



## Wireless Communications

# Simulation of Multipath Fading Channels

**Author:**

Mattia Lecci (1153428)

### Abstract

Throughout the years, many wireless channel simulators have been proposed with different performance objectives. In this project I will implement 8 of them, explaining how they work, the differences from one another and, finally, testing statistical and computation performance, comparing all of them in order to find their strengths and weaknesses.

As expected, there is no single winner, but just two of them will show appreciable characteristics in separate performance indexes. You will see how the best performer will have below average statistical properties and how, maybe surprisingly, the original Clarke's model will actually yield the best overall statistical performance with comparable computational performance to more modern solutions

# 1 Introduction

Since the Seventies, simulating a good wireless channel was recognized to be fundamental for wireless systems design. It was acknowledged right from the start for being simpler, cheaper and more repeatable than a channel sounding: no external expensive gear was required and a new simulation could be done in a matter of seconds or less.

Early on, a simple yet fairly realistic channel model was presented by Clarke [Cla68], on which all the subsequent simulators were based on and many different statistics were extrapolated. A few years later, Jakes created the de facto standard simulator [Jak74] for many years to follow. His proposal was to artificially impose certain symmetries to how the channel behaves, thus being able to simulate an approximation of Clarke's model but only with a fourth of the oscillators. Jakes' approach, though, had some problems and throughout the years many proposals ([A1],[A2],[A3],[C1],[C2]) have been done in order to fix most, if not all of them.

Finally, while proposing a new simulator, *Xiao, Zheng and Beaulieu* [B1] recognized how well *Clarke's* model actually behaves even with a low number of sinusoidal components. In fact, they proved how fast it converges to the ideal formulas, thus not needing as much oscillators as *Jakes* (and all those that followed) actually thought.

The report is structured as follow: in Section 2 I will present the technical aspects of the project, starting from Subsection 2.1 which delineates the main objectives, then in Subsection 2.2 the mathematical models of all the implemented simulators will be carefully described and in Subsection 2.3 an idea of the code structure will be presented. Finally, in Subsection 2.4 I will briefly talk about a few complications encountered while completing this project. In Section 3, then, the results will be presented and lastly in Section 4 the conclusions will be drawn.

## 2 Technical Approach

In this Section all the technical aspects of the project will be presented. A quick explanation of the various simulators implemented will follow in Subsection 2.2, following the history of the subject. That is the reason why the order of the citations will be different from the one proposed. Note that all of the equations have been slightly modified in order to obtain unit power fading channels and to unify the notation.

### 2.1 Objectives

The main objective of the project is to evaluate the performance of different types of wireless channel simulators. In particular, I'm interested, here, in simulating a Rayleigh fading channel, namely a channel where no Line-Of-Sight (LOS) component is present. I will show both statistical and performance results for all of the 8 simulators implemented, comparing them to one another.

### 2.2 Mathematical models used

Almost all of the references start by introducing the ideal statistical properties that a Rayleigh channel should have, obtained for the classical model of such channel. This model is presented in [Cla68] and it is often referred to as **Clarke's** 2D isotropic (both scattering and antenna gain) Rayleigh fading model, given by

$$X(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^N e^{j(2\pi f_d \cos \alpha_n t + \phi_n)} \quad (1)$$

where  $N$  is the number of propagation paths,  $f_d$  is the maximum Doppler frequency,  $\alpha_n$  is the angle of arrival of the  $n$ -th ray and  $\phi_n$  its initial phase. Both  $\alpha_n$  and  $\phi_n$  are uniformly distributed in  $(-\pi, \pi]$  for all  $n$  and they are mutually independent, for a total of  $2 \times N$  random variables. Since in general many rays reach the receiver at the same time, the *Central Limit Theorem (CLT)* justifies the approximation of the channel to a Complex Normal ( $\mathcal{CN}$ ) distribution. Actually the independence of real and imaginary part is not trivial, but it will not be further clarified here. From this, we know that the magnitude of a Complex Normal random variable yields a Rayleigh distributed one (since it's equivalent to the euclidean norm of a 2D Gaussian random vector) and, by symmetry of the distribution (given by the independence of real and imaginary part of the Complex Gaussian), the phase is uniformly distributed in  $(-\pi, \pi]$ . In formulas,

