LOW-COMPLEXITY GEOMETRY BASED MIMO CHANNEL EMULATION

Florian Kaltenberger[1], Gerhard Steinböck[1], Gerhard Humer[1], and Thomas Zemen[2]

[1]*ARC Seibersdorf research GmbH, Donau-City-Strasse 1, 1220 Vienna, Austria, Email:*
florian.kaltenberger@arcs.ac.at,

[2]*Forschungszentrum Telekommunikation Wien, Donau-City-Strasse 1, 1220 Vienna, Austria, Email:*
thomas.zemen@ftw.at,

ABSTRACT

The design and optimization of modern radio communication systems requires realistic models of the radio propagation channel. Especially for the test of mobile radio hardware devices, real-time implementations of such channel models are needed. Geometry based channel models are computationally intensive, since for every propagation path, every time instance and every delay or frequency bin a complex exponential has to be evaluated. On a real-time hardware channel emulator, like the ARC SmartSim, the number of paths P, that can be simulated, is limited by the available processing power. In this paper, a new method is introduced, that allows both to increase the number of paths P and to reduce the overall computational complexity.

1. INTRODUCTION

The COST 259 geometry based stochastic channel model (GSCM) [1, 2] assumes that the electromagnetic waves are scattered at objects randomly placed in the simulation environment. It uses ray tracing techniques to calculate the attenuation, delay, Doppler shift, direction of departure, direction of arrival, and polarization of every multipath component (MPC) of the channel. Scattering objects with similar properties are combined in clusters.

This very general channel model can be used to simulate a wide range of scenarios. The COST 259 GSCM distinguishes macrocells (outdoor urban), microcells (outdoor city) and picocells (indoor). Each of them uses different parameters for the placement of the clusters and the scatterers. While for macrocells a small number of clusters (1-2) is sufficient to model the channel accurately (compared with measurements), picocells require a mean number of 16 clusters [1]. To achieve a Rayleigh fading statistic, approximately 20-30 paths per cluster are necessary [3]. This results in approximately 400 paths in total. However, the simulation of such scenarios has a very high computational complexity, since for every propagation path, every time instance and every delay or frequency bin a complex exponential has to be evaluated. On a real-time hardware channel emulator, like the *ARC SmartSim* [4], the number of paths P, that can be simulated, is limited by the available processing power.

In [5, 6] we have shown that the overall computational complexity can be reduced by several orders of magnitude by projecting the time-variant transfer function of the channel in the subspace spanned by *multidimensional discrete prolate spheroidal (DPS) sequences*. By adjusting the dimension of the subspace, it is possible to trade complexity for accuracy. Furthermore, the subspace representation is independent of the number of paths and the projection can be calculated efficiently in $\mathcal{O}(1)$ operations directly from the path parameters given by the model.

In this paper we show how the subspace based method can be incorporated in a GSCM and implemented on the ARC SmartSim channel emulator. Therefore we derive a subspace representation of the time-variant impulse response of the channel. We show, that this implementation is equivalent to the implementation described in [4]. The complexity of the proposed method is however smaller for a wide range of scenarios.

The paper is organized as follows. The notation is introduced in Section 2. In Section 3, the ARC SmartSim channel emulator and its signal model is described. Section 4 describes the GSCM in more detail. In Section 5, the subspace representation of time-variant, frequency selective channels is derived. The implementation of the subspace methods on the ARC SmartSim is treated in Section 6. Finally, a complexity comparison and the memory requirements of the methods are given in Section 7. Conclusions are drawn in Section 8.

Figure 1. The baseband processing unit of the ARC SmartSim channel emulator.

2. NOTATION

Vectors are denoted by $\boldsymbol{v}$ and matrices by $\mathbf{V}$. Their elements are denoted by v_i and $\mathrm{V}_{i,l}$. The transpose of a vector or a matrix is given by $\cdot^{\mathrm{T}}$ and its conjugate transpose by $\cdot^{\mathrm{H}}$. The norm of vector $\boldsymbol{a}$ is denoted through $\|\boldsymbol{a}\|$. The Kronecker product is denoted by $\otimes$. $|X|$ denotes the number of elements of X if X is a discrete index set or the area of X if X is a continuous region. The identity matrix of dimension M is given by $\mathbf{I}_M$ and the Fourier transform matrix $\mathbf{F}_Q$ of length Q is given by $(\mathrm{F}_Q)_{n,q} = e^{2\pi jnq/Q}$.

