# Real-Time Channel Emulation of a Geometry-Based Stochastic Channel Model on a SDR Platform

Markus Hofer, Zhinan Xu, Thomas Zemen
AIT Austrian Institute of Technology, Vienna, Austria
Donau-City-Straße 1, A-1220 Vienna
Email: {markus.hofer, xu.zhinan, thomas.zemen}@ait.ac.at

***Abstract*—In wireless vehicular communication, channel properties change rapidly over time. Both, the transmitter and the receiver, are moving, which generates not only time and frequency (doubly) selective channels but also channel statistics that are non-stationary, i.e., they change over time. New wireless vehicular communication systems for connected autonomous vehicles require validation and verification in vehicular environments to assure their proper functionality. To avoid time intensive, costly and difficult to repeat real-world measurements on the road, real-time channel emulators that target on emulating the wireless vehicular channel as accurately as possible, are needed. In this paper, we present a real-time channel emulator based on a software defined radio platform that is able to emulate real-valued path delays and Doppler shifts within a certain delay and Doppler region. The emulator uses a low-complexity subspace expansion model where the emulation complexity on the field programmable gate array (FPGA) is *independent* from the number of propagation paths. This makes it suitable to emulate realistic geometry-based non-stationary channel models with a large number of propagation paths.**

***Index Terms*—real-time, channel emulation, SDR, geometry-based stochastic channel model, FPGA**

## I. Introduction

In recent years the introduction of intelligent transportation systems (ITSs) [1] has given rise to the development of new wireless communication systems. Vehicle-to-vehicle communication based on IEEE 802.11p [2] is used to exchange cooperative awareness messages in order to improve the safety and the efficiency on the road. Connected autonomously driving vehicles will use vehicular communication links to exchange sensor information for improved road safety. To assure the correct functionality of these communication systems they have to be tested and validated thoroughly on the road, however, this is not only time consuming, but also costly and difficult to repeat. Hence, channel emulators that aim at emulating real world scenarios as accurately as possible, are urgently needed.

Besides commercially available channel emulation solutions from Spirent [3] and Anite [4] also software defined radio (SDR) based channel emulator solutions [5–10] have been introduced. While the channel emulator solution developed by National Instruments (NI) [5] utilizes a vector signal transceiver to emulate a MIMO channel based on a tapped delay line (TDL), Vlastaras et al. [6] implement a TDL with two active taps and equal power for stress testing IEEE 802.11p modems. Ghiaasi et al. [7] use a TDL in connection with a clustering algorithm to reduce the number of propagation paths that have to be simultaneously emulated. The maximum number of simultaneously active delay taps is ten. The channel emulator shown by Mkg systems [8] allows for a maximum number of twelve delay taps.

The main drawback of these emulator solutions is that the implemented models are limited to delays that are integer multiples of the sampling rate. Real world communication scenarios, however, allow for any real valued delay and Doppler shift. Furthermore, the channel statistics may change over time [11].

*Scientific contribution*: In this paper we use a low-complexity reduced-rank subspace model that was introduced by Kaltenberger et. al. [12–14] to emulate channel models with real-valued path delay and Doppler shift within a certain delay and Doppler region. We split the algorithm shown in [12] into two one-dimensional reconstructions, such that the algorithm can be implemented on a SDR platform. To the best of our knowledge, this paper shows the first time how such an emulation approach is implemented on a SDR platform. This allows for a flexible, low-cost, highly performant channel emulator. We describe the architecture of the channel emulator and explain the corresponding building blocks in detail.

## II. Emulator Architecture

To emulate the effects of wireless wave propagation, a real-time emulator convolves the input signal with the time-variant channel impulse response. The architecture is shown in Fig. 1. The emulator consists of a host computer that is equipped

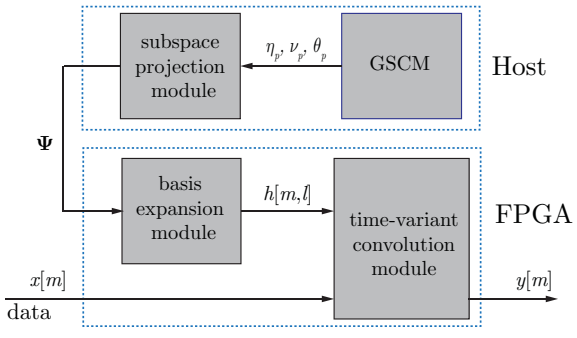


Fig. 1. Basic architecture of the channel emulator

978-1-5090-3009-5/17/$31.00 

with a PCI express card to communicate with the SDR. The time-variant channel impulse response is directly calculated on the FPGA by utilizing a reduced-rank basis-expansion model that was shown in [12–14]. For the calculation only the basis coefficients have to be transmitted from the host to the SDR which allows for a significant bandwidth reduction.

