarking the set of commercial modems in V2V scenario on highway with obstructed line of sight.

After setting the parameters such as center frequency and the set of relevant MPCs and their associated parameters,

we would need to determine the required update rates for each set of parameters in order to maximize the channel accuracy and to account for rapid variation in the vehicular channel. Depending on the vehicular scenario, it may be required to account for time-varying antenna patterns due to changes in their orientation and antenna adaptivity. These parameters have different variation rates: the variation in shadowing relates to large-scale fading and is characterized by the stationarity time of the channel. The required update rate is low compared to the channel's complex-valued tap weights which are modeled as stochastic processes with the Doppler bandwidth of the channel.

For the purpose of in-lab tests, the traces are calculated offline and the resulting samples forming the $h_p(t)$ s are stored in a fast memory, which streams the corresponding parameters to the FPGA (referred to as time-variant convolution module), which carries out the real-time calculation of output (RX) samples by convolving the discrete-time varying impulse response of the channel with the input samples (from TX) according to (4). We will look into architecture of the time-varying convolution module in Section 5.

4. Real-Time Emulator Design

Since time-variant convolution is the module which performs the real-time calculation in the FPGA, its architecture is the key to achieving the trade-off between complexity and accuracy. The first option is to implement the most general form of convolution by calculation of the discrete version of (4) (shown in (5)). The impulse response coefficients are complex-valued and given by (2). This architecture is capable of emulating any linear time-varying channel response by simple operations: complex number multiplication and additions:

$$r[m] = \sum_{p=1}^P C_p[m] s[m - d_p]. \quad (5)$$

Here, $\tau_p = d_p T_s$; therefore d_p is the number of samples that a tap is offset. The drawback is that the rate of update of C_p s is determined by the necessary rate for accurate modeling of $e^{j2\pi\nu_p t}$ elements. This rate may become too high to stream to FPGAs in typical SDRs.

For this reason, we implement a tapped-delay line architecture where each tap is parameterized by its delay, weighted in magnitude, and phase-rotated. The Doppler shift associated with a tap is explicitly parameterized which significantly reduces the required update rate for the channel parameters. Due to the time-discrete nature of the FPGA implementation, the architecture realizes only discrete delays and assumes that delay lags are on the sampling grid; hence, the MPC delay lags which are off-grid are approximated by shifting them to one of the neighboring on-grid delay (in a future extension of this work, fractional delays are foreseen as well).

The architecture shown in Figure 1 implements this tapped-delay line architecture restricted to on-grid delays

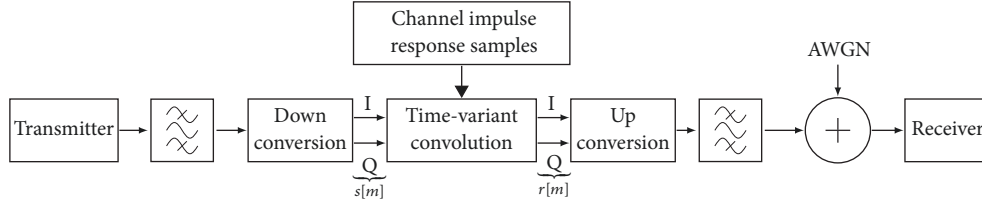


FIGURE 1: System view of the emulator.

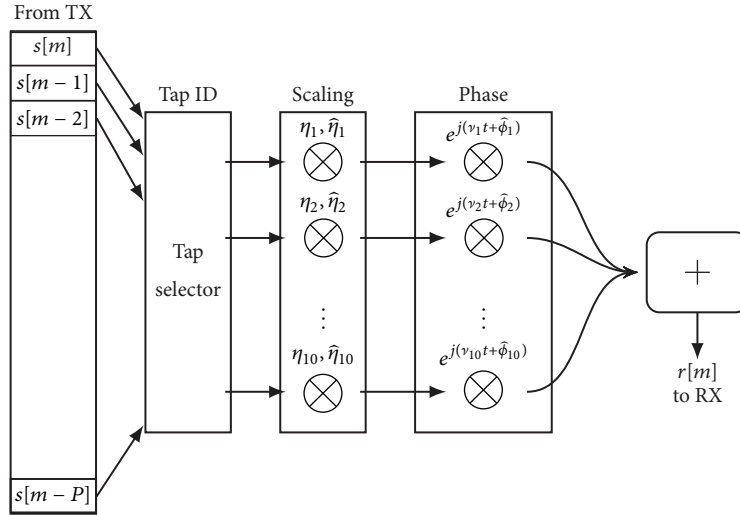


FIGURE 2: Tapped-delay architecture for time-variant convolution.

according to the time-discrete approximation of (4), that is,

$$r[m] = \sum_{p=1}^P \eta_p \hat{\eta}_p e^{j2\pi(\nu_p m T_s + \hat{\phi}_p)} s[m - d_p]. \quad (6)$$

The FPGA implementation follows the architecture in Figure 2 implementing (6). In each vehicular scenario snapshot, for each active tap, the index determines the relevant equalized delay which then is passed on to the FPGA along with four other parameters: η_p and $\hat{\eta}_p$ which represent the deterministic and stochastic path-loss components, along with ν_p and $\hat{\phi}_p(t)$ which represent the Doppler shift and the phase shift components for the selected tap. η_p and ν_p are slowly varying factors while $\hat{\eta}_p$ and $\hat{\phi}_p$ model the fast-varying behavior of the coefficients which are passed on through high speed DMA FIFO structures.

