A Versatile Propagation Channel Simulator for MIMO Link Level Simulation

Jean-Marc Conrat, Patrice Pajusco

France Telecom NSM/R&D/RESA/NET

6, av. des Usines, BP 382, 90007 Belfort Cedex

jeanmarc.conrat@orange-ftgroup.com, patrice.pajusco@orange-ftgroup.com

Abstract—**This paper presents a propagation channel simulator for polarized bidirectional wideband propagation channels. The generic channel model implemented in the simulator is a set of rays described by geometrical and propagation features such as the delay, 3-D direction at the base station and mobile station and the polarization matrix. Thus, most of the wideband channel models including tapped delay line models, tap directional models, scatterer or geometrical models, ray-tracing or ray-launching results can be simulated. The simulator is composed of two major parts: firstly the channel complex impulse responses (CIR) generation and secondly the channel filtering. CIRs (or CIR matrices for MIMO configurations) are processed by specifying a propagation model, an antenna array configuration, a mobile direction and a spatial sampling factor. For each sensor, independent arbitrary 3-D vectorial antenna patterns can be defined. The channel filtering is based on the overlap-and-add method. The time-efficiency and parameterization of this method is discussed with realistic simulation setups. The global processing time for the CIR generation and the channel filtering is also evaluated for realistic configuration. A simulation example based on a bidirectional wideband channel model in urban environments illustrates the usefulness of the simulator.**

Index Terms—**Propagation channel, MIMO, simulation, rays, models**

I. INTRODUCTION

Multiple antenna radio access (MIMO) based on antenna arrays at both the mobile station (MS) and the base station (BS) have recently emerged as a key technology in wireless communications for increasing the data rates and system performances [1, 2]. The benefits of multiple antenna technologies can be shown by achieving link-level simulations. The reliability of the results from link-level simulations depends strongly on a realistic modeling of the propagation channel. This is particularly true for wideband MIMO systems, when polarization and spatial diversities are foreseen at the Base Station (BS) or at the Mobile Station (MS).

There are basically two MIMO propagation channel types [3, 4]: physical and non-physical models. Non-physical models are based on the statistical description of the channel using non-physical parameters, such as the signal correlation between the different antenna elements at the receiver and transmitter [5, 6]. In contrast, physical models provide either the location and electromagnetic properties of scatterers or the physical description of rays. For instance, geometrical models [7-9], directional tap models [10-12] or ray tracing [13, 14] are examples of physical models. Both approaches

have advantages and disadvantages but physical models seem to be more suitable for MIMO applications because they are independent from the antenna array configuration [15]. Furthermore, they inherently preserve the joint properties of the propagation channel in temporal, spatial and frequential domains. By taking into account antenna diagrams, Doppler spectrum or correlation matrices can be coherently deduced from a physical model.

The implementation of physical models in a link-level simulation chain is not always straightforward for scientists involved in signal processing research. This paper presents a time-efficient and flexible MIMO propagation channel simulator which is compatible with all physical models. This propagation simulator was developed by the Research and Development Division of France Télécom R&D and is called Mascaraa. The key feature of Mascaraa is the consideration of each physical model as a set of rays. The ray-based approach used in Mascaraa is similar to the double directional radio channel concept introduced in [16]. A ray is characterized by geometrical and propagation characteristics. The geometrical characteristics of a ray are the path length or the delay in time domain, the 3-D direction at BS and MS. The propagation characteristics are the channel complex gains depending on the transmitted and received polarization. The main objective of this paper is not to describe all theoretical concepts of the physical modeling but to underline how they can be efficiently implemented in a propagation simulator.

This paper is divided into six major parts. The first four parts contain the theoretical concepts of Mascaraa: ray generation, impulse response processing, and channel filtering. Part V describes the software implementation and gives some details about the processing time performances. Finally, a simulation example is given in part VI.

II. RAY GENERATION

This section describes the properties of each ray and explains how Mascaraa processes a set of rays from four usual wideband propagation models. As the topic of this paper is to introduce a propagation simulator, the advantages and disadvantages of these different models will not be discussed here.

A. Ray characteristics

Each ray is characterized by its geometrical properties and electromagnetic properties. The geometrical properties of a

ray are the length and the azimuth/elevation at BS and MS. Usually, the elevation is defined as being the angle between axis Z and the ray direction (fig.1). The elevation is set between 0° and 180°. The azimuth is defined as being the angle between axis X and the perpendicular projection of the ray in the x-y plan. The azimuth varies in a range of 360°. We note θ and ϕ the elevation and azimuth.
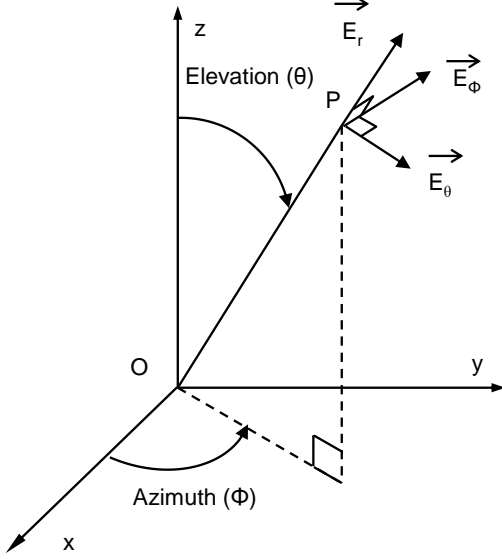

Fig. 1: Reference system for ray characterisation

The electromagnetic properties of rays allow the determination of the received field as a function of the transmitted field. By assuming the plane wave propagation hypothesis, the transmitted or received field is always perpendicular to the propagation direction. It is thus more convenient to express this field in the spherical base (E_r, E_θ, E_ϕ) than in the base (E_x, E_y, E_z) common for all directions. Vector E_r has the same direction as vector OP. Vector E_θ is perpendicular to OP and is contained in the zOP plan. Vector E_ϕ is perpendicular to OP and is included in the x-y plan. Whatever the polarization of the wave, E_r component is always null (plane wave assumption). If E_ϕ is null, the polarization is vertical. If E_θ is null, the polarization is horizontal. As the propagation channel causes a complex attenuation and a rotation of the polarization vector about the ray axis, the received field is given by the following matrix equation:

$$\begin{pmatrix} E_\theta^{Rx} \\ E_\phi^{Rx} \end{pmatrix} = A \cdot \begin{pmatrix} E_\theta^{Tx} \\ E_\phi^{Tx} \end{pmatrix} = \begin{pmatrix} G_{\theta\theta} & G_{\phi\theta} \\ G_{\theta\phi} & G_{\phi\phi} \end{pmatrix} \begin{pmatrix} E_\theta^{Tx} \\ E_\phi^{Tx} \end{pmatrix} \qquad (1)$$

$G_{\theta\theta}$, $G_{\theta\phi}$, $G_{\phi\theta}$ and $G_{\phi\phi}$ are four complex gain values that completely characterize the electromagnetic properties of the ray. They can represent either the relative or the absolute complex attenuation and depend on the carrier frequency. The matrix A is called the polarization matrix and depends on the link direction. If A is the polarization matrix for the direct-link, the polarisation matrix for the reverse-link is A^T. The reverse-link is obtained by permuting the transmitter and the receiver. Generally, the polarisation matrix is given by assuming that the base station is the transmitter.

From a strictly theoretical point of view, a set of rays with constant properties models a constant channel. Practically, a constant set of rays also models a wide sense stationary situation as the mobile motion over a short distance.