On the host a geometry-based stochastic channel model (GSCM) generates $P$ propagation paths where each propagation path has a complex weighting coefficient $\eta_p$, normalized path delay $\theta_p$ and normalized Doppler shift $\nu_p$. The parameters of the propagation paths are the input to a subspace projection module that calculates the generalized basis coefficients matrix $\mathbf{\Psi}$. The basis coefficients are streamed to the FPGA via direct memory access (DMA) first-input first-output (FIFO) buffers, where a basis-expansion module utilizes the coefficients to calculate the time-variant channel impulse response $h[m,l]$. Subsequently, the time-variant channel impulse response is convolved with the input signal $x[m]$ in the time-variant convolution module to obtain the output signal $y[m]$.

As SDR we utilize a NI USRP-2953R [15] that is equipped with an Xilinx Kintex-7 FPGA.

## III. System Model

### A. Geometry-Based Stochastic Channel Model

To allow for a realistic modelling of real world propagation scenarios we assume that the channel propagation parameters originate from a GSCM [16]. The non-stationary time-variant frequency response [17, 18]

$$g_{\text{Ph}}(t,f) = \sum_{p=0}^{P-1} \eta_p(t) \mathrm{e}^{-\mathrm{j}2\pi\tau_p(t)f} \tag{1}$$

can be described as the superposition of $P$ individual paths, given $P$ is sufficiently large. Here, $g_{\text{Ph}}(t,f)$ denotes the non band-limited physical channel. Each path is characterized by the complex time-variant weighting coefficient $\eta_p(t) = \beta_p(t)\mathrm{e}^{\mathrm{j}2\pi\phi_p}$ with amplitude $\beta_p(t)$, initial phase $\phi_p$ and real-valued time-variant path delay $\tau_p(t)$.

The non-stationary fading process can be approximated as wide-sense stationary inside a stationary region that is defined by the stationary time $T_{\text{stat}}$ and the stationary bandwidth $B_{\text{stat}}$ [11]. We assume that within a stationary region the complex weighting coefficients $\eta_p(t)$ do not change for $t_1 \leq t < t_1 + T_{\text{stat}}$, i.e., $\eta_p(t) \approx \eta_p$ and that the relative velocity between transmitter and receiver stays constant. Hence, we model the time-variant path delay as

$$\tau_p(t) = \tau_p(0) - \frac{f_p}{f} t, \tag{2}$$

where $\tau_p(0)$ denotes the initial path delay that is determined by the distance between transmitter and receiver and $f_p$ denotes the Doppler shift of path $p$. The Doppler shift is defined as

$$f_p = f_c \frac{v\cos(\alpha_p)}{c_0}, \tag{3}$$

where $f_c$ denotes the carrier frequency of the system, $v$ denotes the relative velocity between transmitter and receiver, $\alpha_p$ is the angle of arrival of the $p-$th path [18] and $c_0$ denotes the speed of light.

The discrete time-variant channel frequency response is obtained by sampling $g_{\text{Ph}}(t,f)$ at rate $T_C = 1/B$ in the time direction and at rate $F_s = B/N$ in the frequency direction. Here, $B$ denotes the system bandwidth that is chosen smaller than $B_{\text{stat}}$ and $N$ denotes the number of bins in frequency direction. Considering a band-limiting filter $g_R(f)$ and using (2) the discrete time-variant channel frequency response is obtained by

$$\begin{aligned} g[m,q] :&= g_R(qF_s) g_{\text{Ph}}(mT_C, qF_s) \\ &= g_R[q] \sum_{p=0}^{P-1} \eta_p \mathrm{e}^{-\mathrm{j}2\pi\theta_p q} \mathrm{e}^{\mathrm{j}2\pi\nu_p m}, \end{aligned} \tag{4}$$

where $\nu_p = f_p T_C$ denotes the normalized Doppler shift and $\theta_p = \tau_p(0)/(NT_C)$ denotes the normalized path delay, with $|\nu_p| < \frac{1}{2}$ and $0 < \theta_p < 1$, respectively.