3. THE ARC SMARTSIM CHANNEL EMULATOR

The hardware of the ARC SmartSim channel emulator is a modular architecture comprised of a baseband signal processing unit, an analogue frontend and a radio frequency (RF) frontend. The baseband signal processing unit is a parallel architecture of DSP boards, which are described in Subsection 3.1. The analogue and RF boards are described in Subsection 3.2.

3.1. The Baseband Processing Unit

The DSP boards basically consist of a TI C6416 DSP and a Xilinx Virtex 2 FPGA. Figure 2 depicts a block diagram of the board. The DSP has a fixed point architecture with 16 bit precision. The FPGA provides digital baseband interfaces for input and output signals, as well as for communication over the internal backplane. The baseband signals are implemented by means of LVDS (Low Voltage Differential Signalling) and using standard "Channel Link" connectors. A maximum sample rate of 40MSPS is supported and the dynamic range is 16 bit for inputs and 32 bit for outputs. A Motorola Coldfire microprocessor is used for configuration of the board and communication with a PC. A more detailed description of the board is given in [7, 8].

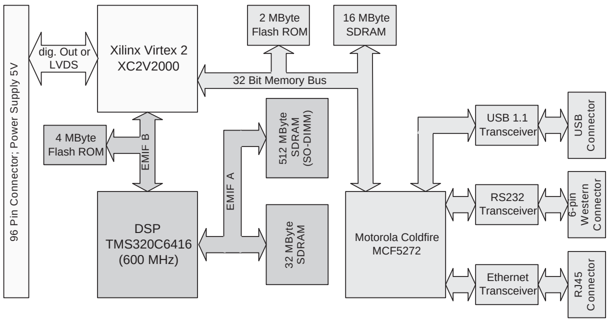

Figure 2. Block diagram of the DSP-board

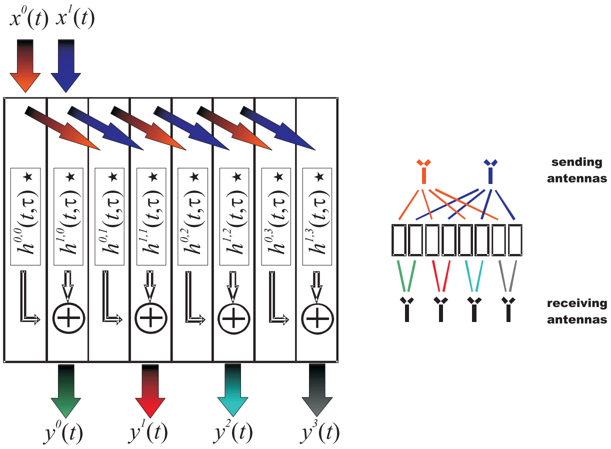

Figure 3. MIMO 2 → 4 scenario and the corresponding configuration of development platform.

The MIMO channel emulation method is depicted in Figure 3 for a MIMO channel with $N_T = 2$ transmit antennas and $N_R = 4$ receive antennas. The input signals $x^j(t)$, $j = 0, \ldots, N_T - 1$, are fed into the first N_T boards and then distributed to the other boards. Each DSP board computes the channel's impulse response $h^{i,j}(t,\tau)$ based on the COST259 GSCM for one pair of transmit and receive antennas (i, j) and its convolution with the input signal

$$y^{i,j}(t) = \int h^{i,j}(t,\tau)x^j(t-\tau)\mathrm{d}\tau. \qquad (1)$$

Thus, in a $N_T \times N_R$ MIMO scenario, $N_T N_R$ DSP-boards are needed. After the convolution, the results of N_T adjacent boards are summed up, resulting in

$$y^i(t) = \sum_{j=0}^{N_T-1} y^{i,j}(t). \qquad (2)$$

3.2. Analogue and RF Units

Each DSP-board can optionally be equipped with an analogue frontend providing in- and outputs at an

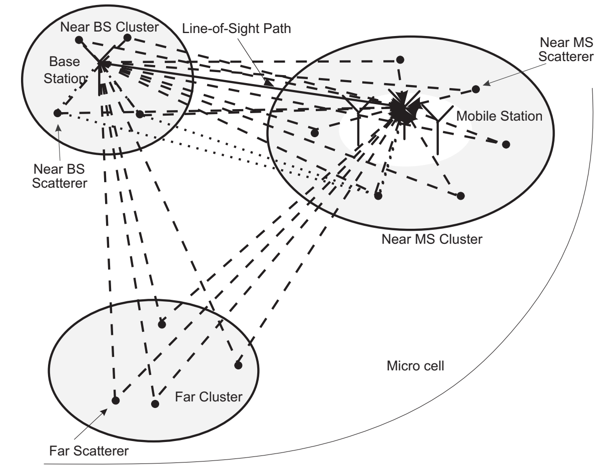

Figure 4. Geometrical model for signal propagation

intermediate frequency (IF) of 140MHz with a maximum bandwidth of 40MHz and a dynamic range of 14 bit input and 16 bit output respectively. The board comprises a digitizer, which is built by two TI ADS5500, a D/A converter plus a Xilinx XC2VP30 FPGA, which handles I/O interfacing, digital mixing, and controlling of the board.