As for implementation of the real-time delays, this architecture relies on a shift-and-discard array structure in order to store the delayed input samples at equally sampled delay taps, rather than implementing the time-delay operation on the FPGA. At each instance of the FPGA clock, the new sample from emulator input arrives and the stored samples are shifted in the array. To perform the emulation, according to the delays of the taps in the channel model, the samples of the input data from the array are selected and then scaled by path-loss factor and rotated by corresponding

Doppler/phase shifts. To keep the gate count of the FPGA in limits, the frequency and phase shift operation as well as the IQ additions and multiplications are implemented by a low complexity algorithm. The input $s[m]$ and output $r[m]$ samples are complex numbers relating to inphase (I) and quadrature (Q) components. In our implementation, the system is designed to have a bandwidth of 20 MHz which relates to 40 MS/s and 25 ns of sampling time. In order to cover the contributions of the reflectors which are 0.5 km away from the transceiver, we have dedicated a total of 68 taps with delay resolution of 25 ns from which a maximum of 10 taps may be active simultaneously. The impulse response samples are passed on to the emulator by the host PC as depicted in Figure 1.

5. Emulator Testbed

The testbed consists of a USRP as the emulator connected to a personal computer (PC) running LabVIEW 2014 and two IEEE 802.11p modems connected according to Figure 3. The PC fitted with the PCIe interface card is set up as LabVIEW Host which at the start of the operation loads the compiled bitfiles into the FPGA and configures several parameters related to the RF chains.

We have two sets of modems for our experiments presented in Section 6. For one set of experiments, we use a set of COTS IEEE 802.11p modems. In the second experiment,

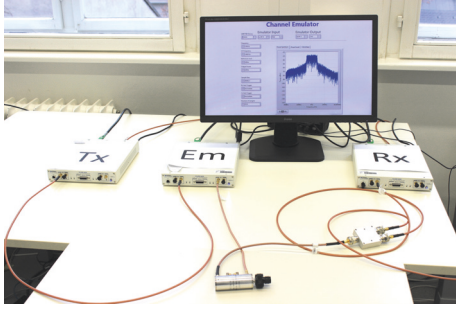


FIGURE 3: Testbed setup: transmitter (Tx), vehicular channel emulator (Em), stepped attenuator, and receiver (Rx) wired by RF coaxial cables.

the modems are implemented on a set of USRPs running the NI IEEE 802.11 Framework code which has been modified to operate as IEEE 802.11p stack.

The modems are set up as transmitter (TX) and receiver (RX). The TX is connected to the USRP port RF0 which is configured as a receiver, while the RX is connected to the port RF1 which is configured as a transmitter. There is a 30 dB attenuator placed between the TX modem and RF0 port of the USRP to avoid saturation of subsequent stages. Similarly, there is a 60 dB attenuator placed between the RF1 port of the USRP and the RX modem to avoid an overdrive. The operation is as follows: the IQ data generated by the TX modem is downconverted by the RF0 chain, processed through the FPGA which acts as the channel. Then, the signal is again upconverted by the RF1 chain and is received by the RX modem.

6. Channel Emulation Measurements

In this section, we describe the experiments we have performed using the emulator in two sets of measurement. To this end, we first demonstrate the ability to realize, emulate, and measure an arbitrary, time-variant channel, by means of sparse fits. We use the results to demonstrate our performance analysis and validate the behavior of the emulator. In a second step, we introduce stationary tapped-delay line models designed for testing and validation of commercial modems. We compare our results to the general case and continue to show the application of performance evaluation. Before we continue to the measurements, we introduce our performance metrics.

6.1. Performance Metrics. From a system level point of view, we describe the performance of the channel via a time-variant probability of the packet transmission being unsuccessful, denoted by the packet error probability $P(\mathcal{E})$ [2], where \mathcal{E} is an unsuccessful transmission event. In our case, the time-variant behavior is caused by the time-variance of the fading, which is characterized by five sets of parameters. These sets can be seen in (6) and constitute η , $\hat{\eta}$, $\underline{\nu}$, $\hat{\phi}$, and \underline{d} . Here the underline denotes the set of all P parameters needed at a given time step t . However, for ease of writing, we can combine the multiplicative terms into small-scale fading

parameters $h_p = \eta_p \hat{\eta}_p e^{j2\pi(\nu_p m T_s + \hat{\phi}_p)}$ and thus the time-variant error probability $P(\mathcal{E})$ is fully characterized by 2 sets, the P relevant channel coefficients \underline{h} and the corresponding delays \underline{d} . To measure an estimate of $P(\mathcal{E})$, we connect a transmitter and a receiver to the emulator and log all successful and unsuccessful transmissions. If we are able to sample the emulator at all time instance where the fading parameters \underline{h} and \underline{d} assume the same values \underline{h}' , \underline{d}' , we arrive at an estimated $P(\mathcal{E})$ for those parameter combinations, by calculating an *ensemble mean* over all realizations where the parameters were equal to \underline{h}' , \underline{d}' :

$$P(\mathcal{E} | \underline{h}', \underline{d}') = \frac{\text{Occurrences of } \mathcal{E} \text{ given } \underline{h}', \underline{d}'}{\text{Number of realizations}}. \quad (7)$$