$$f_{|X|}(x) = 2x e^{-x^2}, \quad x \geq 0 \quad (2a)$$

$$f_{\theta}(x) = \frac{1}{2\pi}, \quad x \in (-\pi, \pi] \quad (2b)$$

As  $N$  tends to infinity, defining  $X(t) = X_c(t) + jX_s(t)$  it is possible to prove the following correlations:

$$R_{X_c X_c}(\tau) = E[X_c(t)X_c(t - \tau)] = \frac{1}{2}J_0(2\pi f_d \tau) \quad (3a)$$

$$R_{X_s X_s}(\tau) = \frac{1}{2}J_0(2\pi f_d \tau) \quad (3b)$$

$$R_{X_c X_s}(\tau) = R_{X_s X_c}(\tau) = 0 \quad (3c)$$

$$R_X(\tau) = E[X(t)X^*(t - \tau)] = J_0(2\pi f_d \tau) \quad (3d)$$

$$R_{|X|^2}(\tau) = 1 + J_0^2(2\pi f_d \tau) \quad (3e)$$

Where  $J_0(x)$  is the zero-order Bessel function of the first kind, defined as

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \cos \theta) d\theta \quad (4)$$

As you can see, all of these correlations are obtained from a *Wide Sense Stationary* (WSS) process, since they only depend on the variable  $\tau$ .

Another two very interesting properties which can be extracted from Clarke's model are called *Level Crossing Rate* (LCR) and *Average Fade Duration* (AFD). They both characterize what happen at certain thresholds for the wireless channel and it's generally a higher order behavior: while LCR determines the rate at which the envelope crosses a thresholds with positive slope, AFD indicates for how long the channel will stay below the given threshold in average. They are clearly important parameters to consider when designing a wireless system and the appropriate channel coding. Ideally, their formulas are respectively:

$$L_{|X|}(\lambda) = \sqrt{2\pi} f_d \lambda e^{-\lambda^2} \quad (5)$$

$$T_{|X|}(\lambda) = \frac{e^{\lambda^2} - 1}{\sqrt{2\pi} f_d \lambda} \quad (6)$$

where  $\lambda$  is the normalized fading envelope threshold defined as  $\lambda = |X_{thr}|/|X_{rms}|$ . Since we are dealing with unit power simulations, then,  $\lambda$  is simply equal to the threshold itself.

Since Clarke's model deals with multiple complex sinusoids and random variables, which are both computationally expensive to calculate, **Jakes** proposed in [Jak74] its well known simplification of such model, which basically became a standard for wireless channel simulation for over 20 years. In order to cut down on computational complexity he makes some assumptions: instead of being random variables, he forces  $\alpha_n = \frac{2\pi n}{N}$  and correlates  $\phi_n$  in quadruplets in order to obtain the following simplified model:

$$X_c(t) = \sqrt{\frac{2}{N}} \left[ \cos(2\pi f_d t) + \sum_{n=1}^M 2 \cos\left(\frac{\pi n}{M}\right) \cos(2\pi f_d \cos \alpha_n t) \right] \quad (7a)$$

$$X_s(t) = \sqrt{\frac{2}{N}} \left[ \cos(2\pi f_d t) + \sum_{n=1}^M 2 \sin\left(\frac{\pi n}{M}\right) \cos(2\pi f_d \cos \alpha_n t) \right] \quad (7b)$$

You can see that the model is now fully deterministic and there are about a quarter of the oscillators of the corresponding Clarke's model. In fact, by defining  $N = 4M + 2$ , there are only  $M + 1$  low frequency oscillators needed. Note that the directions with maximum Doppler spread are forcefully kept.