The RF frontends are separated from the rest of the system and reside in an own rack identical to the rack of the platform to reduce influences on the system. Both the transmitter and the receiver are implemented on a separate board. The current version of the RF receiver is designed for WiMAX signals with a center frequency of 2.58GHz and a bandwidth of 20MHz. Extensions complying WLAN and UMTS standards are planned for the next version.

4. THE GEOMETRY BASED STOCHASTIC CHANNEL MODEL

The principle of the geometry based stochastic channel model (GSCM) is shown in Figure 4 [1,2]. A base station (BS) and a mobile station (MS), each having several antennas, are placed within the simulation area and a velocity vector is assigned to the MS.

The model assumes that scatterers are arranged in clusters. For each cluster an angular delay power spectra (ADPS) is defined. There is a cluster of scatterers near the BS, another one around the MS and several far clusters model the large scale environment. The positions of the scatterers in the clusters are modeled stochastically.

Assuming specular reflection at the scatterers, raytracing is used to compute the attenuation, delay, Doppler shift, direction of departure, direction of arrival of every MPC of the channel. Transmitted signals can be scattered once (dashed lines) or twice (dotted lines); cf. Figure 4. The solid line denotes the line of sight path.

For every MPC p the attenuation $\eta_p(t)$, the initial phase ϕ_p, the Doppler shift ω_p, and the delay $\tau_p(t)$ is calculated. $\eta_p(t)$ incorporates the path loss given by the path loss model and the channel variations due to the large scale fading. The phase ϕ_p incorporates the angles of arrival and departure respectively. The impulse response between transmit antenna i and receive antenna j is then given by

$$h'^{i,j}(t,\tau) = \sum_{p \in P_{i,j}} \eta_p(t) e^{2\pi j(\omega_p t + \phi_p)} \delta(\tau - \tau_p(t)), \quad (3)$$

where $P_{i,j}$ is the set of paths from transmit antenna i to receive antenna j and δ is the dirac impulse.

5. SUBSPACE REPRESENTATION OF TIME-VARIANT FREQUENCY-SELECTIVE CHANNELS

In this section we describe a new method for the simulation of time-variant frequency selective channels using a subspace representation. The subspace representation is independent of the number of propagation paths and thus able to reduce the complexity of geometry based channel models [5, 6].

5.1. Sampled Signal Model

The subspace representation of time-variant frequency selective channels is best viewed in the frequency domain. Thus, we define the time-variant transfer function $L'^{i,j}(t,f)$ of the channel by Fourier transforming the time-variant impulse response (3)

$$\begin{aligned} L'^{i,j}(t,f) &= \int h'^{(i,j)}(t,\tau) e^{-2\pi j f \tau} \mathrm{d}\tau & (4) \\ &= \sum_{p \in P_{i,j}} \eta_p(t) e^{-2\pi j(\omega_p t + \phi_p)} e^{-2\pi j \tau_p(t) f}. & (5) \end{aligned}$$

The channel can only be measured respectively simulated for a finite number of time samples in the delay domain or—equivalently—a finite number of frequency samples in the frequency domain. This corresponds to applying a rectangular filter to the channel. Let $r(\tau)$ denote the filter in the delay-domain and $R(f)$ the filter in the frequency domain and

$$h^{(i,j)}(t,\tau) = h'^{(i,j)}(t,\tau) * r(\tau) \quad (6)$$

$$L^{(i,j)}(t,f) = L'^{(i,j)}(t,f) \cdot R(f), \quad (7)$$

the filtered impulse response and transfer function respectively ($*$ denotes convolution).