This ensemble mean is however not always computable, especially considering the intended application of benchmarking for modems. These may not be able to provide timing information of packet events. However, without precise timing information, we cannot correlate packet events with the channel parameter state of the emulator. Therefore, a more practical performance metric is used, the packet error ratio (PER). For calculating the PER, we assume stationarity in a given time window w and introduce the PER calculation as

$$\text{PER}_w = \frac{\text{Number of } \mathcal{E} \text{ in } w}{\text{Total number of events in } w}. \quad (8)$$

Since the PER does not require knowledge on the underlying channel, we only expect to know the time frame in which packets were transmitted and received. This is a much more lenient condition than knowing the exact transmission time, since we consider timing accuracy in the order of 100 μ s, while averaging window lengths are at the order of at least 100 ms, up to 10 s. This also means that we can compute a meaningful PER measure on a single realization of the channel. This would not be possible for $P(\mathcal{E})$, because there we require a suitably large number of realizations for the estimate to be good.

Furthermore, if the channel parameters within w are assumed constant, we expect the given channel behavior to be *ergodic*, and the PER is expected to tend towards $P(\mathcal{E})$.

6.2. Nonstationary Model for Street Crossing. In this section, we present the test carried out to benchmark the performance of two SDR-based IEEE 802.11p modems connected to the emulator which is emulating the nonstationary fading model presented in [16]. The model we chose relates to the vehicular scenario: *road crossing: urban single lane* with measured data collected in DRIVEWAY'09 V2V measurement campaign in Lund, Sweden [15]. The duration of the measured scenario is 10 s. As depicted in the general schematics of the scenario in Figure 4, the TX and the RX drive towards one another in two perpendicular streets, they reach one another at the intersection, and then they drive away.

The Power Delay Profile (PDP) of the measurement can be seen in Figure 5(a) plotted over measurement time. In the beginning of the scenario, the only links existing are the Non-Line-of-Sight (NLOS) components reflected from the buildings at two sides of the roads, as two cars approach the

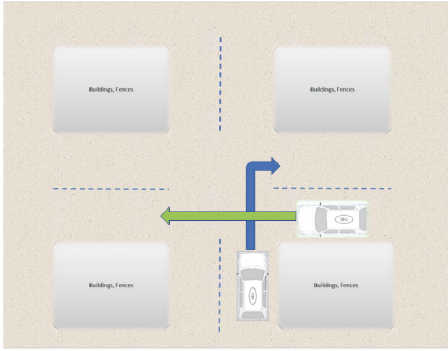


FIGURE 4: Vehicular scenario: urban crossing.

intersection, at 8 s measurement time, there appears a strong component which indicates Line-of-Sight (LOS) path, when the cars pass one another and drive away we observe the change in the Doppler direction [16]. The specific scenario is of particular interest, since not only, similar to other vehicular scenarios, large Doppler and delay spreads are observed, but also the abrupt change from NLOS components as the dominant paths to the LOS component along with the change in the direction of Doppler frequency contribute to the high nonstationarity of the channel. In [16], the channel sounder measured data was estimated to a tapped-delay line impulse response in such a way that the fit is optimal in the bandwidth of interest, chosen to be 20 MHz, to ensure that the 10 MHz of interest lies in the interior of the emulation bandwidth. The trace samples including delays, path-loss, and phase shifts are updated every 302.7 μ s.

6.2.1. Measurement Setup. Two SDR-based 802.11p modems configured as the TX and the RX are connected as described in Figure 3 to the SDR device. When the channel emulator bitfile is loaded in the USRP device and the emulator is ready, the TX modem starts sending the specified number of packets with specified payload. At the RX, we log the number of successfully received packets. Table 1 summarizes the parameter values used for configuring the transmission and reception of packets. The 10 s measurement is converted to a suitable trace beforehand and forwarded to the emulator via the host PC. The trace is looped, and the measurement is run for a total of 10 minutes. A transmission is attempted every 2 ms, and the payload contains an identification number and random data to fill the packet length to the desired length.

The sum received power per snapshot rises exponentially over the 10-second trace and thus covers a large dynamic range. For very small values, we see two effects. First, the tapped-delay line fit itself is dominated by measurement noise; hence the tap locations are varying strongly in the first 5 seconds. Second, the tap values overall are very small, and Figure 5(b) shows visually that, at around 6 seconds, the taps become discernible. This is an issue for the emulator because we have to make sure that the largest value does not clip, and therefore small values may vanish in quantization.

TABLE 1: Measurement setup parameters for nonstationary model test.

Measurement parameter	Value
Frequency channel	180 (5.9 GHz)
Packet length	200, 500, 1000 Bytes
Measurement duration	10 min
Average RX power	-65 dBm
Power level at channel emulator input	-30 dBm
Data rate	6 Mbps
Baseband modulation	QPSK
Coding rate	1/2