Between two mobile locations, only a phase offset is added to the polarization matrix, all the other ray characteristics remain unchanged. (see III.C)

B. Ray generation from usual channel models

1) Tapped delay line models

Tapped delay line (TDL) models are the most popular wideband propagation models. The power delay profile is described by a limited number of paths. A path is characterized by a relative amplitude, a Doppler spectrum and a relative delay. The common Doppler spectra are the Rayleigh spectrum also called classical spectrum, the flat spectrum and the Rice spectrum [17, 18]. TDL models are generally defined for the vertical polarization and do not provide any indication on the depolarization. Only $G_{\theta\theta}$ can be determined from the relative amplitude of each path. By default, $G_{\theta\phi}$, $G_{\phi\theta}$ and $G_{\phi\phi}$ are set to zero.

Each path is split in a sub-group of rays with a delay equal to the path delay. The cumulative power of sub-rays coming from the same path is equal to the path power. The sub-ray direction at MS depends on the Doppler spectrum. A classical Doppler spectrum corresponds to a sub-group of rays with equal power and uniformly distributed in a horizontal plane (Clarke's model). A flat spectrum corresponds to a sub-group of rays with equal power and uniformly distributed in 3-D. A Rice Doppler Spectrum is the addition of a Rayleigh Doppler Spectrum with a strong single ray.

The method implemented in Mascaraa to calculate the DoAs at MS from a Doppler Spectrum is based on conclusions of previous studies [19-22]. The authors of these references have developed methods to generate a Rayleigh Doppler Spectrum from a sum-of-sinusoids signal. Three recommendations can be made from the synthesis of all methods: asymmetrical DoA arrangements, random initial phases, high number of sinusoids (at least 10). For the particular case of the Rayleigh Doppler spectrum, these recommendations imply that:

- the phase of $G_{\theta\theta}$ is a random variable uniformly distributed between 0 and 2π.

- $\theta_i = \dfrac{2\pi}{N}(i - 1 + \alpha), \quad i \in [1, N]$ with θ_i the azimuth of the ith sub-ray, N the total number of sub-rays per path and α a U[0,1] random variable.

TDL models do not define the DoAs at BS. In order to be used in MIMO simulation chains, they can be improved by adding to each path an elevation/azimuth at BS [23].

2) Ray-tracing/launching models

The ray-tracing and ray-launching models process all possible rays between a transmitter location and a receiver location. Simulations are based on geometrical optics and the Uniform Theory of Diffraction. They require geographical databases that contain the description of the indoor and/or outdoor environment. This type of models provides immediately all the ray characteristics and is implemented in Mascaraa by reading a result file from a ray-tracing or ray-launching simulation.

3) Scattering or geometrical model

The scattering or geometrical models define a spatial distribution of scatterers in relation to the transmitter or receiver location. A group of near scatterers is called a cluster and could represents a building that reflects waves. Rays are generated by joining the BS to the MS, passing through one or more scatterers. $G_{\theta\theta}$ is deduced from a path loss model. By default, $G_{\theta\phi}$, $G_{\phi\theta}$, $G_{\phi\phi}$ are set to zero. The phase of $G_{\theta\theta}$ is a random variable with uniform distribution U(0,2π).

4) Directional tap models

Directional tap models are based on TDL models. The Doppler spectrum is replaced by two statistical distributions that characterize the power angular spectrum (PAS) at BS and MS. The Laplacian function is generally used. The mean value defines the main path direction. The path splitting method in sub-rays is similar to the one described for TDL models, except for the direction at MS or BS that shall respect the power angular distribution mentioned above. This can be done by: splitting each path in equally spaced sub-rays whose amplitude is given by the PAS distribution or by splitting each path in equally powered sub-rays whose direction is more or less concentrated around the path direction according the PAS distribution. An analysis of the different splitting methods can be found in [24] for the Gaussian distribution.

5) Polarization modeling

Most of the geometrical models or tap models determine only the $G_{\theta\theta}$ component. They can be completed by polarization models that give statistical distributions to characterize three depolarization ratios, $G_{\theta\phi}/G_{\theta\theta}$, $G_{\phi\theta}/G_{\theta\theta}$, $G_{\phi\phi}/G_{\theta\theta}$ [25, 26]. The depolarization ratios can be specific to each ray, identical for all rays or identical for all rays belonging to a same cluster or path. The phase of $G_{\theta\phi}$, $G_{\phi\theta}$, $G_{\phi\phi}$ are random variables with uniform distribution U(0,2π).

III. GENERATING CHANNEL COEFFICIENTS

A. Weighting by antenna pattern

We note h^{dirac} the complex impulse response of the propagation channel.

$$h^{dirac}(t) = \sum_{i=1}^{nbRays} a(i)\delta(t - \tau(i)) \tag{2}$$

$a(i)$ and $\tau(i)$ are respectively the amplitude and delay of the ith ray. $a(i)$ are the channel coefficients.

If the reference system of the antenna pattern is the same as the reference system of the polarization matrix, h^{dirac} is given by the following equation:

$$h^{dirac}(t) = \sum_{i=1}^{nbRays} \begin{pmatrix} G_\theta^{MS}(i) & G_\phi^{MS}(i) \end{pmatrix} A(i) \begin{pmatrix} G_\theta^{BS}(i) \\ G_\phi^{BS}(i) \end{pmatrix} \delta(t - \tau_i)$$
$$= \sum_{i=1}^{nbRays} \begin{pmatrix} G_\theta^{BS}(i) & G_\phi^{BS}(i) \end{pmatrix} A(i)^T \begin{pmatrix} G_\theta^{MS}(i) \\ G_\phi^{MS}(i) \end{pmatrix} \delta(t - \tau_i) \tag{3}$$

$G_\phi^{MS}(i)$ and $G_\theta^{MS}(i)$ are respectively the E_ϕ and E_θ components of the MS antenna gain in the direction of the ith ray. $G_\phi^{BS}(i)$ and $G_\theta^{BS}(i)$ are respectively the E_ϕ and E_θ components of the BS antenna gain in the direction of the ith ray. Equation (3) is valid for any kind of antenna polarization (e.g. linear or circular).

B. MIMO cases

Fig. 2 shows a MIMO configuration with *nbBsSensor* sensors at BS and *nbMsSensor* sensors at MS. A MIMO propagation model shall provide an *nbBsSensor*nbMsSensor* matrix of impulse responses. We note h_{mn}^{dirac}, the impulse response from the mth BS-sensor to the nth MS-sensor
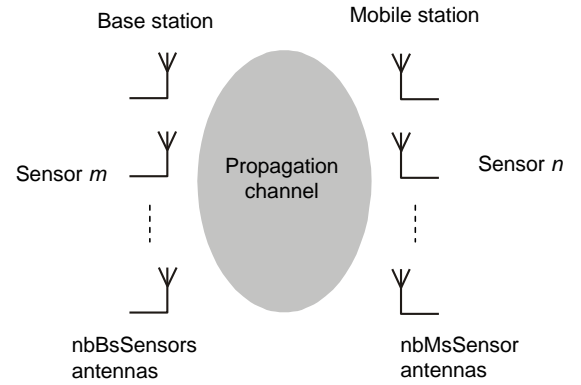

Fig. 2: Example of MIMO configuration

For usual wireless communication systems frequencies (900 MHz - 5 GHz), the distance between sensors is much smaller than the distance between sensors and scatterers. A reasonable approximation is to consider that every SISO channels of a MIMO link have the same physical properties [27]. In this case, (3) can be extended to the MIMO cases by adding a phase offset:

$$h_{mn}^{dirac}(t) = \sum_{i=1}^{nbRays} a(i)e^{j\varphi_m(i)}e^{j\varphi_n(i)}\delta(t - \tau(i)) \tag{4}$$

where $e^{j\varphi_m(i)}$ is the phase offset of ith ray applied to the mth BS-sensor and $e^{j\varphi_n(i)}$ the phase offset applied to the nth BS-sensor. These offsets depend on the 3-D relative position of the sensor compared to the antenna center and the 3-D ray orientation. If the antenna array is assumed to be a uniform linear array, the phase offset between two successive sensors is equal to $2\pi.\delta x.cos(\alpha)/\lambda$, δx being the distance between sensors, λ the wavelength and α the ray direction compared to the antenna array (fig.3). Index p represents either the BS sensors index or the MS sensors index.
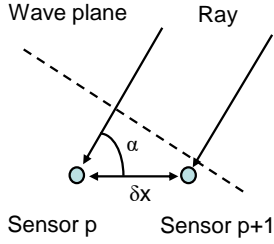
Fig. 3: Phase offset between two sensors

C. Mobile motion simulation

The basic way to compute a series of impulse responses corresponding to the mobile motion is to sample spatially the mobile route and then to compute the set of rays for each position. This solution is very time-expensive. The most efficient solution to simulate the fast fading is to refresh only the phase of the channel coefficients according to the mobile motion. The amplitude, delay and direction remain unchanged during the simulation.