To obtain the sampled time-variant transfer function, t and τ are sampled at rate $1/T_S$ and f is sampled at rate $Q/F_S = T_S Q$, where Q is the number of frequency bins within the bandwidth B. Q has to be chosen such that $Q > N$, where $N = N_C + 2N_F + 1$ and $N_C = \tau_{\max}/T_S$ is the support of the impulse response and $2N_F + 1$ is the length of the filter $r(\tau)$. Further, it is assumed that the attenuation $\eta_p(t) = \eta_p$ and $\tau_p(t) = \tau_p$ are constant for the duration T. Then the sampled time-variant impulse response and transfer function write

$$\begin{aligned} h^{i,j}_{m,n} &= h^{i,j}(mT_S, nT_S) \\ &= \sum_{p \in P_{i,j}} \eta_p e^{j2\pi(\nu_p m + \phi_p)} r(nT_S - \tau_p) \qquad (8) \\ g^{i,j}_{m,q} &= L^{i,j}(mT_S, q/(T_S Q)) \\ &= \sum_{p \in P_{i,j}} \eta_p e^{j2\pi(\nu_p m + \phi_p)} e^{-j2\pi\theta_p q/Q}, \qquad (9) \end{aligned}$$

where $\nu_p = \omega_p T_S$ is the normalized Doppler frequency and $\theta_p = \tau_p/T_S$ is the normalized delay.

Denote by $\boldsymbol{g}$ the vector of elements of the time-variant transfer function $g^{i,j}_{m,q}$ in the finite time interval $m \in \{0, \ldots, M-1\}$, where $M = T/T_S$, and in the finite frequency bandwidth $q \in \{-\lfloor Q/2 \rfloor, \ldots, \lceil Q/2 \rceil - 1\}$, indexed lexicographically,

$$\boldsymbol{g} = [g_{0,-\lfloor Q/2 \rfloor}, g_{0,-\lfloor Q/2 \rfloor + 1}, \ldots, g_{M-1,\lceil Q/2 \rceil - 1}]^T. \qquad (10)$$

Further, define $\boldsymbol{h}$ as the vector of elements of the time-variant impulse response $h_{m,n}$ similarly to (10). $\boldsymbol{h}$ can also be calculated by a DFT in the frequency domain from (9) by

$$\boldsymbol{h} = (\mathbf{I}_M \otimes \mathbf{F}_Q)\boldsymbol{g}. \qquad (11)$$

5.2. Subspace Representation

The subspace representation is based on the following two observations. Firstly, the simulation is carried out for a limited time T and a limited bandwidth B. Secondly, the variation of the channel in time and frequency is limited by the maximum Doppler shift $\omega_{\max}$ and the maximum delay $\tau_{\max}$, i. e., $|\omega_p| < \omega_{\max}$ and $0 \leq \tau_p < \tau_{\max}$ for $p \in P$.

Therefore, $\boldsymbol{g}$ can be represented in the subspace spanned by the first D indexlimited two-dimensional discrete prolate spheroidal (DPS) sequences $v^{(d)}_{m,q}(W, I)$, $d = 0, \ldots, D-1$ [6], where

$$\begin{aligned} W &= [-\nu_{\mathrm{Dmax}}, \nu_{\mathrm{Dmax}}] \times [0, \theta_{\max}], \quad \text{and} \qquad (12) \\ I &= \{0, \ldots, M-1\} \times \{-\lfloor Q/2 \rfloor, \ldots, \lceil Q/2 \rceil - 1\} \qquad (13) \end{aligned}$$

denote the bandlimit region and the region of energy-concentration respectively. If the indexlimited DPS sequences $v^{(d)}_{m,q}(W, I)$ are collected in a vector $\boldsymbol{v}^{(d)}$ similar to (10) and the vectors are further collected in the columns of the matrix

$$\mathbf{V} = [\boldsymbol{v}^{(0)}(W, I), \ldots, \boldsymbol{v}^{(D-1)}(W, I)], \qquad (14)$$

the subspace representation of $\boldsymbol{g}$ is

$$\boldsymbol{\alpha} = \mathbf{V}^H \boldsymbol{g}, \qquad (15)$$

$$\hat{\boldsymbol{g}} = \mathbf{V}\boldsymbol{\alpha}. \qquad (16)$$

The projection of a one-dimensional complex exponential,

$$\gamma_d(\nu_p) = \sum_{m=0}^{M-1} v^{(d)}_m(\nu_{\mathrm{Dmax}}, M) e^{2\pi j \nu_p m} \qquad (17)$$

can be calculated approximately by a simple scaling and shifting of the sequence $v^{(d)}_m(\nu_{\mathrm{Dmax}}, M)$ [5, 6]

$$\gamma_d(\nu_p) \approx \tilde{\gamma}_d(\nu_p) = c(\nu_{\mathrm{Dmax}}, M) v^{(d)}_{m(\nu_p/\nu_{Dmax})}(\nu_{\mathrm{Dmax}}, M) \qquad (18)$$

where $c(\nu_{\mathrm{Dmax}}, M)$ is the scaling factor and $m(x) = \lfloor (1+x)(M-1)/2 \rfloor$ is the shifting function. In [5, 6] is is shown, that $\tilde{\gamma}_d(\nu_p) \to \gamma_d(\nu_p)$ as $M \to \infty$. Thus, oversampling M results in better approximation.