6.2.2. Measurement Results. Figure 6 shows the results of the measurements conducted on the nonstationary model. PER evaluations are shown in Figures 6(a), 6(c), and 6(e) and depict PER curves for 6 realizations of the 10-second trace. We calculated them by setting the window length to 200 ms and generating a sliding average. Since the channel repeats 60 times in 10 minutes, we show this sliding average for 6 channel realizations spread out over the measurement time. We want to point out that this is not generated by triggering the channel; we only used the timestamp at transmitter and receiver for aligning the measurement traces. This demonstrates that the emulation timing corresponds perfectly to real-world timing and remains stable over 10 minutes of measurement time. Furthermore, the generated PER curves show very small deviations between realizations. This is expected, as the measurements are conducted at high received powers and consequently additive noise added after the emulator is not expected to contribute. Overall, the PER curves show that the emulator allows perfect repeatability of tests due to the stability of behavior between channel realizations. With this result, we can now concentrate on Figures 6(b), 6(d), and 6(f). There, we calculate $P(\mathcal{E})$ through *ensemble means*. Since the PER analysis shows that the emulator repeats the channel trace at exactly 10 s, we calculate $P(\mathcal{E})$ by averaging over all events that are separated exactly 10 s and do this for all offsets. The graph shows that the ensemble mean follows the trend of the PER but shows more details. For example, at 8.5 s, a short dip in $P(\mathcal{E})$ is shown that is so short that it is averaged out in the PER graph. Thus, we are able to calculate a packet error estimate and also demonstrate that the PER is a good first estimate; however, when treating nonstationary scenarios, it misses fast changes and $P(\mathcal{E})$ should be used.

Overall, the performance in the nonstationary case can be split into 3 regions. For the first 6-7 s, the small channel values dominate, and no packet passes successfully. Then, we see an almost binary dependence on the LOS path. At the exact instant in Figure 5(a) where the LOS path appears, the packet error probability goes from close to 1 down strongly, depending on the packet length. In general, however, we see for all packet lengths the same qualitative behavior, even if the exact values differ. Therefore, for further measurements, we will consider 500 bytes as packet size. We do see a limitation in the dynamic range; however, the regions left and right of

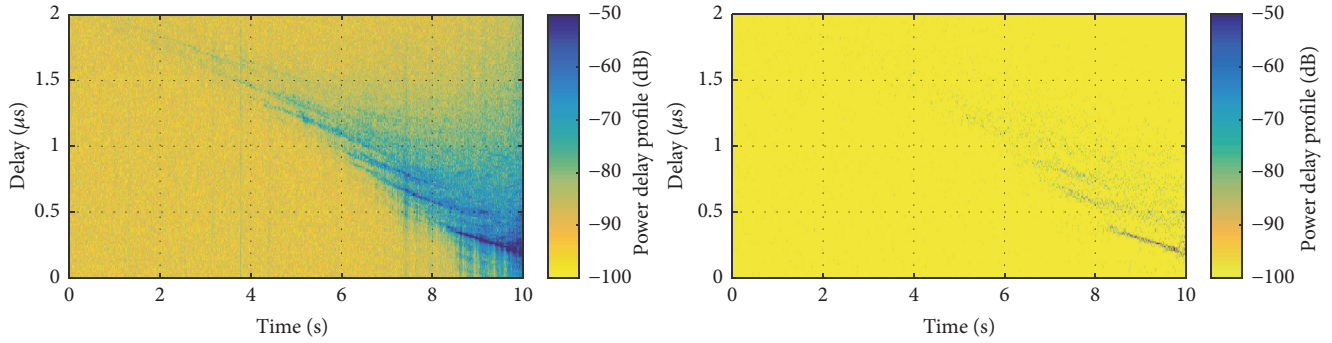


FIGURE 5: Measured PDP and 10-tap LASSO fit (cf. [16]).

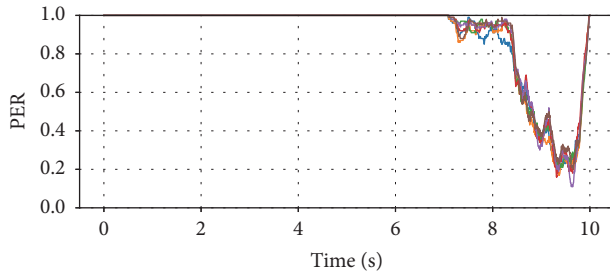
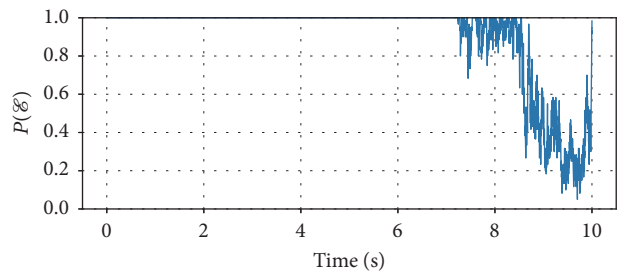
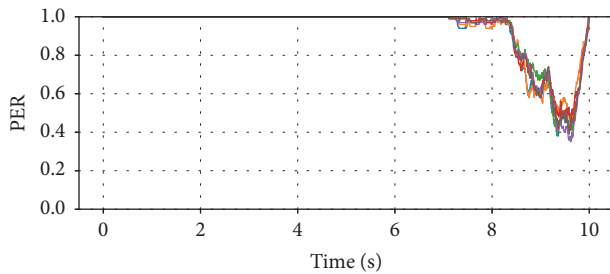
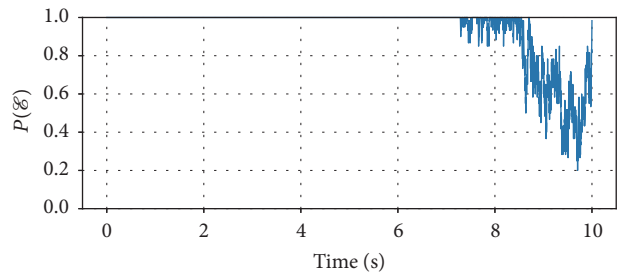
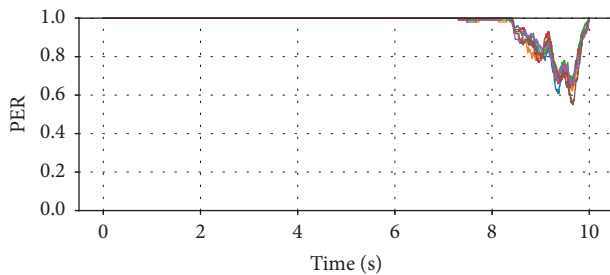
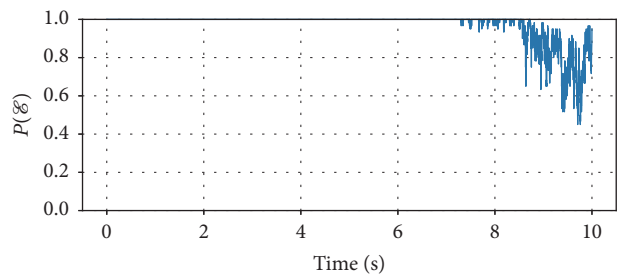