This solution is very similar to that adopted for the extension of SISO models to MIMO applications. The different locations of the mobile can be viewed as a virtual array. In this paragraph, the only case that is considered is a vehicle linear trajectory with a constant speed. This is generally the case over a WSS distance of a few tens of wavelengths. But the method described below could be generalized for other simulation scenarios.

In figure 3, the expressions "sensor p" and "sensor p+1" are replaced by the expressions "mobile position p" and "mobile position p+1". The phase offset of a ray incident to the linear trajectory with an angle α, is equal to $2\pi.\delta x.cos(\alpha(i))/\lambda$. α is deduced from the ray azimuth, the ray elevation and the trajectory direction. δx is the distance between two mobile positions. We note $h_{mn,p}^{dirac}(t)$ the impulse response at position p.

$$h_{mn,p+1}^{dirac}(t) = \sum_{i=1}^{nbRays} a_{mn,p+1}^{'}(i)\delta(t-\tau(i))$$

$$= \sum_{i=1}^{nbRays} a_{mn,p}^{'}(i)e^{2.\pi j.\delta x.cos(\alpha(i))/\lambda}\delta(t-\tau(i)) \quad (5)$$

with $a_{mn,0}^{'}(i) = e^{j\varphi_n(i)}e^{j\varphi_m(i)}e^{j.start(i)}$ (6)

$e^{j.start(i)}$ are random starting phases attributed to each ray using a U(0,2 π) distribution. They simulate a random starting position on the virtual mobile trajectory.

The ratio $\delta x/\lambda$ is called spatial step and is an important parameter of Mascaraa. The setting of this parameter allows the generation of spatial series of correlated or uncorrelated CIRs. A spatial series of correlated CIRs accurately samples the short-term fading. Fig. 4 shows an example with a high spatial selectivity. The fading is generated by recombination of 50 rays having the same delay, the same amplitude and uniformly distributed around the mobile (typical Rayleigh configuration). A fast fading repetition, approximately equal to $\lambda/2$, is observed. A spatial step equal to $\lambda/10$ is unsatisfactory, the amplitude difference between two consecutive position is obviously too high. A spatial step of $\lambda/100$ gives better results. Amplitude discontinuities are lower than 1 % of the amplitude maximal variation. An

intermediate value of $\lambda/50$ is a good trade-off between accuracy and fast processing time (see ch. V).

Most of the time, link-level simulations are performed with correlated CIR series to realistically simulate the fast fading experienced by the mobile. But it is sometimes quicker and more convenient to make the following assumptions. Firstly, the transmitted signal is made up of independent data blocks. Secondly, the CIR is invariant during the block duration. Thirdly, consecutive CIRs are independent. For this kind of link-level simulation, an uncorrelated CIR series is needed. Fig. 5 shows an example with a low spatial selectivity. Rays are distributed uniformly on $10°$. A slower fading repetition is observed, approximately every 20λ. Consequently, the simulation of uncorrelated CIR series for any propagation models requires a minimum channel spatial sampling of about 100λ. The processing time is independent of the spatial step value. (see ch. V)
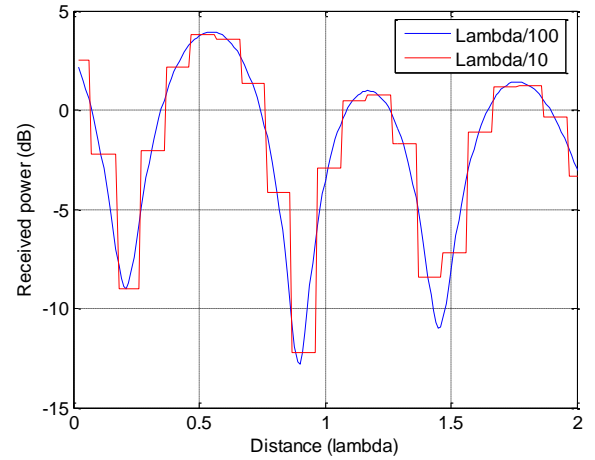

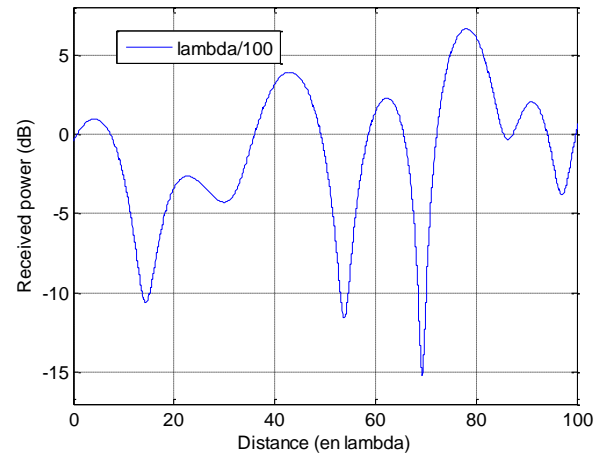Fig. 4: Fading generated from a 360° azimuth distribution

Fig. 5: Fading generated from a 10° azimuth distribution.

IV. FILTERING

A. Impulse response shaping

Section III described a method to process the continuous-time impulse response but a propagation block used in link-level simulation requires a discrete-time impulse response,

sampled at a frequency *fs* equal to the signal sampling frequency. The main problem of the continuous-to-discrete conversion is that the ray delays are not multiples of the sampling period *ts*. A method to sample the impulse response consists in approximating the ray delay to the nearest multiple of *ts* [28]. This ray mapping method is generally used for tap models with a reduced tap number *nbTap*. In this case, the channel filtering is equivalent to a filter of length *nbTap*. The received signal is the sum of *nbTap* copies of the transmitted signal that are multiplied by *a'(i)* and delayed by *τ(i)*. Although this mapping method is very simple, it significantly modifies the space-time characteristics of the original channel and consequently the system performances. Increasing the ray delay accuracy by oversampling the signal could reduce this disadvantage but will increase the filtering processing time. As a result, this method was not adopted in Mascaraa.