Since W can be written as a Cartesian product (12), the basis coefficients $\boldsymbol{\alpha}_d$ can be calculated approximately as

$$\tilde{\boldsymbol{\alpha}}_d = \sum_{p \in P_{i,j}} \eta_p e^{2\pi i \phi_p} \tilde{\gamma}_{d_0}(\nu_p) \tilde{\gamma}_{d_1}(\theta_p). \qquad (19)$$

Note, that this calculation is independent of M and Q.

We define the approximate subspace representation of $\boldsymbol{g}$ as

$$\tilde{\boldsymbol{g}} = \mathbf{V}\tilde{\boldsymbol{\alpha}}. \qquad (20)$$

A subspace representation $\hat{\boldsymbol{h}}$ of the sampled time-variant impulse response can be calculated from $\hat{\boldsymbol{g}}$ by a discrete Fourier transform (DFT) in the frequency variable q.

$$\hat{\boldsymbol{h}} = (\mathbf{I}_M \otimes \mathbf{F}_Q)\hat{\boldsymbol{g}}. \qquad (21)$$

Defining the basis functions in the delay domain by $\mathbf{U} := (\mathbf{I}_M \otimes \mathbf{F}_Q)\mathbf{V}$, $\hat{\boldsymbol{h}}$ can be calculated using the same basis coefficients $\boldsymbol{\alpha}$

$$\hat{\boldsymbol{h}} = \mathbf{U}\boldsymbol{\alpha}. \qquad (22)$$

The approximate subspace representation $\tilde{\boldsymbol{h}}$ is defined similarly to (20).

5.3. Bias and Subspace Dimension

The bias of the approximate subspace representation is given by

$$\mathrm{bias}^{(D)}_{\tilde{\boldsymbol{h}}} = \mathcal{E}\left\{ \frac{1}{MQ} \|\boldsymbol{h} - \tilde{\boldsymbol{h}}^{(D)}\|^2 \right\}, \qquad (23)$$

where the superscript (D) denotes the number of basis functions used. In [5,6] it was shown that the bias decreases exponentially with the subspace dimension D until the error floor given by the approximation (18) is reached.

We wish to simulate channels with an accuracy not lower than the machine precision of 16 bit, $E_{\text{prec}} = 2^{-15}$ (cf. Section 3.1). This can be accomplished in a two-step procedure. First the oversampling factor of the approximation is determined, so that the approximation error is smaller than E_{prec}. Then, the minimum subspace dimension is calculated using

$$D_{\min} = \underset{D}{\operatorname{argmin}}\{\text{bias}_{\tilde{\boldsymbol{h}}}^{(D)} \le E_{\text{prec}}\}. \qquad (24)$$

6. REAL-TIME IMPLEMENTATION ASPECTS

In this Section, we describe how the GSCM is implemented on the hardware described in Section 3 and how the DPS subspace representation can be used to increase the number of paths.

At the beginning of each simulation run, the global parameters of the scenario are initialized and distributed to the DSP boards, such that each DSP board is able to compute the radio channel for one pair of transmit and receive antennas.

The real-time channel emulation is then divided into two main parts, namely the propagation module, and the convolution module. The propagation module is implemented in the DSP and calculates the parameters of all MPCs of the channel. The convolution module is implemented in the FPGA. It takes the input samples and convolves them with the calculated impulse response.

6.1. Propagation Module

The computation of parameters of the MPCs is divided in a large scale and a small scale update to fulfill the real-time constraints.

The large scale update updates the positions of the scatterers according to the stochastic properties of the channel model and thus simulates long term fading. Since these values are varying significantly only after a MS movement of approx. ten wave lengths they are refreshed every 10 ms (corresponding to a carrier frequency of 2GHz and a maximum speed of 500km/h). In the large scale update, the different attenuation values like the shadow fading, the attenuation of the scatterers and the rice factor are calculated and are linearly interpolated in between. The path lengths of the MPCs are calculated twice in this 10 ms update interval and are piecewise quadratically interpolated in between. For a detailed view on the calculations of this parameters see [1, 2, 9].