 (a) 200 bytes, $PER_{200\text{ms}}$

 (b) 200 bytes, $P(\xi)$

 (c) 500 bytes, $PER_{200\text{ms}}$

 (d) 500 bytes, $P(\xi)$

 (e) 1000 bytes, $PER_{200\text{ms}}$

 (f) 1000 bytes, $P(\xi)$

 FIGURE 6: Packet error ratio (PER, (a), (c), and (e)) and packet error probability ($P(\xi)$, (b), (d), and (f)) estimates from emulated nonstationary vehicular channel.

6 s still have extremely close packet error probabilities close to 1; hence, we argue that the performance is adequate.

6.3. ETSI Draft Channel Models. Our second goal is to step from playback of measured scenarios to the implementation of standardized scenarios. To achieve this, we address stationary vehicular channel models that were defined in a draft by ETSI specifically with the application in benchmarking

in mind. The given models are summarized in Table 2 [2] and encompass 5 typical vehicular scenarios. Rural traffic is designed as Additive White Gaussian Noise (AWGN) like; furthermore there are the 2 critical areas for vehicular traffic, city, and highways. The urban models show strong multipath components, while the highway models generally have large Doppler spreads. Each model is presented by means of a tapped-delay line with 3 or 4 active taps. The

TABLE 2: ETSI draft channel model parameters [2].

Draft scenarios	Tap	η_p [dB]	τ_p [ns]	ν_p	Spectral shape
Rural LOS	$p = 1$	0	0	0	Static
	$p = 2$	-14	83	492	HalfBT
	$p = 3$	-17	183	-295	HalfBT
Urban LOS	$p = 1$	0	0	0	Static
	$p = 2$	-3	267	295	HalfBT
	$p = 3$	-4	400	-98	HalfBT
	$p = 4$	-10	533	591	HalfBT
Urban NLOS	$p = 1$	0	0	0	Static
	$p = 2$	-8	117	236	HalfBT
	$p = 3$	-10	183	-157	HalfBT
	$p = 4$	-15	333	492	HalfBT
Highway LOS	$p = 1$	0	0	0	Static
	$p = 2$	-10	100	689	HalfBT
	$p = 3$	-15	167	-492	HalfBT
	$p = 4$	-20	500	886	HalfBT
Highway NLOS	$p = 1$	0	0	0	Static
	$p = 2$	-2	200	689	HalfBT
	$p = 3$	-5	433	-492	HalfBT
	$p = 4$	-7	700	886	HalfBT

first tap is defined as static with delay at zero and zero Doppler frequency to serve as normalization reference; the remaining taps are considered to be Rayleigh distributed: each is represented with its delay τ_p , with η_p and ν_p as deterministic path-loss and maximal Doppler shift of the Doppler spectrum, respectively, along with a spectral shape of the Power Spectral Density (PSD). To prepare the channel trace, we first quantized the delay values to the sampling time of the USRP and determined the active taps. Secondly, the stochastic trace for the Rayleigh fading was generated from a Half Jake's (aka Half-Bathtub) PSD with sampling rate of $100 \mu\text{s}$ [2]. The Half-Bathtub sequence was formed by generating a typical Jake's Bath tub sequence with maximum Doppler frequency and then setting one-half of Doppler spectrum to zero according to the sign of the Doppler frequency.

6.3.1. Measurement Setup. We use the same transmitter-receiver setup as introduced in Section 6.2.1. The chief difference here is that we use a 4-tap configuration according to the channel models. Furthermore, we only analyze 500-byte packet sizes.