### B. Subspace Projection Module

The channel model presented in (4) is of high computational complexity since for every propagation path and every time and frequency instant a sum of complex exponentials (SoCE) has to be evaluated. Its complexity for a stationary region $M = \lceil T_{\text{stat}}/T_C \rceil$ and $N$ frequency bins is growing with $\mathcal{O}(MNP)$. This renders a straightforward implementation on an FPGA difficult since the complexity ultimately depends on the number of propagation paths that are considered.

Alternatively, we implement an approximate channel model that exploits the limited accuracy of the fixed point (FP) precision on the FPGA. Firstly, we approximate the SoCE by a reduced-rank basis-expansion model (BEM) [12] that uses discrete prolate spheroidal (DPS) sequences [19] as basis functions. The accuracy of the BEM depends on the number of basis functions that are utilized. Secondly, we calculate the basis coefficients as approximate projections onto the basis functions [12] which allows an efficient calculation of the basis-expansion coefficients. The accuracy of the two approximations must not be worse than the accuracy obtained by the FP precision of the FPGA.

The reduced-rank BEM uses two assumptions. Firstly, it assumes that the channel is reconstructed block-wise for limited time-frequency snapshots defined by the Cartesian index set $\mathcal{I} = I_t \times I_f = [0,\ldots,M-1] \times [0,\ldots,N-1]$. This assumption matches well with the size of the local stationarity regions. Secondly, it assumes that $\nu_p$ and $\theta_p$ are limited to a band-limited region that is defined by the Cartesian product $\mathcal{W} = W_t \times W_f = [-\nu_{\text{Dmax}}, \nu_{\text{Dmax}}] \times [0, \theta_{\text{Pmax}}]$. Here, $\nu_{\text{Dmax}} = f_c v_{\text{max}}/c_0$ where $v_{\text{max}}$ denotes the maximum relative velocity between transmitter and receiver and $\theta_{\text{Pmax}} = \tau_{\text{max}}/(NT_C)$ where $\tau_{\text{max}}$ is the maximum expected path delay of the channel.

In case the band-limited region $\mathcal{W}$ and the index-limited region $\mathcal{I}$ can be written as Cartesian products the time-variant transfer function can be represented by two-dimensional DPS

sequences [12, 17, 19, 20]. They are obtained by the Kronecker product of one-dimensional DPS sequences in time and frequency direction according to $\mathbf{s}_d(\mathcal{W},\mathcal{I}) = \mathbf{u}_i(W_t,I_t) \otimes \mathbf{v}_k(W_f,I_f) \in \mathbb{C}^{MN\times 1}$ with their corresponding eigenvalues $\lambda_d(\mathcal{W},\mathcal{I}) = \lambda_i(W_t,I_t)\lambda_k(W_f,I_f)$. Here, $\otimes$ denotes the Kronecker product, $\mathbf{u}_i(W_t,I_t) \in \mathbb{R}^{M\times 1}$ denotes the $i-$th DPS sequence in time direction defined by the Doppler support $W_t$ and the index set $I_t$ and $\lambda_i(W_t,I_t)$ denotes its corresponding eigenvalue. Similarly, $\mathbf{v}_k(W_f,I_f) \in \mathbb{C}^{N\times 1}$ denotes the $k-$th DPS sequence in frequency direction defined by the delay support $W_f$ and the index set $I_f$ with $\lambda_k(W_f,I_f)$ its corresponding eigenvalue.

Defining $\mathbf{g} = [g[0,0], g[0,1], \ldots, g[M-1,N-1]]^T \in \mathbb{C}^{MN\times 1}$, with $(\cdot)^T$ denoting the transpose, the model in (4) can therefore be approximated by [12, 17, 20]

$$\hat{\mathbf{g}} = \mathbf{S}\boldsymbol{\alpha}. \tag{5}$$

The matrix $\mathbf{S} = [\mathbf{s}_0(\mathcal{W},\mathcal{I}), \mathbf{s}_1(\mathcal{W},\mathcal{I}), \ldots, \mathbf{s}_{D-1}(\mathcal{W},\mathcal{I})] \in \mathbb{C}^{MN\times D}$ contains the first $D$ two-dimensional eigenvectors that are sorted such that their eigenvalues are in descending order, i.e., $\lambda_0(\mathcal{W},\mathcal{I}) \geq \lambda_1(\mathcal{W},\mathcal{I}) \geq \ldots \geq \lambda_{|\mathcal{I}|-1}(\mathcal{W},\mathcal{I})$. Furthermore, $\boldsymbol{\alpha} \in \mathbb{C}^{D\times 1}$ denotes the vector of basis coefficients and can be calculated by

$$\boldsymbol{\alpha} = \mathbf{S}^H\mathbf{g} \tag{6}$$

with $(\cdot)^H$ denoting the conjugate transpose.