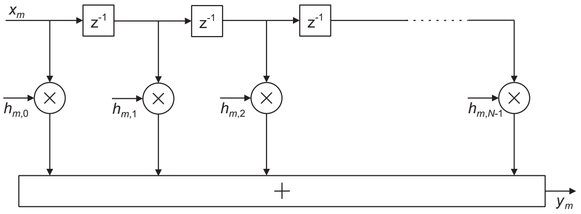

Figure 5. Conventional FIR Filter structure

In the small scale update, the attenuation η_p, the initial phase ϕ_p, the Doppler shift ω_p and the delay τ_p of all MPCs $p \in P_{i,j}$ are calculated. The small scale update occurs every $T = 0.5$ms, since those values are varying very fast even when the MS is moving only a fraction of a wavelength.

The parameters η_p, ϕ_p, ω_p and τ_p are now used to calculate the approximate basis coefficients $\tilde{\boldsymbol{\alpha}}$ according to Equation (19).

6.2. Convolutional Module

The convolutional module receives the subspace coefficients $\tilde{\boldsymbol{\alpha}}$ from the DSP and calculates the time-variant impulse response $\tilde{\boldsymbol{h}}$ according to Equation (22).

Then, the sampled input signal $x_m = x(mT_S)$ is convolved with the impulse response $\tilde{h}_{m,n}$

$$y_m = \sum_{n=0}^{N-1} \tilde{h}_{m,n} x_{m-n}. \qquad (25)$$

Equation (25) can be implemented using a tapped delay line (TDL) depicted in Fig. 5. This implementation is different to the one used in [4], where the input signal is highly oversampled with oversampling factor O_T and written to a "History RAM" (see Fig. 6). From there, for each path, the correctly delayed input samples are multiplied with their complex attenuation $\beta_{m,p} = \eta_p e^{2\pi j(m\nu_p + \phi_p)}$ and added up to yield one output sample. In the Appendix it is shown, that the proposed implementation is equivalent to the History RAM implementation.

7. COMPLEXITY COMPARISON AND MEMORY REQUIREMENTS

In this section, the complexities and the memory requirements of three different implementations for the

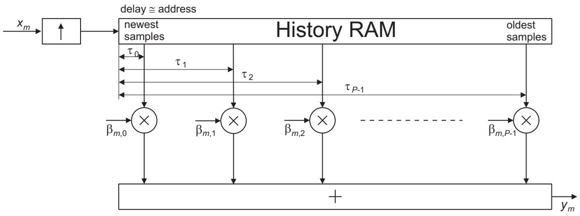

Figure 6. Convolution with the History RAM

channel simulation are reported. The first two implementations use a tapped delay line, where the tap weights are either generated using DPS sequences (22) or where the tap weights are calculated using (8). They are compared to the implementation using the History RAM [4]. Further, the memory requirements of the new method are given.

7.1. Complexity Comparison

The complexity is measured using number of complex multiplications (CM) and number of evaluations of a complex exponential (CE). We count the number of operations for a single time step m of length T_S.

The complexity of the History RAM (HR) implementation is

$$C_{HR} = (O_T(2N_F + 1))\mathrm{CM} + P(\mathrm{CM} + \mathrm{CE}), \quad (26)$$

where the first term accounts for the oversampling filter with oversampling factor O_T (29) and the second term accounts for the number of paths (30).

The complexity of the TDL implementation, where the tap weights are calculated using (8) is

$$C_{TDL} = (2N_F + N_C + 1)\mathrm{CM} + P(\mathrm{CM} + \mathrm{CE}), \quad (27)$$

where the first term accounts for the implementation of the tapped delay line (25) and the second term accounts for calculating the time-variant impulse response (8).

The complexity of the TDL implementation, where the tap weights are calculated using DPS sequences (22) is

$$C_{DPSS} = (2N_F + N_C + 1)\mathrm{CM} + D\mathrm{CM}, \qquad (28)$$

where the first term accounts for the implementation of the tapped delay line (25) and and the second term accounts for calculating the time-variant impulse response (22).

For the complexity comparison, the parameters of Table 1 are used. The maximum normalized Doppler shift is fixed to $\nu_{\mathrm{Dmax}} = 1.2044 \cdot 10^{-5}$, which corresponds to a user velocity of 100kmph. In Figure 7, the minimum complexity regions of the different implementations are plotted in dependence of the delay of the channel in samples N_C and the number of paths P. The subspace dimension D has been calculated using (24).