6.3.2. Measurement Results. The results of the stationary measurements are shown in Figure 7. Figures 7(a), 7(c), 7(e), 7(g), and 7(i) show the 200 ms PER averages. In the stationary case, these vary around a constant value. The variation that we see is due to the fact that the averaging window is short enough to still let the small-scale fading variations be visible. They are seen strongest in the Urban NLOS case, which has strong multipath components and long correlation times due to low Doppler speeds. Again, the overlaid traces show that we are able to provide a repeatable channel with repeatable

characteristics. As shown in [2], rural LOS corresponds to almost perfect transmission, and highway NLOS to a $P(\mathcal{E})$ of almost 1. Switching to Figures 7(b), 7(d), 7(f), 7(h), and 7(j), we calculate the $P(\mathcal{E})$ estimates again. The stationary results confirm our nonstationary results, as the actual $P(\mathcal{E})$ shows more pronounced short-term variations that are averaged out in the PER calculation. The results show that the fading impact can be seen in the variation of the packet error probability, and cannot be neglected a priori. However, disregarding this aspect, the measurements demonstrate an important property. In the stationary case, we are able to make the averaging window for the PER arbitrarily large since the statistics cannot change. If w goes to infinity, PER_w and $P(\mathcal{E})$ averaged over w will converge to the same value, since even for 10 s traces, their shapes are very similar. Thus, if we are only interested in a global error probability in a single value, we can calculate the PER directly using a single, sufficiently long realization, without resorting to analysis of the ensemble means. While this result is trivial for stationary channels in theory, it is very important that the hardware implementation on our emulator actually follows this and behaves as expected. Therefore, we are now able to benchmark modems via calculating the PER, which means we only need access to the total number of transmissions and receptions.

6.4. Benchmarking. In this section, we demonstrate the use of the emulator for benchmarking purposes. For this, we want to compare the performance of the USRP-Rio application framework with a COTS modem. The COTS modem under test is designed to work as RSU under normal conditions. As was established previously, we do this by calculating the PER via the total number of transmissions and the numbers

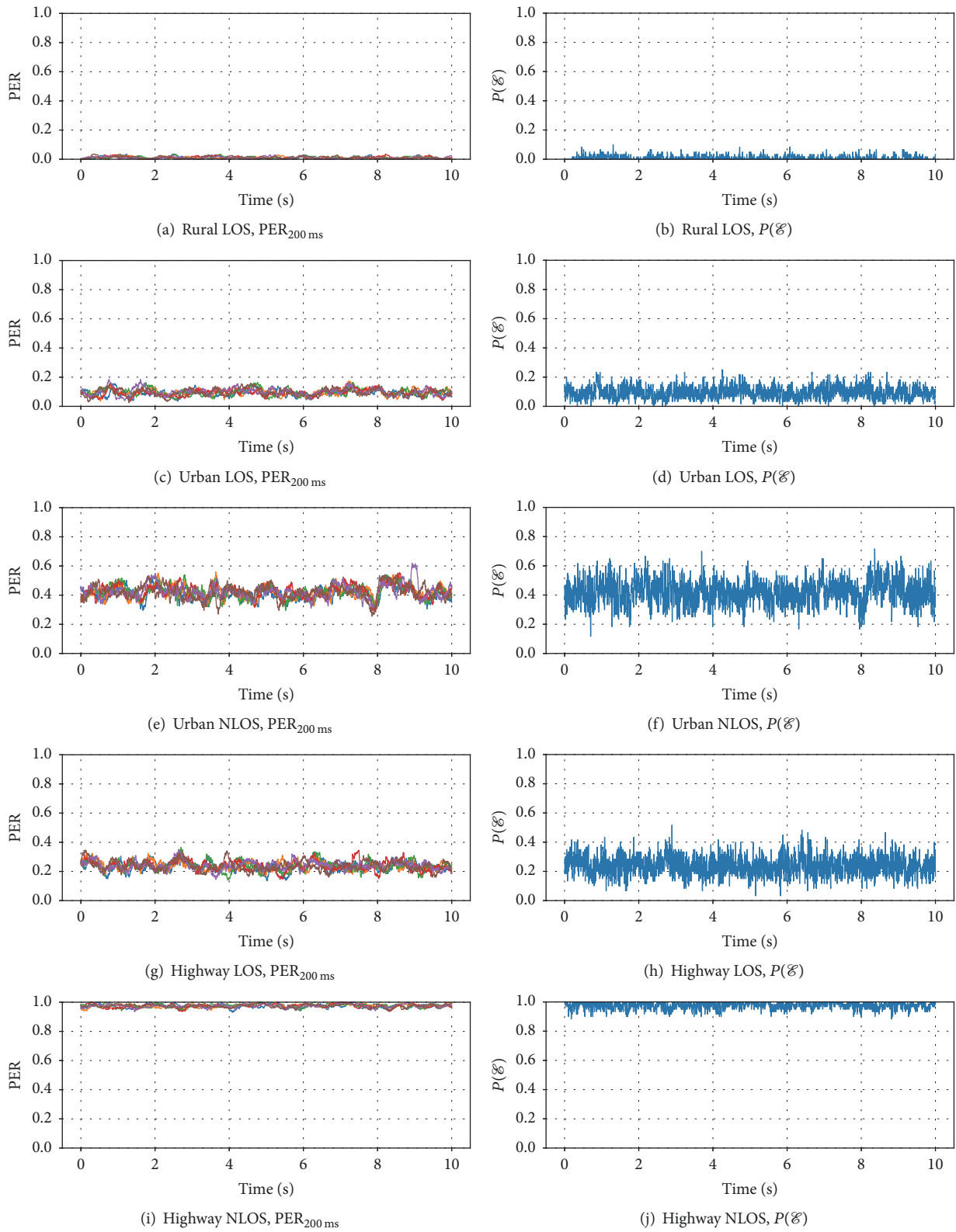


FIGURE 7: Packet error ratio ((a), (c), (e), (g), and (i)) and packet error probability ((b), (d), (f), (h), and (j)) estimates from emulated stationary vehicular channel.