This, though, comes at a price: **Pop** and **Beaulieu** state in [A1] that Jakes' simulator is not even stationary and yields poor higher order statistics. To overcome this phenomena a simple modification is proposed: the addition of an initial random phase to the low frequency oscillators. From Eqs. 7a,7b the oscillator terms become:  $\cos(2\pi f_d t + \phi_0)$  and  $\cos(2\pi f_d \cos \alpha_n t + \phi_n)$ , where  $\phi_n$  are mutually independent uniform random variables in  $(-\pi, \pi]$ .

Now, a small addition that I did with respect to the original paper is the addition of the multichannel support. This may be useful in different interesting scenarios: multiple independent channels are usually used to model a frequency selective fading and MIMO systems. I, instead, used this feature to estimate all the statistics of the simulators without relying on any ergodicity. For the case of the Pop-Beaulieu simulator, this addition is very simple: calling  $X_k(t)$  the  $k$ -th channel ( $k = 1, 2, \dots, K$ ), we just need to generate  $K \times (M + 1)$  random variables  $\phi_{k,n}$ .

In 2002, **Zheng** and **Xiao** [C2]. In this paper a model for multichannel simulation is directly given, with the following real and imaginary components:

$$X_{k,c}(t) = \frac{1}{\sqrt{M}} \sum_{n=1}^M \cos \left( 2\pi f_d t \cos \left( \frac{2\pi n - \pi + \theta_k}{4M} \right) + \phi_{n,k}^{(c)} \right) \quad (8a)$$

$$X_{k,s}(t) = \frac{1}{\sqrt{M}} \sum_{n=1}^M \cos \left( 2\pi f_d t \sin \left( \frac{2\pi n - \pi + \theta_k}{4M} \right) + \phi_{n,k}^{(s)} \right) \quad (8b)$$

where  $\theta_k$ ,  $\phi_{n,k}^{(c)}$  and  $\phi_{n,k}^{(s)}$  are mutually independent random variables uniformly distributed in  $(-\pi, \pi]$ , for a total of  $K \times (2M + 1)$  random variables. This ensures that all the channels have the same statistical properties while being uncorrelated. Note that with respect to the Pop-Beaulieu model it adds randomness on the angle of arrival  $\alpha_n$  and uncorrelates the initial phases of real and imaginary components, while not retaining the angles with maximum Doppler spread, having then  $N = 4M$ . This means more than double the quantity of random variables required to perform the simulation.

Later in 2002, **Li** and **Huang** [C1] proposed another approach to a multichannel simulator. Instead of randomizing the directions of arrival, they follow a deterministic approach, similar to Jakes'. Defining  $\alpha_{n,k} = \frac{2\pi n}{N} + \frac{2\pi k}{NK} + \alpha_{0,0}$  for  $n = 0, \dots, N - 1$  and  $k = 0, \dots, K - 1$ , the formulas for the  $k$ -th ray are:

$$X_{k,c}(t) = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} \cos \left( 2\pi f_d t \cos \alpha_{n,k} + \phi_{n,k}^{(c)} \right) \quad (9a)$$

$$X_{k,s}(t) = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} \sin \left( 2\pi f_d t \sin \alpha_{n,k} + \phi_{n,k}^{(s)} \right) \quad (9b)$$

as usual  $\phi_{n,k}^{(c)}$  and  $\phi_{n,k}^{(s)}$  are mutually independent random variables uniformly distributed in  $(-\pi, \pi]$ , for a total of  $2 \times M \times K$  random variables. It is highlighted the fact that any combination of sine and cosine functions will not affect the actual statistics of the channels. The choice  $\alpha_{0,0}$  is suggested by the authors to be in  $(0, \frac{2\pi}{NK}) \setminus \{\frac{\pi}{NK}\}$ . In the end I decided to use their same initial angle, meaning  $\alpha_{0,0} = \frac{\pi}{2NK}$ . The

paper then also tries to reduce the high cost of calculating trigonometric functions by proposing different approximations and comparing then the results. I decided, though, to not implement this further in-depth analysis.