Parameter	Value
Carrier frequency f_c	$2 \cdot 10^9$ Hz
Bandwidth B	$3.84 \cdot 10^6$ Hz
Sample rate $1/T_\mathrm{S}$	$15.36 \cdot 10^6$ Hz
Update interval T	$0.5 \cdot 10^{-3}$ sec
Blocklength $M = T/T_S$	7680 samples
Delay bins Q	256 bins
Filter group delay N_F	5
OSF History RAM O_T	8

Table 1. Simulation parameters for the numerical experiments.

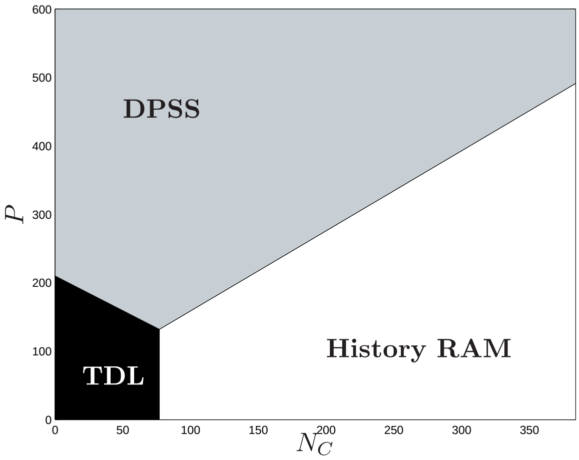

Figure 7. Minimum complexity regions of the implementation of the convolution in dependence of the number of paths P and maximum delay of the channel N_C (in samples).

It can be seen that the tapped delay line implementation performs best for small delay spreads and small number of paths. If the delay spread is large, but the number of paths is small, i. e., the impulse response of the channel is sparse, then the implementation using the History RAM is favorable. In all other cases, i. e., when the number of paths is large, then the implementation using the DPS sequences should be preferred.

7.2. Memory requirements

The special choice of the bandlimit region (12) and the time-concentration regions (13) as Cartesian products enables efficient storage of the two-dimensional DPS sequences, because they can be calculated as a Kronecker product of one-dimensional

	$\nu_{\mathrm{Dmax}}(\cdot 10^{-4})$				
N_C	0.1204	0.2409	0.3613	0.4818	0.6022
64	95	99	103	107	110
128	159	163	167	171	174
192	223	227	231	235	238
256	287	291	295	299	302
320	351	355	359	363	366
384	415	419	423	427	430

Table 2. Memory requirements for the two-dimensional DPS sequences in kByte in dependence of the maximum normalized Doppler shift ν_{Dmax} and maximum number of delay taps N_C. Every complex sample is allocated 32 bits.

sequences. Therefore, to store $\mathbf{U}$, only $MD_0 + QD_1$ elements need to be stored, where D_0 and D_1 are the maximum number of one-dimensional DPS vectors needed to construct the two-dimensional vectors. D_0 and D_1 can be calculated using the one-dimensional equivalent of Equation (24). The results are summarized in Table 2 using the parameters of Table 1. For comparison, the History RAM implementation of [4] needs only 8kByte of memory.

8. CONCLUSIONS

We have presented a low-complexity implementation of the COST 259 geometry based stochastic channel model (GSCM) that is especially suited for real-time channel emulation. The method allows to simulate scenarios with a very high number of propagation paths. Compared to other implementations the newly proposed method has less complexity for most scenarios. On the other hand, the memory requirements are higher.

ACKNOWLEDGEMENTS

Parts of this work were funded by the Wiener Wissenschafts-, Forschungs- und Technologiefonds (WWTF) in the ftw. project Future Mobile Communication Systems (Math+MIMO).

REFERENCES

1. L. M. Correia, editor. *Wireless Flexible Personalised Communications.* Wiley, 2001.

2. H. Hofstetter and G. Steinböck. A geometric based stochastic channel model for MIMO systems. In *Proc. ITG Workshop on Smart Antennas*, Munich, Germany, March 18th-19th 2004.

3. Y. R. Zheng and C. Xiao. Simulation models with correct statistical properties for Rayleigh fading channels. *IEEE Trans. Commun.*, 51(6):920–928, June 2003.

4. F. Kaltenberger, G. Steinböck, R. Kloibhofer, R. Lieger, and G. Humer. A multi-band development platform for rapid prototyping of MIMO systems. In *Proc. ITG Workshop on Smart Antennas*, Duisburg, Germany, April 2005.