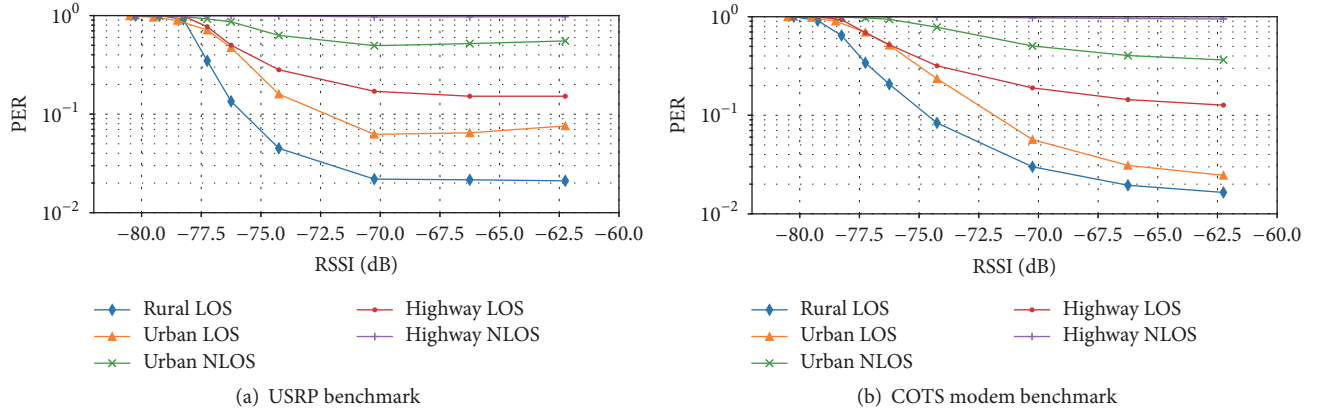


FIGURE 8: Benchmark analysis of USRP and COTS Modem.

TABLE 3: Measurement setup parameters for ETSI draft channel models.

Measurement parameter	Value
Frequency channel	180 (5.9 GHz)
Packet length	500 bytes
Number of packets (=frames) in one burst	50000
Burst repetitions	1
Data rate	6 Mbps
Baseband modulation	QPSK
Coding rate	1/2

of packets that are lost. At this point, this is vital because the modem does not provide more detailed statistics.

6.4.1. Measurement Setup. Tests were performed using two pairs of IEEE 802.11p modems: a pair of COTS modems and a pair of SDR-based modems implemented on USRP using the modified version of NI IEEE 802.11 stack while the traces for each scenario model were streamed to the emulator. We benchmarked PER in both pairs of modems for various Received Signal Strength Indicator (RSSI) points. In order to vary RSSI, a RADITEK RVATTN-DC-6 (0–70 dB) switched attenuator was placed between transmitter and receiver with 1 dB resolution (see Figure 3). After configuring the emulator and receiver, the transmitting modem starts sending the specified number of packets with specified payload of length 500 bytes. At the receiver, we get the number of correctly decoded packets which is then used for the calculation of PER. Table 3 summarizes the parameter values used for configuring the transmission and reception of packets. For the comparison against the USRP-Rio, we are able to use the setup we employed previously.

6.4.2. Measurement Results. The results of the benchmarking process are collected in Figure 8. The results for the USRP are shown in Figure 8(a), while the COTS results are depicted in Figure 8(b). Both setups have been tested under all 5 stationary channel models, for packet lengths of 500 bytes. The figures show a strong spread in performance for the

different models, which is consistent across both devices under test. The main difference is the fact that the USRP hits a hard error floor for all models and even deteriorates for high RSSIs in some models. This is due to the Automated Gain Control (AGC) loop not setting ideal values for the subsequent Analog-Digital Converter (ADC), introducing additional noise in that region. The modem still shows small improvements in the high-RSSI regions. Qualitatively, however, the devices show highly comparable performance. This is important, since it simultaneously shows that the emulator is validated in its performance against devices of completely different build, and the given channel models show on 2 real-world devices a spread that facilitates assessment of further devices under test.

7. Conclusion

This work reports on the real-time emulation of V2X channels for benchmarking vehicular modems. V2X channels exhibit non-WSSUS behavior in safety-relevant vehicular scenarios and our emulation approach respects this by allowing real-time updates to the multipath propagation model for each local stationarity region. Two performance metrics are compared: the packet error ratio (PER) estimated from *temporal averaging* over a *single* virtual drive and the packet error probability (PEP) estimated from *ensemble averaging* over *repeated* virtual drives. Two categories of vehicular channel models are used in the benchmarking: a vehicular nonstationary channel model derived for street crossing: urban single lane scenario in the DRIVEWAY'09 measurement campaign and the stationary ETSI models. To show the viability, the proposed architecture is realized as a virtual instrument on National Instruments LabVIEW and running on the USRP-Rio 2953R platform.