Going on to 2003, **Zheng** and **Xiao** [A2] further enhance their simulator by adding a random gain to each oscillator:

$$X_{k,c}(t) = \frac{1}{\sqrt{M}} \sum_{n=1}^M \cos(\psi_{n,k}) \cos \left( 2\pi f_d t \cos \left( \frac{2\pi n - \pi + \theta_k}{4M} \right) + \phi_k \right) \quad (10a)$$

$$X_{k,s}(t) = \frac{1}{\sqrt{M}} \sum_{n=1}^M \sin(\psi_{n,k}) \cos \left( 2\pi f_d t \cos \left( \frac{2\pi n - \pi + \theta_k}{4M} \right) + \phi_k \right) \quad (10b)$$

Note that the randomization of the angle of arrival is just the same as their 2002 paper but there are a couple of differences: the initial phase of the oscillators is now constant for all of the oscillators of the same channel (and equal for real and imaginary parts, similarly to the Pop-Beaulieu model), whereas the difference between oscillators of the same channel is given by the random amplitude, which, even though it's different, it is correlated between real and imaginary components ( $\psi_{n,k}$  is the same for both of them). A total of  $K \times (M + 2)$  is needed.

In 2006, the most recent paper about *Sum of Sinusoids* simulator that I considered was written by **Xiao**, **Zheng** and **Beaulieu** [B1]. Once again, the simulator is a *SoS* and introduces slight differences with respect to the others

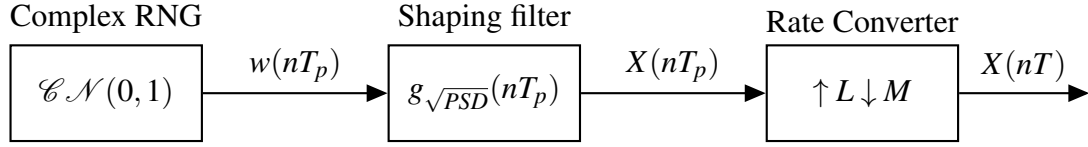
$$X_{k,c}(t) = \frac{1}{\sqrt{M}} \sum_{n=1}^M \cos \left( 2\pi f_d t \cos \left( \frac{2\pi n + \theta_{n,k}}{M} \right) + \phi_{n,k} \right) \quad (11a)$$

$$X_{k,s}(t) = \frac{1}{\sqrt{M}} \sum_{n=1}^M \sin \left( 2\pi f_d t \cos \left( \frac{2\pi n + \theta_{n,k}}{M} \right) + \phi_{n,k} \right) \quad (11b)$$

The main differences with previous models are: no random amplitude for the oscillators is present, both  $\theta$  and  $\phi$  are independent between channels and oscillators (with a total number of  $2 \times K \times M$  random variables required) although they are the same for real and imaginary components, the angle of arrival  $\alpha_{n,k}$  has changed using only  $M$  instead of  $4M$  as the denominator. Similarly to the 2 previous papers from Zheng and Xiao, the angle of arrival is randomized but within sectors, in order to force a "more uniform" distribution, particularly for small values of  $M$ , with respect to the pure uniform distribution of the original Clarke's model.

As the last one, I decided to keep the **Komninakis** simulator [A3] since it's completely different from all the others, even though it's actually from 2003. In fact, while all the previous ones are based on the *Sum of Sinusoids* method, this one works based on a *filtering* method.

The basic idea is the following: starting from a white process  $w(t)$  (which has constant *Power Spectral Density*), in principle it is possible to obtain a stochastic process  $X(t)$  with arbitrary PSD filtering it through an *LTI* system with frequency response  $G_{\sqrt{PSD}}(f)$  since  $P_X(f) = P_w |G_{\sqrt{PSD}}(f)|^2$ , where the magnitude of