5. F. Kaltenberger, T. Zemen, and C. W. Ueberhuber. Low complexity simulation of wireless channels using discrete prolate spheroidal sequences. In *Proc. MATHMOD Conference*, Vienna, Austria, February 2006.

6. F. Kaltenberger, T. Zemen, and C. W. Ueberhuber. Low complexity doubly selective channel simulation using multidimensional discrete prolate spheroidal sequences. In *Proc. IST Mobile and Wireless Communications Summit*, Myconos, Greece, June 2006.

7. G. Meindl-Pfeiffer, R. Kloibhofer, F. Kaltenberger, and G. Humer. Multistandard development platform for MIMO software defined radio. In *Proc. EUSIPCO*, Antalya, Turkey, Sept. 2005.

8. G. Meindl-Pfeiffer, R. Kloibhofer, F. Kaltenberger, and G.Humer. Development platform for MIMO software defined radio. In *Proc. Embedded World Conference*, pages 243–251, Nuernberg, Germany, Feb. 2005.

9. G. Humer, R. Kloibhofer, G. Pfeiffer, and G. Steinböck. COST 259 channel model implementation for real-time simulation. Technical Report TD(03)202, COST 273, Prague, Chech Republic, Sept. 2003.

APPENDIX: EQUIVALENCE OF HISTORY RAM AND TAPPED DELAY LINE

In the History RAM implementation [4], the input signal $x_m = x(mT_S)$ is first oversampled to a rate $1/T_D$, with an integer oversampling factor $O_T = T_S/T_D$ using an interpolation filter with coefficients $r_l = r(lT_D)$. The filter length is $L = 2N_FO_T + 1$. Thus, the oversampled signal writes

$$\bar{x}_k = x(kT_D) = \sum_{l=-N_FO_T}^{N_FO_T} r_l \bar{\bar{x}}_{k-l}, \tag{29}$$

where

$$\bar{\bar{x}}_k = \begin{cases} x_m & k = mO_T \\ 0 & \text{otherwise} \end{cases}$$

Further, also the time-variant impulse response (3) is oversampled with rate $1/T_D$,

$$\bar{h}'_{k,l} = \sum_{p=0}^{P-1} \gamma_p(kT_D)\delta_{l-\bar{\theta}_p},$$

where

$$\begin{aligned}\gamma_p(t) &= \eta_p e^{2\pi j(\phi_p+\omega_p t)} \\ \bar{\theta}_p &= \lfloor \tau_p/T_D \rfloor,\end{aligned}$$

The oversampled input signal $\bar{x}_k$ is then convolved with the oversampled impulse response $\bar{h}_{k,l}$ to yield

$$\begin{aligned}\bar{y}_k &= \sum_{l=0}^{N_C O_T-1} \bar{h}'_{k,l}\bar{x}_{k-l} \qquad (30) \\ &= \sum_{p=0}^{P-1} \gamma_p(kT_D)\tilde{x}_{k-\bar{\theta}_p},\end{aligned}$$

which is exactly the History RAM implementation in Figure 6.

On the other hand, using (29), (30) can be written as

$$\begin{aligned}\bar{y}_k &= \sum_{l=0}^{N_C O_T-1} \bar{h}'_{k,l}\left\{\sum_{l'=-N_F O_T}^{N_F O_T} r_{l'}\bar{\bar{x}}_{k-l-l'}\right\} \\ &= \sum_{l'=-N_F O_T}^{(N_F+N_C)O_T} \left\{\sum_{l=0}^{N_C O_T-1} \bar{h}'_{k,l} r_{l-l'}\right\}\bar{\bar{x}}_{k-l'}.\end{aligned}$$

Since $\bar{\bar{x}}_k = 0$ if $k = 0 \mod O_T$, the downsampled version of $\bar{y}_k$ is

$$\begin{aligned}y_m &= \bar{y}_{mO_T} \\ &= \sum_{n'=-N_F}^{N_F+N_C}\left\{\sum_{n=0}^{N_C-1} \bar{h}'_{mO_T,nO_T} r_{(n-n')O_T}\right\} x_{m-n'} \\ &= \sum_{n'=-N_F}^{N_F+N_C} h_{m,n'}x_{m-n'},\end{aligned}$$

where

$$h_{m,n} = \sum_{p=0}^{P-1} \gamma_p(mT_S)r(nT_S-\tau_p),$$

which is the same as (8).