Mascaraa processes the filtered time-discrete impulse response $h_{mn}(k)$ following (6).

$$h_{mn}\left(k\right) = \sum_{i=1}^{nbRays} a'_{mn}(i) g\left(k.ts - \tau(i)\right) \qquad (6)$$

with $a'_{nm}\left(i\right) = a(i)e^{j\varphi_n(i)} e^{j\varphi_m(i)} e^{j.\delta x.\cos(a)} e^{j.start(i)} \qquad (7)$

g(t) is the temporal response of the Mascaraa shaping filter. We note *g(f)* the frequency response of this filter. *g(f)* is a raised cosine filter as shown in fig. 6. The flat bandwidth is equal to the transmit signal bandwidth *signalBW* and the maximum total bandwidth *filterBW* is equal to *fs/2* in order to respect the Shannon sampling theorem. *fs* and *signalBW* are two input parameters of Mascaraa. This particular frequency response allows the spectral properties of the transmitted signal to remain unchanged. In case of an ideal channel (dirac with null delay and amplitude of 1), the received signal is equal to the transmitted signal.
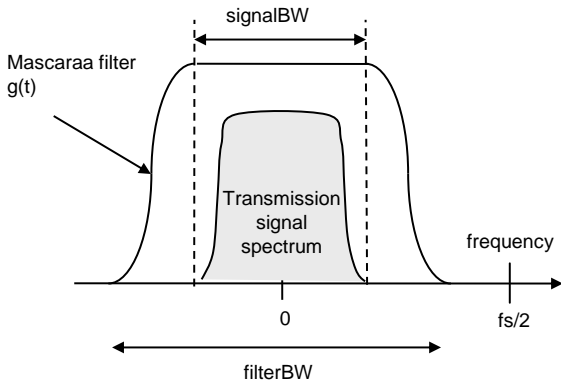

Fig. 6 Frequency response of the Mascaraa shaping filter

The shaping filter method has several advantages
- It does not quantize the ray delays. The simulated power delay profile and Doppler Spectrum are continuous even if the signal bandwidth is high. For each bin of the impulse response, the fast fading is due to the interferences of non-resolvable rays compared to the Mascarraa filter bandwidth.
- The ray delay is arbitrary, i.e. the delay accuracy does not depend on the sampling frequency. Signal oversampling is not required to increase the delay accuracy. The time shifting of a ray can be finely simulated. For instance, the Rake receiver performances can be evaluated precisely.

B. *Mascaraa shaping filter synthesis*

g(t) is generated in two steps. Step 1 is the theoretical definition of *g(f)* as indicated in the previous section. Step 2 is the temporal truncation of *g(t)* that is theoretically time-infinite. *g(t)* is a succession of decreasing amplitude sidelobes. The temporal truncation is done by suppressing the sidelobes, the amplitude of which is below a given threshold of about 40 dB.

This truncation method does not necessarily optimise the length of *g(t)* but minimizes the difference between the specified filter and the realized filter. When the total bandwidth is higher than twice the signal bandwidth, this difference is quasi-null (Fig. 7).
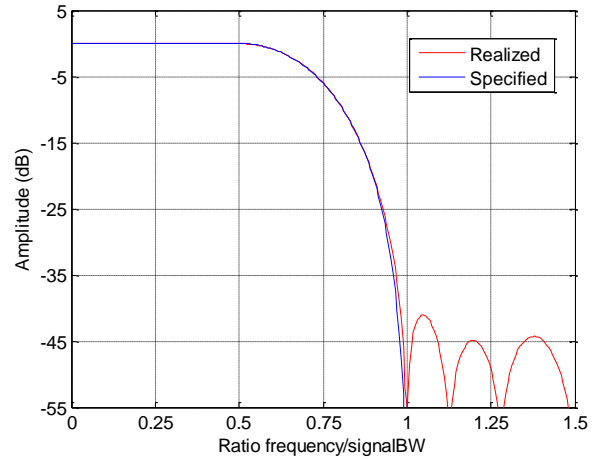

Fig. 7: Realized filter transfer function

The impulse response calculated in (6) is the discrete-time baseband impulse response of the propagation channel. By default, it does not include system specifications as the Rx or Tx Filter used in digital modulation. *g(t)* is not to be confused with the pulse shaping filter used in digital modulation. The expressions "transmitted signal" or "received signal" are not related to digital sequences but respectively to the discrete-time baseband version of the signal before the Tx-antenna and the discrete-time baseband version of the signal after the Rx-antenna.

In some configurations, it could be possible to merge the Mascaraa shaping filter with the Rx/Tx filters or with the transfer function of RF components. This item is not discussed in this paper because it depends on the link-level simulation requirements and can not be generalized for any kind of simulations.

C. *Ray delay accuracy*

According to (6), it would be theoretically possible to compute the impulse response from a set of rays with arbitrary delays. Practically, the continuous-time function *g(t)* may be not analytically defined because of the filter synthesis method (Fourier transform and time truncation). Furthermore, the calculation of $g\left(k.ts - \tau(i)\right)$ during the simulation is unnecessary because *g(t)* is constant during all the simulation.

Mascaraa solves these two problems by processing the time-discrete function *g(k)* before the simulation. *g(k)* is equal to *g(t)* oversampled at *ovSp*fs*. *ovSp* is chosen in order

to not affect the characteristics of the propagation channel. $h(k)$ is given by

$$h_{mn}(k) = \sum_{i=1}^{nbRays} a'_{mn}(i) g\left(k.ovSp - floor\left(\frac{ovSp.\tau(i)}{ts} + 0.5 \right) \right) \quad (8)$$

The delay accuracy is constant for the whole simulation but can be user-defined by changing the value of *ovSp*. Increasing the delay accuracy requires a little more memory space to store *g(k)* but do not affect the impulse response processing time. By default in Mascaraa, *ovSp* is set to 50.

D. Impulse response length optimisation

The impulse response length strongly influences the running time performances of the simulator. It is thus important to evaluate, for a given value of *signalBW*, the optimal values of *filterBW* and *fs* that minimize the length of the impulse response *lengthIR*. *lengthIR* is the sum of the length of *g(t)* noted *lengthFilter* and the length of the propagation channel noted *lengthChannel*. *lengthChannel* is given by (9).

$$lengthChannel = floor\left(\frac{Max(\tau(i)) - Min(\tau(i))}{fs} \right) \quad (9)$$

Fig. 8 gives the relation between *lengthFilter* and the ratio *filterBW/signalBW*. *FilterBW* is equal to *fs*.
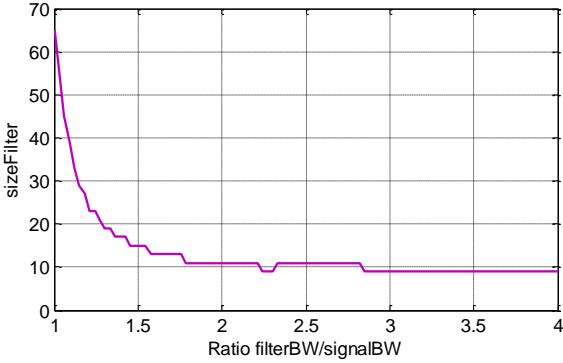

Fig. 8: Filter length variation

LengthFilter is minimum when *filterBW* is maximum that implies that *fs* is maximum. On the other hand, *lengthChannel* increases when *fs* increases. The optimal sampling frequency depends on the propagation channel and the signal bandwidth. A good trade-off is a sampling frequency equal to twice the signal bandwidth, which corresponds roughly to a standard simulation configuration with 2 samples per chip.

E. Amplitude and delay normalization

If a propagation model provides the ray delays and the polarization matrices with absolute values, the impulse response calculated according to (6) expresses an absolute gain as a function of an absolute delay. In this case, the effects due to the transmitter-receiver distance are included in the channel impulse response as well as the wideband effects. Usually, this solution does not suit the simulation requirements for two reasons:
- The results of link-level simulations are usually presented in the form of performance tables that give the error rate as a function to the signal-to-noise ratio (S/N). A convenient

way to modify the S/N value is to assume that the average received power remains constant and that the noise power is set to have the required S/N. In this case, the impulse response power has to be normalized to assure a constant average level at the output of the propagation simulator. Furthermore, to avoid processing errors due to the limited computer precision, it is generally recommended to process data that have the same order of length
- The beginning of the absolute impulse response contains null values equivalent to the shortest ray delay. This null part of the impulse response would unnecessarily slow down the channel filtering while it could be with relative simplicity simulated by shifting the input or output signal of the propagation simulator.