the frequency response of the filter equals the square root of the desired PSD. In practice, though, it is not trivial to create a filter with arbitrary shape. In its paper, Komninakis proposes a way of approximating the magnitude of an IIR filter to a desired shape (in particular the classical PSD given by Clarke's model). There is one more problem to solve: particularly when modeling bit-level simulations, the sampling period  $T$  may be much smaller than the *coherence period* usually defined as  $T_{coh} = \frac{1}{f_d}$ . Equivalently, it means that the sampling frequency  $F = \frac{1}{T} \gg f_d$ , thus a very narrowband filter  $g_{\sqrt{PSD}}$  would be required in order to correctly produce the desired output. It is widely known, though, that narrowband filters tend to require very high order which mean lots of calculations per sample to be done, very long transients and possible numerical instabilities given by the fact that poles have to be very close to the unit circle. This is definitely not the way to proceed. It is much wiser to compute only one or two filters with fixed values for the product  $f_d T_p$  in order to balance order, stability and complexity, considering that an interpolation (or in general a rate conversion) will have to be done later. Note that the most correct way to proceed with the interpolation would be an up-sampling with zero padding followed by a low pass filter (either using low order IIR filter or a polyphase FIR filter). If the product  $f_d T$  is small, though, it means that the channel will be slowly changing with respect to the chosen sampling period. This would allow us to use a simpler *spline* interpolation.

The big disadvantage of this simulator, then, becomes the high number of random variables needed. This depends, though, on the value of the product  $f_d T_p$  for the reference shaping filter as well as the the actual product  $f_d T$  needed.

## 2.3 Scenario and Implementation

The project has been implemented using MATLAB<sup>®</sup> R2017a using the student license provided by the department.

A separate function for each simulator has been created as well as the interface `createChannel` to create a realization of a channel with all different kinds of parameters, such as  $f_d$ ,  $T$ , duration (either in number of samples, seconds or multiples of  $T_{coh}$ ), type of simulator, number of oscillators (i.e. the parameter  $M \approx N/4$  for most simulators), number of independent channels and the interpolation method for the Komninakis simulator (either by `filter`, `spline`, `pchip` or `linear` interpolation). For what concerns the Komninakis simulator, shaping filter's coefficients were taken from [CB02, p. 317] which assumes  $f_d T_p = 0.1$ .

Separate functions for the computation of all the statistics required have been done and again all of them have multiple optional parameters to be set. Note that when talking about correlation I mean the statistical one. Hence, I also created a function that computes an estimate of the statistical correlations (all of those described in Eqs. 3) given a number of independent channels (which can be easily created exploiting the Name-Value parameter `NChannels` from the function `createChannel`). The only exception to this rule is Jakes' simulator since it's fully deterministic. The standard temporal correlation has been used in this specific case.

More functions were created in order to properly and simply plot all of the numerous statistics and to ensure a homogeneous look. Finally, different `main`s were created to obtain different types of results in an orderly manner and the possibility to store and load statistics has been added in order to promptly plot previously computed statistics.

## 2.4 Complications found

No major complications were found while doing this project. The only thing to be careful about was the different notation and normalization used in the various papers. Towards the end it has also been non trivial how to choose all of the numerous parameters for the simulation in order to properly highlight strengths and weaknesses of all the different models and how certain parameters affect the results.

## 3 Results

I'll present here statistical and performance results from all of the implemented simulators. The specific parameters have been chosen differently for almost every graph in order to emphasize its characteristics. For more details check out all the different `main` scripts that can be found in GitHub's folder `Code/`. Important parameters that I kept constant are  $M = 8$  (the number of oscillators for the simulators. Note that for *Clarke's* simulator  $N = M$ , while for all the other *SOS* simulators  $N \approx 4M$ ) and the interpolation method (for *Komninakis'* simulator) chosen to be `pchip`. With this, Matlab refers to a piecewise cubic polynomial interpolation with  $C^1$  connections, instead of  $C^2$  connection as for the more famous `spline`. A motivation for this choice will be given later on.

For all of the plots of the statistical results, a thick black line is firstly plotted with the ideal function, as described at the beginning of Section 2.2.