The dimension $D$ determines the accuracy of the reduced-rank BEM. The bias of the reduced-rank BEM is calculated by $\text{bias}^2_{\hat{\mathbf{g}}^D} = \mathbb{E}\left\{\frac{1}{MN}||\mathbf{g}-\hat{\mathbf{g}}||^2\right\}$ [12, 20], where $\mathbb{E}(\cdot)$ denotes the expectation operator. If Doppler shifts $\nu_p$ and delays $\theta_p$ are independently and uniformly distributed on $W_t$ and $W_f$, respectively, it can be shown that

$$\text{bias}_{\hat{\mathbf{g}}^D} = \frac{1}{|\mathcal{W}||\mathcal{I}|}\sum_{d=D}^{NM-1}\lambda_d(\mathcal{W},\mathcal{I}). \tag{7}$$

The dimension $D$ has to be chosen such that $\text{bias}^2_{\hat{\mathbf{g}}^D} \leq \text{E}_{\text{th}}$ and $\text{E}_{\text{th}}$ is the targeted bias that is defined by the FP precision of the FPGA.

Since the basis functions for reconstruction do not change, matrix $\mathbf{S}$ can be pre-calculated and stored on the FPGA. However, this requires a large memory. Alternatively, the Kronecker product can be calculated on the FPGA which on the other hand is computationally expensive. For an efficient implementation, in our paper we therefore utilize a separate one-dimensional reconstruction approach. Concretely, the model in (4) can be written as [12, 17, 20]

$$\hat{g}[m,q] \approx \sum_{p=0}^{P-1}\eta_p\sum_{k=0}^{D_f-1}v_k[q]\epsilon_{k,p}\sum_{i=0}^{D_t-1}u_i[m]\gamma_{i,p} \tag{8}$$

where $D_t$ and $\gamma_{i,p}$ and $D_f$ and $\epsilon_{k,p}$ denote the number of DPS sequences and the basis coefficient of path $p$ in time and frequency direction, respectively. Although this approach is not optimal in terms of required dimensions, because of its double-sum structure it offers an efficient implementation on the FPGA with reduced hardware utilization and equivalent reconstruction accuracy.

The dimensions $D_t$ and $D_f$ determine the accuracy of the reduced-rank subspace approximation and similar to (7) it can be shown that [13]

$$\text{bias}^2_{\hat{\mathbf{g}}^{D_t}} = \frac{1}{2\nu_{\text{Dmax}}M}\sum_{d=D_t}^{M-1}\lambda_d(W_t,I_t) \tag{9}$$

$$\text{bias}^2_{\hat{\mathbf{g}}^{D_f}} = \frac{1}{\theta_{\text{Pmax}}N}\sum_{d=D_f}^{N-1}\lambda_d(W_f,I_f) \tag{10}$$

where $\lambda_d(W_t,I_t)$ and $\lambda_d(W_f,I_f)$ are the sorted eigenvalues of the one dimensional DPS sequences in time and frequency direction, respectively. For the reconstruction we set $D_t$ and $D_f$ such that $\text{bias}^2_{\hat{\mathbf{g}}^{D_t}} \leq \text{E}_{\text{th}}$ and $\text{bias}^2_{\hat{\mathbf{g}}^{D_f}} \leq \text{E}_{\text{th}}$, respectively.

The basis coefficients $\gamma_{i,p}$ and $\epsilon_{i,p}$ of each path can be calculated approximately by scaled and shifted approximate DPS wave functions in $\mathcal{O}(1)$ operations as shown in [12]. This allows for a strong computational complexity reduction of the projection. The accuracy of the projection depends on the resolution factor $r$ of the DPS sequences [12] and increases with a higher resolution factor. The resolution factors have to be chosen such that the obtained accuracy is better than the FP resolution of the FPGA. We refer the reader to [12, 14] for a more detailed discussion of the resolution factor.