The modem performance measurements carried out with the emulator show that the proposed architecture successfully balances design simplicity with emulation fidelity. Emulation playback of a measured channel is demonstrated, and the results show the strong time-dependence of the resulting packet error performance. Our setup allows time resolutions in the order of milliseconds, demonstrating a fine-grained

analysis of the underlying channel in terms of packet throughput. We see this through the extremely short peaks that are only visible by *ensemble averaging* and are lost when computing a classical PER. To be able to do this, we need high time-stability, which our emulator provides. We see the influence of the packet length through the fact that small packets are able to capitalize on temporarily strong channels. The same analysis for the stationary case confirms the results from the nonstationary case. Our results show the limits and validity of PER analysis. Based on these limits, we benchmark two modem combinations and present the resulting performance, providing data for further benchmarking. The stability of our emulator however means that the introduced setup is capable of deriving more complex statistics, such as ensemble means and ensemble means, given that at least the receiver is able to record event timestamps. Hence, the introduced setup allows flexible and accurate performance evaluation of V2X hardware on a variety of stationary and nonstationary channel, including the playback of recorded channels.

Disclosure

The full work presented in this paper was conducted at the Institute of Telecommunications, TU Wien, Vienna, Austria.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors acknowledge the TU Wien University Library for financial support through its Open Access Funding Program. Furthermore, they acknowledge the CA15104 IRACON COST action for facilitating the exchange of knowledge.

References

- [1] C. F. Mecklenbräuker, A. F. Molisch, J. Karedal et al., "Vehicular channel characterization and its implications for wireless system design and performance," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, 2011.
- [2] T. Blazek, M. Ashury, C. F. Mecklenbräuker, D. Smely, and G. Ghiaasi, "Vehicular channel models: a system level performance analysis of tapped delay line models," in *Proceedings of the 15th International Conference on ITS Telecommunications (ITST '17)*, May 2017.
- [3] L. Bernado, T. Zemen, F. Tufvesson, A. F. Molisch, and C. F. Mecklenbräuker, "Delay and doppler spreads of nonstationary vehicular channels for safety-relevant scenarios," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 82–93, 2014.
- [4] G. Matz, "On non-WSSUS wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2465–2478, 2005.
- [5] *IEEE Std. 802.11p-2010, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Wireless Access in Vehicular Environment*, IEEE Std, IEEE, 2010.
- [6] "ETSI TR 102 638—intelligent transportation systems (ITS); vehicular communications; basic set of applications; definitions, ETSI, Tech".
- [7] A. Paier, F. Tufvesson, and M. Nilsson, "G5 Radio Channel Models (Version 4)", *Car-2-Car Communication Consortium, Architecture Meeting Ulm*, 2013.
- [8] A. F. Molisch, F. Tufvesson, J. Karedal, and C. F. Mecklenbräuker, "A survey on vehicle-to-vehicle propagation channels," *IEEE Wireless Communications Magazine*, vol. 16, no. 6, pp. 12–22, 2009.
- [9] "Propsim channel emulation solutions," <http://www.anite.com>.
- [10] Spirent Communications, 2009, <http://www.spirent.com>.
- [11] T. M. Fernández-Caramés, M. González-López, and L. Castedo, "FPGA-based vehicular channel emulator for evaluation of IEEE 802.11p transceivers," in *Proceedings of the 2009 9th International Conference on Intelligent Transport Systems Telecommunications (ITST '09)*, pp. 592–597, October 2009.
- [12] G. Acosta-Marum and M. A. Ingram, "Six time- and frequency—selective empirical channel models for vehicular wireless LANs," *IEEE Vehicular Technology Magazine*, vol. 2, no. 4, pp. 11–44, 2007.
- [13] D. Vlastaras, S. Malkowsky, and F. Tufvesson, "Stress test of vehicular communication transceivers using software defined radio," in *Proceedings of the 81st IEEE Vehicular Technology Conference (VTC Spring '15)*, pp. 6–9, Boston, MA, USA, May 2015.
- [14] "National Instruments™ USRP-2953R," <http://sine.ni.com/nips/cds/view/p/lang/en/nid/213005>.
- [15] A. Paier, L. Bernadó, J. Karedal, O. Klemp, and A. Kwoczek, "Overview of vehicle-to-vehicle radio channel measurements for collision avoidance applications," in *Proceedings of the 2010 IEEE 71st Vehicular Technology Conference (VTC-Spring '10)*, Taipei, Taiwan, May 2010.
- [16] T. Blazek and C. F. Mecklenbräuker, "Sparse Time-Variant Impulse Response Estimation for Vehicular Channels Using the c-LASSO," in *Proceedings of the IEEE 28th PIMRC - Workshop WS-II on V2X Channel Measurements and Modeling (WVCM '17)*, IEEE, 2017, <https://publik.tuwien.ac.at/showentry.php?ID=261919&lang=2>.
- [17] G. Ghiaasi, M. Ashury, D. Vlastaras, M. Hofer, Z. Xu, and T. Zemen, "Real-time vehicular channel emulator for future conformance tests of wireless ITS modems," in *Proceedings of the 10th European Conference on Antennas and Propagation (EuCAP '16)*, Davos, Switzerland, April 2016.
- [18] Z. Xu, M. Gan, C. F. Mecklenbräuker, and T. Zemen, "Cluster spreads for time-variant vehicular channels," in *Proceedings of the 9th European Conference on Antennas and Propagation (EuCAP '15)*, Lisbon, Portugal, April 2015.
- [19] Z. Xu, M. Gan, and T. Zemen, "Cluster-based non-stationary vehicular channel model," in *Proceedings of the 10th European Conference on Antennas and Propagation (EuCAP '16)*, Davos, Switzerland, April 2016.



Hindawi

Submit your manuscripts at
www.hindawi.com