Mascaraa normalizes the absolute impulse response in time and in amplitude. The relative impulse response is given by (10).

$$h_{mn}^{relative}(k) = h_{mn}^{absolute}(k + delay_{abs})/gain_{Abs} \quad (10)$$

delay_abs is the time normalizing factor. It is equal to the index of the first non-null coefficient of the absolute impulse response. It can be negative if the delay of the shortest ray is lower than half of the length of g(t). *gain_abs* is the power normalizing factor. It is calculated in order that the total power of the power delay profile is equal to 1.

F. Filtering

The channel filtering implemented in Mascaraa is based on the Over-and-Add method (OA method) [29, 30]. The time-efficiency of this method is discussed in chapter V by comparing the OA method with two other convolution methods: direct method and tap method.

To illustrate the application of this well-known algorithm, we consider the input signal *e(k)*, the output signal *s(k)* and the impulse response *h(k)* of length *lengthIR*. *e(k)* is divided into section of *lengthIn* data points. The ith section $e^i(k)$ is defined by:

$$e^i(k) = e(k) \text{ for } i.sizeIn \le k < (i+1).sizeIn \text{ and}$$

$e^i(k) = 0$ otherwise. Then $e(k) = \sum_i e^i(k)$

Since convolution is a linear operation, the convolution of *e(k)* with *h(k)* is equal to the sum of $e^i(k)$ convolved with *h(k)*.

$$s(k) = \sum_i s^i(k) = \sum_i e^i(k) * h(k) \quad (11)$$

$s^i(k)$ are sections of length *lengthOut*, equal to *lengthIn+lengthIR-1*. Sections $s^i(k)$ are overlapped by *lengthIR-1* points (fig. 9).

The convolution is made in frequency domain because the convolution via FFT is more efficient for most simulation configurations (section V.C.3). (12) is the transposition in frequency domain of (11).

$$s(k) = \sum_i FFT^{-1}(S^i(f)) = \sum_i FFT^{-1}(H(f).E^i(f)) \quad (12)$$

$S^i(f)$ is the FFT of $s^i(k)$. *lengthOut* is a power of 2. *H(f)* is the FFT of *h(k)* defined over *lengthOut* points. $E^i(f)$ is the FFT of $e^i(k)$ defined over *lengthOut* points. The global computational effort is minimized when *lengthOut* is equal to the lowest power of 2 and when *lengthIn>lengthIR*.
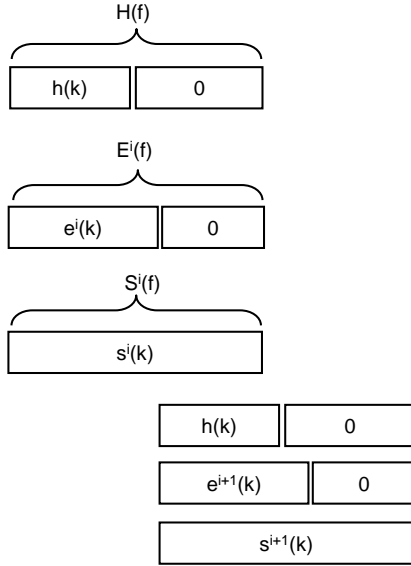
Fig. 9: Overlapp-and-Add convolution

Fig. 10: Sensors rotation and translation definition

V. SOFTWARE IMPLEMENTATION

A. *Coordinate reference system and antenna array definition*

The coordinate reference system allows the coherent definition of:
- The E_ϕ , E_θ components used in the definition of the polarization matrix and the 3-D vectorial antenna pattern
- The direction of rays, paths or clusters according to the propagation model
- The location and orientation of the sensors at MS or BS
- The MS direction

The Mascaraa coordinate reference system consists of two local Cartesian coordinate systems:
- A local Cartesian coordinate system (X_{BS}, Y_{BS}, Z_{BS}) is defined at the base station. Axis Z is the vertical. Axis X points towards the mobile.
- A local Cartesian coordinate system (X_{MS}, Y_{MS}, Z_{MS}) is defined at the mobile. Axis Z is the vertical. Axis X points towards the base station .

The location and orientation of sensors are defined by 6 variables (*x, y, z, rotX, rotY, rotZ*). x, y, z are either the Cartesian coordinates of MS-sensors in (X_{MS}, Y_{MS}, Z_{MS}) or the Cartesian coordinates of BS-sensors coordinates in (X_{BS}, Y_{BS}, Z_{BS}). *rotX, rotY, rotZ* are three successive rotations respectively about X_{MS}, Y_{MS}, Z_{MS} (or X_{BS}, Y_{BS}, Z_{BS}) to point a MS-sensor (or BS-sensor) in a given direction. Fig. 10 illustrates the use of these parameters to create a virtual antenna array for MIMO application. Sensor 1 is defined as the origin of the mobile local coordinate system. The Cartesian coordinates of the other sensors set at the four corners of the computer screen depend on the screen size and tilt. For reasons of clarity, only the rotation of sensor 3 is shown. We assume that the sensor 3 radiation pattern was characterized in an original coordinate system (X_{sensor}, Y_{sensor}, Z_{sensor}) with the antenna boresight in the direction of axis Z_{sensor}. *rotY_3* and *rotZ_3* define respectively the tilt and
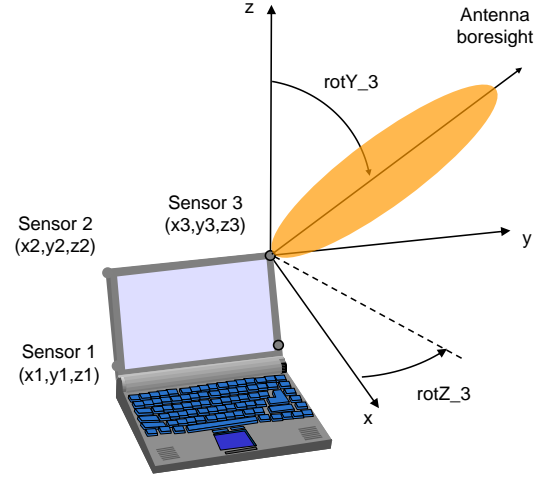
B. *Functional block diagram and configuration parameters*

Mascaraa is a software library written in C Ansi. It is easily portable on various Operating Systems or simulation platforms. The user functionalities are divided into three categories (fig. 11):
- Configuration functions: work session initialization, session parameter setting, session configuration file loading or saving. A work session is related to a MIMO link between a mobile and a base station. Mascaraa is able to create several sessions to simulate several mobile drops during a same system level simulation.
- Preprocessing function: this function gathers all steps described in sections II-III-IV to successively generate the set of rays, the channel coefficients and the first impulse response.
- Simulation functions: impulse response refreshment and channel filtering. These two processes are completely independent. The user is free to update or not the active impulse response used in the channel filtering.

The simulation parameters are:
- The propagation model name
- The random seed that initializes the random generator for the channel coefficients initial phase
- The sensor number at MS or BS
- The carrier frequency
- The signal bandwidth
- The sampling frequency
- For each sensor at BS or BS, a file name that contains the 3-D vectorial and complex antenna pattern (theorical or measured).
- The sensor 3-D location and orientation at MS given in the MS coordinate system
- The sensor 3-D location and orientation at BS given in the BS coordinate system
- The distance in terms of wavelengths between two successive CIRs
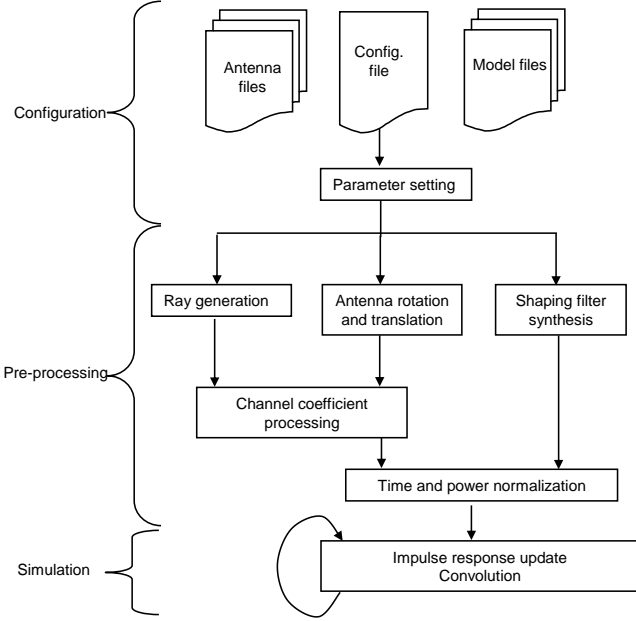- The mobile direction

Fig. 11: Mascaraa block diagram

C. Computing time evaluation

1) Impulse response processing time

Three propagation models are compared in Table I. URB_MED is a typical urban geometrical model at 2 GHz described in [7]. Vehicular A is a TDL model with 6 taps. The indicated processing time is given for a single SISO channel. The computer was a PC Pentium IV 1.8 GHz.