With the approximate basis coefficients $\tilde{\gamma}_{i,p}$ and $\tilde{\epsilon}_{i,p}$ we develop (8) further as

$$\begin{aligned}\tilde{g}[m,q] &= \sum_{k=0}^{D_f-1}v_k[q]\sum_{i=0}^{D_t-1}u_i[m]\sum_{p=0}^{P-1}\eta_p\tilde{\gamma}_{i,p}\tilde{\epsilon}_{k,p}\\ &= \sum_{i=0}^{D_t-1}\sum_{k=0}^{D_f-1}u_i[m]v_k[q]\tilde{\psi}_{i,k}\end{aligned} \tag{11}$$

where we defined the approximate generalized basis coefficients

$$\tilde{\psi}_{i,k} = \sum_{p=0}^{P-1}\eta_p\tilde{\gamma}_{i,p}\tilde{\epsilon}_{k,p}. \tag{12}$$

We see that the computational complexity of the reduced-rank BEM in (11) is *independent* of the number of propagation paths. Thus it is very suitable for an implementation on the FPGA.

### C. Complexity Assessment

Utilizing the scaled and shifted DPS wave functions allows for an efficient calculation of the approximate generalized basis-coefficients $\tilde{\psi}_{i,k}$ on the host computer. The complexity of the calculation is $\mathcal{O}(D_tD_fP)$ compared to $\mathcal{O}(MNP)$ for the SoCE.

### D. Time-Variant Channel Impulse Response - Base Expansion Module

In order to perform the time-variant convolution

$$y[m] = \sum_{l=0}^{L-1}\tilde{h}[m-l,l]x[m-l] \tag{13}$$

on the FPGA the time-variant channel impulse response $\tilde{h}[m,l]$ with $l \in \{0,\dots,L-1\}$ has to be calculated. The number of delay taps $L \leq N$ depends on the maximum expected path delay and the bandwidth and is defined by $L = \lceil \tau_{\max} B \rceil$. We obtain the time-variant channel impulse response by applying an inverse fast Fourier transform (IFFT) to the approximate time-variant transfer function $\tilde{g}[m,n]$, i.e., defining $\tilde{\mathbf{g}}[m] = [\tilde{g}[m,0],\dots,\tilde{g}[m,N-1]]^T \in \mathbb{C}^{N\times 1}$ the approximate time-variant channel impulse response $\tilde{\mathbf{h}}[m] = \left[\tilde{h}[m,0],\dots,\tilde{h}[m,L-1]\right]^T \in \mathbb{C}^{L\times 1}$ can be calculated by

$$\tilde{\mathbf{h}}[m] = \mathbf{D}^H \tilde{\mathbf{g}}[m]. \tag{14}$$

Here, $\mathbf{D} \in \mathbb{C}^{N\times L}$ is a $N \times L$ submatrix of the fast Fourier transform (FFT) matrix $[\mathbf{W}]_{i,j} = \frac{1}{\sqrt{N}} e^{\frac{-j2\pi(i-1)(j-1)}{N}} \in \mathbb{C}^{N\times N}$ and $\forall i,j \in \{1,\dots,N\}$. Calculating the IFFT on the FPGA needs resources and time that we want to save. Thus we consider a reformulation of the subspace model in (11). We firstly define $\mathbf{f}^t[m] = \left[u_0[m], u_1[m],\dots,u_{D_t-1}[m]\right]^T \in \mathbb{R}^{D_t\times 1}$ and

$$\tilde{\mathbf{\Psi}} = \begin{pmatrix} \tilde{\psi}_{0,0} & \cdots & \tilde{\psi}_{0,D_f-1} \\ \vdots & \ddots & \vdots \\ \tilde{\psi}_{D_t-1,0} & \cdots & \tilde{\psi}_{D_t-1,D_f-1} \end{pmatrix} \in \mathbb{C}^{D_t\times D_f} \tag{15}$$

and reformulate (11) in vector-matrix notation

$$\tilde{\mathbf{g}}[m] = \mathbf{V}\tilde{\mathbf{\Psi}}^T \mathbf{f}^t[m] \tag{16}$$

where $\mathbf{V} = \left[\mathbf{v}_0, \mathbf{v}_1,\dots,\mathbf{v}_{D_f-1}\right] \in \mathbb{C}^{N\times D_f}$ is the basis vector matrix in the frequency domain. Utilizing (14) we obtain

$$\tilde{\mathbf{h}}[m] = \mathbf{D}^H \tilde{\mathbf{g}}[m] = \underbrace{\mathbf{D}^H \mathbf{V}}_{\mathbf{V}_t} \tilde{\mathbf{\Psi}}^T \mathbf{f}^t[m] = \mathbf{V}_t \tilde{\mathbf{\Psi}}^T \mathbf{f}^t[m] \tag{17}$$

with $\mathbf{V}_t = \mathbf{D}^H\mathbf{V} \in \mathbb{C}^{L\times D_f}$. Since $\mathbf{V}$ and $\mathbf{D}$ do not change, the matrix $\mathbf{V}_t$ can be pre-calculated and stored on the FPGA. Therefore, the IFFT to obtain $\tilde{h}[m,l]$ can be directly in-cooperated in the channel reconstruction of the basis-expansion module.

## IV. Emulation Results

For the test of the real-time channel emulator we stream the approximated generalized basis coefficients for each stationary region, denoted as frame further on, from the host to the FPGA. The coefficients for the next frame are calculated on the host and then streamed to and stored on the FPGA at the beginning of the current frame. In that way the coefficients are seamlessly available for continuous channel reconstruction.

In order to test the functionality of the emulator we stream a complex random input stream $x[m]$ with unit variance via target-to-host DMA FIFO buffers to the FPGA. The input stream is convolved with the approximate channel impulse response $\tilde{h}[m,l]$ to obtain the output signal $y[m]$ which is streamed back via target-to-host DMA FIFO buffers and analysed on the host computer.

### A. Emulation Parameter Settings

For the emulation we consider a stationary region $T_{\text{stat}} = 0.256\,\text{ms}$ and an emulation bandwidth $B = 10\,\text{MHz}$. We implement the channel emulator oversampled with a bandwidth of 20 MHz on the FPGA to allow for a practical filter implementation to avoid aliasing. The input stream $x[m]$ and output signal $y[m]$ are filtered with a root-raised cosine filter with roll-off factor $\alpha_r = 0.2$. For a proof of concept we emulate $F = 20$ frames. The maximum velocity is set to $\text{v}_{\max} = 400\,\text{km/h}$ and the maximum delay to $\tau_{\max} = 0.4\,\mu\text{s}$. For a carrier frequency of $f_c = 5.9\,\text{GHz}$ we obtain a one sided maximum normalized Doppler $\nu_{\text{Dmax}} = 0.1092 \times 10^{-3}$ and a maximum normalized delay $\theta_{\text{Pmax}} = 0.125$. To achieve a reconstruction bias $E_{\text{th}} \leq 10^{-6}$ we set the resolution factor for the DPS sequences in the time direction $r_{\text{OT}} = 8$ and the resolution factor in the frequency direction $r_{\text{OF}} = 8192$ (see [12] for more details) and we set $D_t = 6$ and $D_f = 14$ corresponding to (9) and (10), respectively. The sequence length in time is $M = 5120$ and number of frequency bins is set to $N = 64$. The number of paths is set, but not limited, to $P = 120$ and we assume for simplicity that each path has the same power, i.e., $\mathbb{E}\left\{||\eta_p||^2\right\} = 1/P$. Finally, the delay and the Doppler shifts of the propagation paths are drawn randomly from the band-limited region $\mathcal{W}$ (cf. Section III-B) for each frame to account for real valued path delays and Doppler shift.

### B. Error Comparison

To verify the functionality of the emulator the output of the emulator $y[m]$ is co-simulated in Matlab with double precision accuracy. In the co-simulation we calculate the accuracy of the convolution for the different approximation steps. We define the squared errors by

$$\hat{e}^y[m] = 10 \cdot \log_{10}||y[m] - \hat{y}[m]||^2 \tag{18}$$

$$\tilde{e}^y[m] = 10 \cdot \log_{10}||y[m] - \tilde{y}[m]||^2 \tag{19}$$

$$\tilde{e}^{y-\text{FP}}[m] = 10 \cdot \log_{10}||y[m] - \tilde{y}^{\text{FP}}[m]||^2. \tag{20}$$