Mascaraa computes the channel transfer function required in the OA method by processing the FFT of the impulse response. According to (6), the impulse response processing time depend on *nbRays* and *lengthFilter* but not on *lengthIR*. In most simulation configurations, it is time-saving to compute the FFT of the impulse response rather than the transfer function from the ray properties.

Model name	Vehicular A, 20 rays/ tap	Vehicular A, 50 rays/tap	URB_MED
nbRays	120	300	650
signalBW	5 MHz	5 MHz	5 MHz
Fs	10 MHz	10 MHz	10 MHz
lengthFilter	10	10	10
lengthIR	37	37	19
IR Processing time	13 µs	30 µs	67 µs

Table 1: Impulse response processing time

The IR processing time includes both the channel coefficient generation and the impulse shaping. For the first two models, the processing times required to compute the taps amplitude only are respectively equal to 4 µs and 9 µs.

To evaluate the run-time efficiency of the Mascaraa impulse response generation, a comparison is made with a common method to process the impulse response. This method is restricted to TDL and tap directional models. The tap complex amplitudes are considered as filtered i.i.d. complex Gaussian variables. To simplify the comparison, we do not take into account the filtering necessary to obtain a particular Doppler spectrum shape. A previous analysis shows that 70 % of the CIR processing time is due to the complex Gaussian variables generation [31]. The processing of an impulse response with 6 taps requires the generation of 12 Gaussian variables. Several algorithms to generate random variables have been implemented. These algorithms are described in [32]. The average processing time of 12 Gaussian variables is around 7 µs depending on the selected random function. This time has the same order of magnitude as the CIR processing time. This brief comparison proves that the method implemented in Mascaraa to process CIRs is not computationally intensive if it is properly time-optimized. The next section describes a simple but time-efficient optimization method based on lookup tables.

2) Use of lookup tables

We reformulate slightly (5) to introduce a new variable $\delta phase$. $\delta phase$ is calculated during the preprocessing step and do not increase the CIR processing time during the simulation.

$$h_{mn,p+1}^{dirac}(t) = \sum_{i=1}^{nbRays} a_{mn,p}'(i)e^{j\delta phase}\delta(t-\tau(i)) \qquad (12)$$

From (12), we can evaluate the number of operations required to compute a CIR: *nbRays* additions and modulo 2π (sum of the angle of $a_{mn,p}'(i)$ with $\delta phase$), *nbRays* cosine functions, *nbRays* sine functions, 2. *nbRays* multiplications, 2.(*nbRays* -1) additions.

Trigonometric operations are time-consuming functions. It is therefore time-saving to replace these functions by lookup tables that contain pre-computed values of cosine and sine functions. The first solution is to replace trigonometric operations by rounding functions (13). We note:

$A(i)=\left|a_{mn,p}'(i)\right|=\left|a_{mn,p+1}'(i)\right|,$

$\beta_p(i)$ the angle of $a_{mn,p}'(i)$,

$\beta_{p+1}(i)$ the angle of $a_{mn,p+1}'(i)$.

$$\begin{aligned}a_{mn,p+1}'(i) &= A.\cos(\beta_p(i)+\delta phase) + jA.\sin(\beta_p(i)+\delta phase)\\&= A.\cos(\beta_{p+1}(i)) + jA.\sin(\beta_{p+1}(i))\\&= A.COS[Round(\beta_{p+1}(i).L/2\pi)] + jA.SIN[Round(\beta_{p+1}(i).L/2\pi)]\end{aligned} \qquad (13)$$

Round() designs the rounding function to the nearest integer. COS[] and SIN[] are trigonometric look-up tables of L points. Mascaraa refines this method by suppressing rounding functions that are time-consuming as well:

$$a_{mn,p+1}'(i) = A.COS[Int\beta_p(i) + Int\delta phase] + jA.SIN[Int\beta_p(i) + Int\delta phase] \quad (14)$$

with $Int\delta phase = Round(\delta phase.L/2\pi)$ and $Int\beta_p(i)$ the angle of $a_{mn,p}'(i)$.

$Int\beta_p(i)$ and $Int\delta phase$ are integer variables defined in [0,L]. $Int\delta phase$ is calculated during the pre-processing step and does not increase the CIR processing time. The conventional solution with trigonometric functions, the solution with rounding functions and the Mascaraa solution are compared in table 2 for the Vehicular A model (20 rays/tap). Rounding operations are implemented with "cast" C-operators.

Method	conventional	Lookup tables (rounding)	Lookup tables (Mascaraa)
time	39 µs	20 µs	13 µs

Table 2: CIR processing time optimization

There are others ways to further decrease the CIR computer time. For instance, [31] presents a method that requires no multiplication. The values of Acos() and Asin() for each ray are stored in lookup tables (2 tables per ray). [19] proposes a hybrid method using linear interpolation. Both methods improve the basic concept of trigonometric look-up tables but make the source code more complex. In the point of view of the authors, a simple use of sine and cosine tables is the best trade-off between source code simplicity and processing time efficiency. Furthermore, we will demonstrate in the next sections that the impulse response processing time is much shorter than the propagation channel convolution time. A reduction of the CIR processing time does not automatically lead to a significant speed improvement of the whole simulation.

3) Filtering computational effort

In this chapter the computational effort of three filtering method is compared: the OA method described in section IV.F; the tap method described at the beginning of section IV.A (sum of *nbTap* shifted copies of the Tx signal); the time method (convolution in time domain). The selected propagation model is a tap model with *nbTap* taps. The input signal to be filtered by the channel contains *nbSamples* and is sampled at twice the chip duration *tc*. The required tap precision is equal to *tc/acFact*, *acFact* being the accuracy factor. Concerning the OA method implemented in Mascaraa, *nbSamples* is equal to *k.lengthIn*, *k* being the number of sections. To simplify the comparison, we do not consider the signal oversampling process necessary in the tap method to achieve the required tap precision and the FFT necessary in the OA method to process the Fourier transform of the impulse response. The computational effort is the number of complex multiplications.

The OA method computes *k* sections of *nbSamples* samples. A section performs two FFTs of *lengthOut* points and an array multiplication of *lengthOut* points. Our FFT algorithm indicates a number of multiplications equal to *n.log2(n)/1.5*, *n* being the size of the FFT. The total number of multiplication is then approximately equal to *k.(lengthOut.(log2(lengthOut)+1))*. The convolution in time domain represents *k.lengthIn.lengthIR* multiplications. In the case of the tap method, the signal has to be oversampled by a factor of *acFact/2*. The number of samples to be filtered is thus equal to *k.lengthIn.actFact/2* and the multiplication number is equal to *k.nbTap.lengthIn.actFact/2*. Table 2 compares the computational effort of the three methods for a set of realistic simulation configurations, with *k* equal to 1. The results show that the OA method is the most time-saving method except in very simplistic configurations where the number of taps and the tap precision are low.

				Computational effort		
nbTap	acFact	lengthIR	lengthIn	Time	Tap	OA
6	2	17	48	816	**288**	456
6	2	32	33	1056	**198**	456
6	2	65	192	12480	**1152**	2314
6	2	128	129	16512	**774**	2314
12	2	17	48	816	576	**456**
12	2	32	33	1056	**396**	456
12	2	65	192	12480	**2304**	2314
12	2	128	129	16512	**1548**	2304
6	4	17	48	816	576	**456**
6	4	32	33	1056	**396**	456
6	4	65	192	12480	**2304**	2314
6	4	128	129	16512	**1548**	2314
12	4	17	48	816	1152	**456**
12	4	32	33	1056	792	**456**
12	4	65	192	12480	4608	**2314**
12	4	128	129	16512	3096	**2314**
6	8	17	48	816	1152	**456**
6	8	32	33	1056	792	**456**
6	8	65	192	12480	4608	**2314**
6	8	128	129	16512	3096	**2314**
12	8	17	48	816	2304	**456**
12	8	32	33	1056	1584	**456**
12	8	65	192	12480	9216	**2314**
12	8	128	129	16512	6192	**2314**

Table 2: Comparison of computational effort between different filtering methods

4) Global simulation duration

In this section, the global processing time to simulate a transmission of 10 minutes (real-time) is evaluated. The simulation configuration is the following:
- Propagation Model: Vehicular A (20 rays/tap)
- Sampling frequency: 10 MHz
- Signal bandwidth: 5 MHz
- Mobile speed: 10 m/s
- Carrier frequency: 2.2 GHz

The sections, defined in the OA method, contain 92 samples, equivalent to a duration of 9.2 µs. Therefore, 10 minutes of simulation are divided in 6.5^E7 sections. Each section requires 37 µs of run-time. The convolution duration is equal to 2400 s. We assume that the impulse response is updated every $\lambda/50$. With a carrier frequency of 2.2 GHz and a mobile speed of 10 m/s, a distance of $\lambda/50$ is covered in 273 µs. During 10 mn, the impulse response is updated 2.2^E6 times. Each impulse response refreshment (Impulse response processing, FFTs,…) requires 37 µs of run-time. The added time due to the impulse response refreshment every $\lambda/50$ is equal to 80 s. The global simulation time is 2480 s.

VI. SIMULATION EXAMPLE

In this section, we give a simulation example where Mascaraa is used to evaluate the performance of a channel parameter estimation method. Estimated parameters are the delay, direction at BS, direction at MS, power in vertical polarization. Figure 12 represents the Azimuth-Delay Power Profile (ADPP) of the propagation channel at BS (Station (BS-ADPP) and at MS (MS-ADPP) in macrocell urban
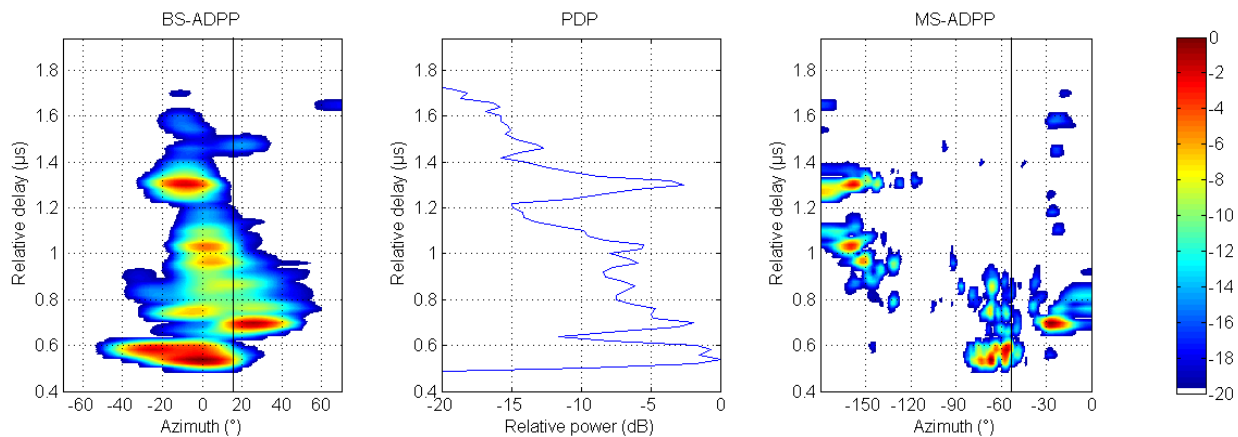
Figure 12.a: Azimut-Delay Power profile at the Base-Station and Mobile Station, measured data
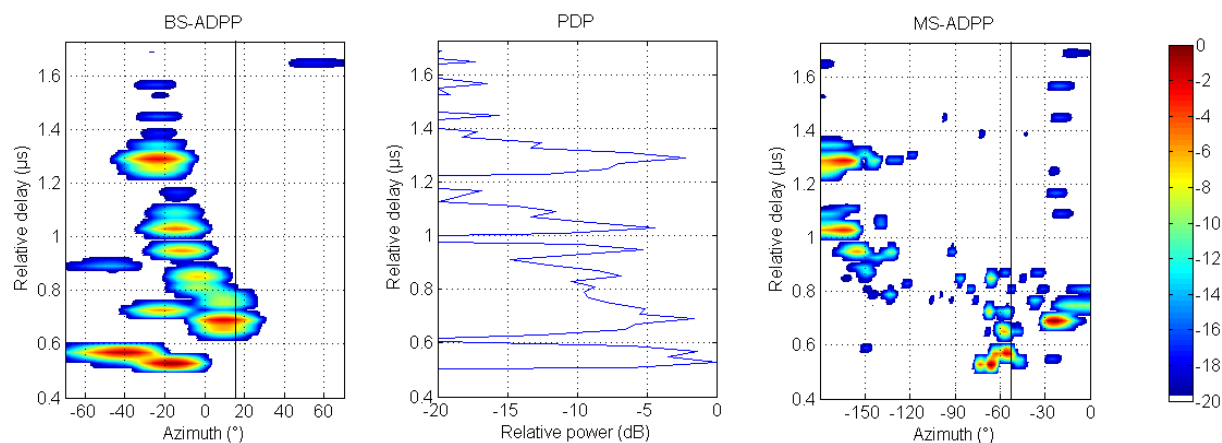

Figure 12.b: Azimut-Delay Power Profile at the Base-Station and Mobile station, synthesized data

environments at 2.2 GHz. Detailed description of the measurement campaign setup, ADPP processing and channel parameters estimation can be found in [12, 33]. Figure 11.a shows the results obtained directly from the measurement data. Figure 11.b shows the results obtained from the data simulated by Mascaraa. The procedure to produce figure 11 was:

1) BS-ADPP and MS-ADPP processing with a conventional beamforming approach, applied to the measurement data (Fig. 11.a)
2) Channel parameter estimation trough detection of local power maxima
3) Impulse response matrix generation with Mascaraa. The configuration for the simulation is the same as the experimental configuration including the carrier frequency, the bandwidth, the antenna array geometry and the antenna pattern
4) BS-ADPP and MS-ADPP processing with a conventional beamforming approach, applied to the synthesized data in step 3.

A visual comparison between fig. 11.a and 11.b shows that the channel parameters estimation method gives satisfactory results. Only the diffuse component is not well modelled which explains the non-continuous shape of the Power Delay Profile (PDP).

Another practical use of Mascaraa is reported in [34]. This paper describes the design steps and final implementation of a MIMO OFDM prototype platform developed to enhance the performance of wireless LAN standards such as 802.11, using multiple transmit and multiple receive antennas. The influence of the propagation channel on code design was analysed through simulation with Mascaraa.