where $\hat{e}^y[m]$, $\tilde{e}^y[m]$ and $\tilde{e}^{y-\text{FP}}[m]$ denote the error of $y[m]$ with respect to the output signal obtained by convolution of $x[m]$ with the time-variant channel impulse responses obtained by the BEM $\hat{h}[m,l]$, the approximate projection BEM $\tilde{h}[m,l]$ and the fixed-point approximate projection BEM $\tilde{h}^{\text{FP}}[m,l]$, respectively. The time-variant channel impulse response $h[m,l]$ is obtained by applying an IFFT to (4), $\hat{h}[m,l]$ is obtained by applying the IFFT to $\hat{g}[m,l]$ and $\tilde{h}^{\text{FP}}[m,l]$ is the co-simulated channel impulse response on the FPGA with a FP precision of 15 bit.

The plot in Fig. 2 shows the probability density functions (pdfs) of the errors defined for all frames. We observe that the targeted error threshold of $E_{\text{th}} < 10^{-6}$ can be achieved which can be seen by the means of the pdfs. Each approximation step introduces a certain amount of error that depends on the chosen approximation parameters. The error, e.g., introduced by the BEM with exact projection can be reduced by utilizing more basis coefficients, i.e., increasing $D_t$ and $D_f$. The error introduced by the approximate projection can be reduced by

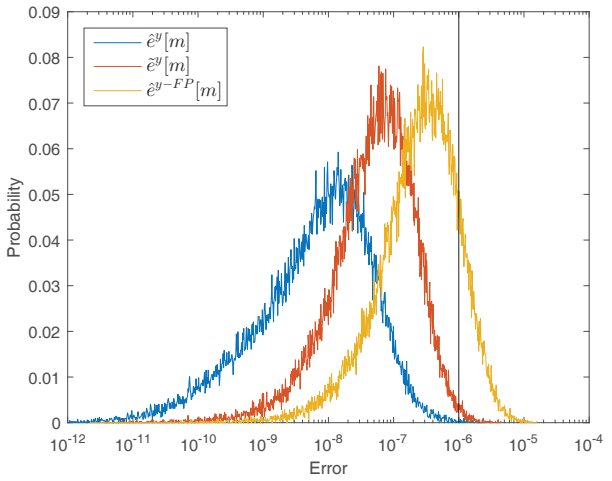


Fig. 2. Error of the output signal with respect to different levels of approximation of the channel impulse response (Matlab co-simulation)

increasing the resolution factors. Finally, increasing the bit resolution on the FPGA reduces the FP precision error. With our configuration we reach a mean squared error smaller than $10^{-6}$ for the channel emulator. This suits well for testing and evaluating communication systems.

Finally, Fig. 3 shows the error pdf of the signal convolved on the FPGA compared with the Matlab co-simulation. We ob-

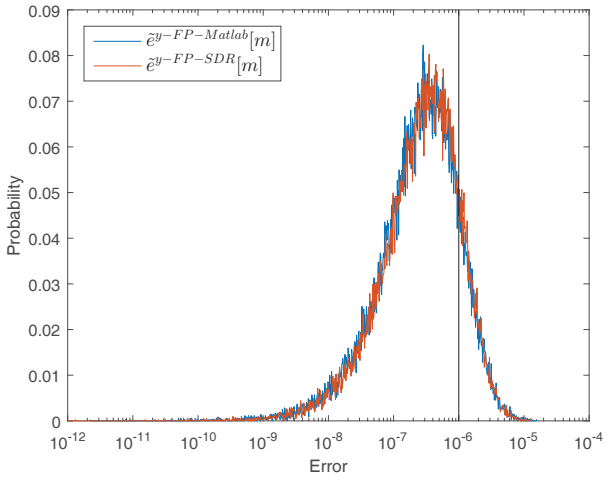


Fig. 3. Error comparison between FPGA implementation and Matlab co-simulation

serve that the FPGA implementation matches almost perfectly with the Matlab co-simulation.

## V. Conclusion

In this paper we showed the application of a low-complexity reduced rank subspace model to emulate time-variant channel impulse responses with real valued path delays and Doppler shifts on a SDR. This is a considerable benefit compared to other emulators on the market that are still limited to path delays that are integer multiples of the sampling rate. With our implementation we achieve an accuracy of $10^{-6}$. The computational complexity on the FPGA is *independent* from the number of propagation paths which makes the emulator very suitable for emulating a GSCM with a large number of propagation paths. Furthermore, the low cost of the SDR makes the emulator very attractive from an economic point of view.