VII. CONCLUSION

The Mascaraa propagation channel simulator was introduced in this paper, from its theoretical basis, to its software implementation. The major characteristics of this simulator are:
- Compatibility with most of propagation models. Each propagation model is converted in a set of rays
- Compatibility with all wireless communication standards
- Free choice of any antenna array geometry or any polarization or any antenna pattern
- Optimized algorithms to process the impulse responses and perform the channel filtering efficiently
- Compatibility with multi-sensor radio access schemes: impulse response matrices instead of single impulse response are generated
- Simulation of the mobile motion: MIMO correlation matrices can be computed with a relative simplicity
- Arbitrary delays of rays or paths: they are independent from the signal sampling frequency
- High operating system or simulation platform portability (written in standard C).

All these functionalities explain why Mascaraa is a versatile and efficient propagation channel simulator mainly dedicated to MIMO link-level simulations. Future work will

focus on the use of Mascaraa to explore an important issue in MIMO technologies: the joint-impact of antenna array design and signal processing algorithms on system performances.

Mascaraa is licensed under the GNU General Public License. C-language routines that implement this design are available via e-mail from the authors.

VIII. REFERENCES

[1]   D. Gesbert, M. Shafi, S. Da-shan, P. J. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems", *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 281, 2003.

[2]   A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels", *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 684, 2003.

[3]   K. Yu and B. Ottersten, "Models for MIMO propagation channels: a review", *Wireless Communications and Mobile Computing*, vol. 2, pp. 653, 2002.

[4]   A. F. Molisch, "Effect of far scatterer clusters in MIMO outdoor channel models", presented at Vehicular Technology Conference (VTC 2003-Spring), 2003.

[5]   J. P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation", *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1211, 2002.

[6]   K. I. Pedersen, J. B. Andersen, J. P. Kermoal, and P. Mogensen, "A stochastic multiple-input-multiple-output radio channel model for evaluation of space-time coding algorithms", presented at Vehicular Technology Conference (VTC-Fall 2000), 2000.

[7]   R. B. Ertel, P. Cardieri, K. W. Sowerby, T. S. Rappaport, and J. H. Reed, "Overview of spatial channel models for antenna array communication systems", *IEEE Personal Communications*, vol. 5, pp. 10, 1998.

[8]   A. F. Molisch, "A generic model for MIMO wireless propagation channels in macro- and microcells", *IEEE Transactions on Signal Processing*, vol. 52, pp. 61, 2004.

[9]   S. M. Seedahmed, M. Z. Hussain, and P. O'Sheay, "Space-time Geometrical-Based Channel Models: A comparativ Study", presented at Australian Telecommunication Networks and Applications Conference, 2003.

[10]   "Spatial channel model for Multiple Input Multiple Output (MIMO) simulations", 3GPP TR 25.996 V6.1.0, 2003.

[11]   X. Hao, D. Chizhik, H. Huang, and R. Valenzuela, "A generalized space-time multiple-input multiple-output (MIMO) channel model", *IEEE Transactions on Wireless Communications*, vol. 3, pp. 966, 2004.

[12]   J. M. Conrat and P. Pajusco, "Typical MIMO propagation channels in urban macrocells at 2 GHz", presented at European Wireless Conference (EW2007), Paris, 2007.

[13]   F. A. Agelet, A. Formella, J. M. H. Rabanos, F. I. d. Vicente, and F. P. Fontan, "Efficient Ray-Tracing Acceleration Techniques for Radio Propagation Modeling", *IEEE Transactions on Vehicular Technology*, 2000.

[14]   T. Fugen, J. Maurer, T. Kayser, and W. Wiesbeck, "Verification of 3D Ray-tracing with Non-Directional and Directional Measurements in Urban Macrocellular Environments", presented at VTC 2006, Melbourne, 2006.

[15]   M. A. Jensen and J. W. Wallace, "A review of antennas and propagation for MIMO wireless communications", *IEEE Transactions on Antennas and Propagation*, vol. 52, pp. 2810, 2004.

[16]   M. Steinbauer, A. F. Molisch, and E. Bonek, "The double-directional radio channel", *IEEE Antennas and Propagation Magazine*, vol. 43, pp. 51, 2001.

[17]   J. Kivinen, Z. Xiongwen, and P. Vainikainen, "Empirical characterization of wideband indoor radio channel at 5.3 GHz", *IEEE Transactions on Antennas and Propagation*, vol. 49, pp. 1192, 2001.

[18]   Z. Xiongwen, J. Kivinen, P. Vainikainen, and K. Skog, "Characterization of Doppler spectra for mobile communications at 5.3 GHz", *IEEE Transactions on Vehicular Technology*, vol. 52, pp. 14, 2003.

[19]   L. Yunxin and H. Xiaojing, "The simulation of independent Rayleigh faders", *IEEE Transactions on Communications*, vol. 50, pp. 1503, 2002.

[20]   Z. Yahong Rosa and X. Chengshan, "Simulation models with correct statistical properties for Rayleigh fading channels", *IEEE Transactions on Communications*, vol. 51, pp. 920, 2003.

[21]   M. F. Pop and N. C. Beaulieu, "Limitations of sum-of-sinusoids fading channel simulators", *IEEE Transactions on Communications*, vol. 49, pp. 699, 2001.

[22]   M. Patzold and F. Laue, "Statistical properties of Jakes' fading channel simulator", presented at 48th IEEE Vehicular Technology Conference, 1998.

[23]   P. Pajusco, "Experimental characterization of DOA at the base station in rural and urban area", presented at Vehicular Technology conference (VTC 98), 1998.

[24]   M. Pätzold, *Mobile fading channels*: John Wiley &Sons, 2002.

[25]   L. Vuokko, A. Kainulainen, and P. Vainikainen, "Polarization behavior in different urban radio environments at 5.3 GHz", presented at IWS / WPMC 05, Aalborg, 2005.

[26]   A. Dunand and J.-M. Conrat, "Dual-Polarized Spatio-temporal Characterization in urban macrocells at 2 GHz", presented at Vehicular Technology Conference (VTC2007 Fall), Baltimore, 2007.

[27]   K. H. Ng, E. K. Tameh, and A. R. Nix, "Modelling and performance prediction for multiple antenna systems using enhanced ray tracing", presented at IEEE Wireless Communications and Networking Conference, 2005.

[28]   S. Morosi, M. Tosi, E. D. Re, R. Fantacci, and P. Riva, "Implementation of a wideband directional channel model for a units link level simulator", presented at Globecom 2003, 2003.

[29]   M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of communication Systems*: Plenum Press, New York, 1992.

[30]   E. C. Ifeachor and B. W. Jervis, *Digital signal processing: a pratical approach*: Addison-Wesley, 1993.

[31]   M. Patzold, R. Garcia, and F. Laue, "Design of high-speed simulation models for mobile fading channels by using table look-up techniques", *IEEE Transactions on Vehicular Technology*, vol. 49, pp. 1178, 2000.

[32]   W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. lannery, *Numerical recipes in C, The Art of Scientific Computing, Second Edition*: Cambridge University Press, 1992.

[33]   J. M. Conrat and P. Pajusco, "Clusterization of the MIMO Propagation Channel in urban macrocells at 2 GHz", presented at European Conference on Wireless Technology (ECWT), Paris, 2005.

[34]   A. G. Fàbregas, M. Guillaud, D. Slock, G. Caire, K. Gosse, S. Rouquette, A. R. Dias, P. Bernardin, X. Miet, J.-M. Conrat, Y. Toutain, A. Peden, and Z. Li, "A MIMO-OFDM Testbed for Wireless Local Area Networks", *EURASIP Journal on Applied Signal Processing*, vol. ID 18083, 2006.