## Acknowledgement

This work was partially funded by the Austrian research funding association (FFG) under the scope of an industrial PhD.

## References

[1] "ETSI TR 102 638 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions."
[2] "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*, July 2010.
[3] Accessed: 2017-01-15. [Online]. Available: http://www.spirent.com/
[4] Accessed: 2017-01-15. [Online]. Available: http://www.anite.com/
[5] (2013, May) Real-Time MIMO Channel Emulation on the NI PXIe-5644R. National Instruments. Accessed: 2017-01-15. [Online]. Available: http://www.ni.com/example/31556/en/
[6] D. Vlastaras, S. Malkowsky, and F. Tufvesson, "Stress Test of Vehicular Communication Transceivers Using Software Defined Radio," in *IEEE Veh. Technol. Conf. (VTC Spring)*, vol. 41, no. 2, May 2015, pp. 1–4.
[7] 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 *Proc. 10th Eur. Conf. Antennas Propag. (EuCAP)*, April 2016.
[8] "Mkg Sys," Accessed: 2017-01-15. [Online]. Available: https://www.mkgsys.com/
[9] T. Fernández-Caramés, M. González-López, and L. Castedo, "FPGA-Based Vehicular Channel Emulator for Real-Time Performance Evaluation of IEEE 802.11p Transceivers," *EURASIP J. Wirel. Commun. Netw.*, vol. 2010, no. 1, p. 607467, 2010.
[10] J. Matai, P. Meng, L. Wu, B. Weals, and R. Kastner, "Designing a Hardware in the Loop Wireless Digital Channel Emulator for Software Defined Radio," in *2012 International Conference on Field-Programmable Technology*, Dec. 2012, pp. 206–214.
[11] L. Bernadó, T. Zemen, F. Tufvesson, A. F. Molisch, and C. F. Mecklenbrauker, "Delay and Doppler Spreads of Nonstationary Vehicular Channels for Safety-Relevant Scenarios," *IEEE Trans. Veh. Techol.*, vol. 63, no. 1, pp. 82–93, Jan. 2014.
[12] F. Kaltenberger, T. Zemen, and C. W. Ueberhuber, "Low-Complexity Geometry-Based MIMO Channel Simulation," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, p. 095281, 2007.
[13] F. Kaltenberger, "Low Complexity Simulation of Wireless Channels using Discrete Prolate Spheroidal Sequences," Ph.D. dissertation, Technical University Vienna, May 2007.
[14] F. Kaltenberger, G. Steinböck, G. Humer, and T. Zemen, "Low-Complexity Geometry Based MIMO Channel Emulation," *European Space Agency, (Special Publication) ESA SP*, vol. 626 SP, no. October, 2006.
[15] *Device Specifications, NI USRP-2593R, 1.2 GHz to 6 GHz Tunable RF Transceiver*, National Instruments.
[16] J. Karedal, F. Tufvesson, N. Czink, A. Paier, C. Dumard, T. Zemen, C. F. Mecklenbräuker, and A. F. Molisch, "A Geometry-Based Stochastic MIMO Model for Vehicle-to-Vehicle Communications," *IEEE Trans. Wirel. Commun.*, vol. 8, no. 7, pp. 3646–3657, Jul. 2009.
[17] T. Zemen, L. Bernadó, N. Czink, and A. Molisch, "Iterative Time-Variant Channel Estimation for 802.11p Using Generalized Discrete Prolate Spheroidal Sequences," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 1222–1233, March 2012.
[18] N. Czink, F. Kaltenberger, Y. Zhou, L. Bernado, T. Zemen, and X. Yin, "Low-Complexity Geometry-Based Modeling of Diffuse Scattering," *in Proc. 4th Eur. Conf. Antennas Propag. (EuCAP)*, no. i, pp. 1–4, 2010.
[19] D. Slepian, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty-V: The Discrete Case," *Bell Syst. Tech. J.*, vol. 57, no. 5, pp. 1371–1430, May 1978.
[20] F. Kaltenberger, T. Zemen, and C. Ueberhuber, "Low-Complexity Doubly Selective Channel Simulation Using Multidimensional Discrete Prolate Spheroidal Sequences," *15th IST Mobile and Wireless Communication Summit (IST SUMMIT)*, 2006.