### 3.1 Probability Distribution Functions

The first results presented are simple Probability Distribution Functions, for both magnitude and phase of the channels. Assuming stationarity, many independent channels of every simulators were created and



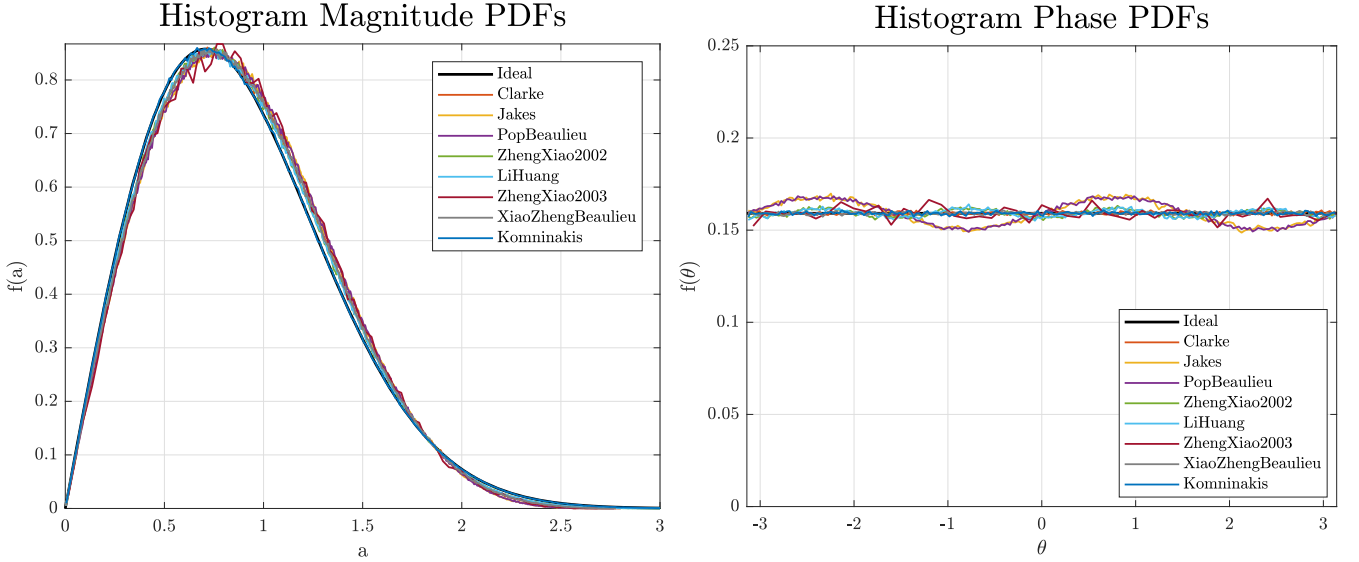


Figure 1: Histograms of Magnitudes and Phases of all the implemented simulators. Note the aforementioned problem with ZhengXiao2003

Matlab's `histogram` function was called on a specific samples across the channels. This was done to follow the probabilistic nature of the problem, avoiding the concept of ergodicity. One notable exception is *Jakes'* simulator, which is fully deterministic, and thus its "PDF" was calculated on the single realization. Another exception to this rule is ZhengXiao2003: as it turns out, it is not really stationary. Calculating its PDF on the first sample, in fact, returned a very different result. In order to avoid this artifact, I created longer channels (about  $T_{coh}$ ) and used the last sample instead. By doing this, though, I also had to provide less channels in order to not exceed memory limitations. The effect of this can be seen in Fig. 1, where ZhengXiao2003 has a rougher line.

Note, though, that all of the simulators approximate very accurately both PDFs (Eqs. 2), with the exception of *Jakes'* and *PopBeaulieu's* simulators which clearly have an oscillatory behavior instead of a constant one for what concerns the phase PDF.

### 3.2 Correlations

Moving on to the correlations (Eqs. 3, Figs. 2), it is clear that once again *Jakes'* and *PopBeaulieu's* simulators cannot really match up with more modern solutions, resulting very early very imprecise especially when dealing with single real or imaginary components. For what concerns the correlation of the signal  $X(t)$  as a whole, instead, they both tend to behave better, but still in the real part of  $R_X$  they are both the first ones to significantly diverge from the ideal function.

*Komninakis'* simulator is the other one that significantly diverges from the ideal case when non-zero functions are involved. Since its core idea is to design a filter that tries to mimic a given PSD (which I recall from *Wiener–Khinchin theorem* is the *Fourier Transform* of the autocorrelation function of the

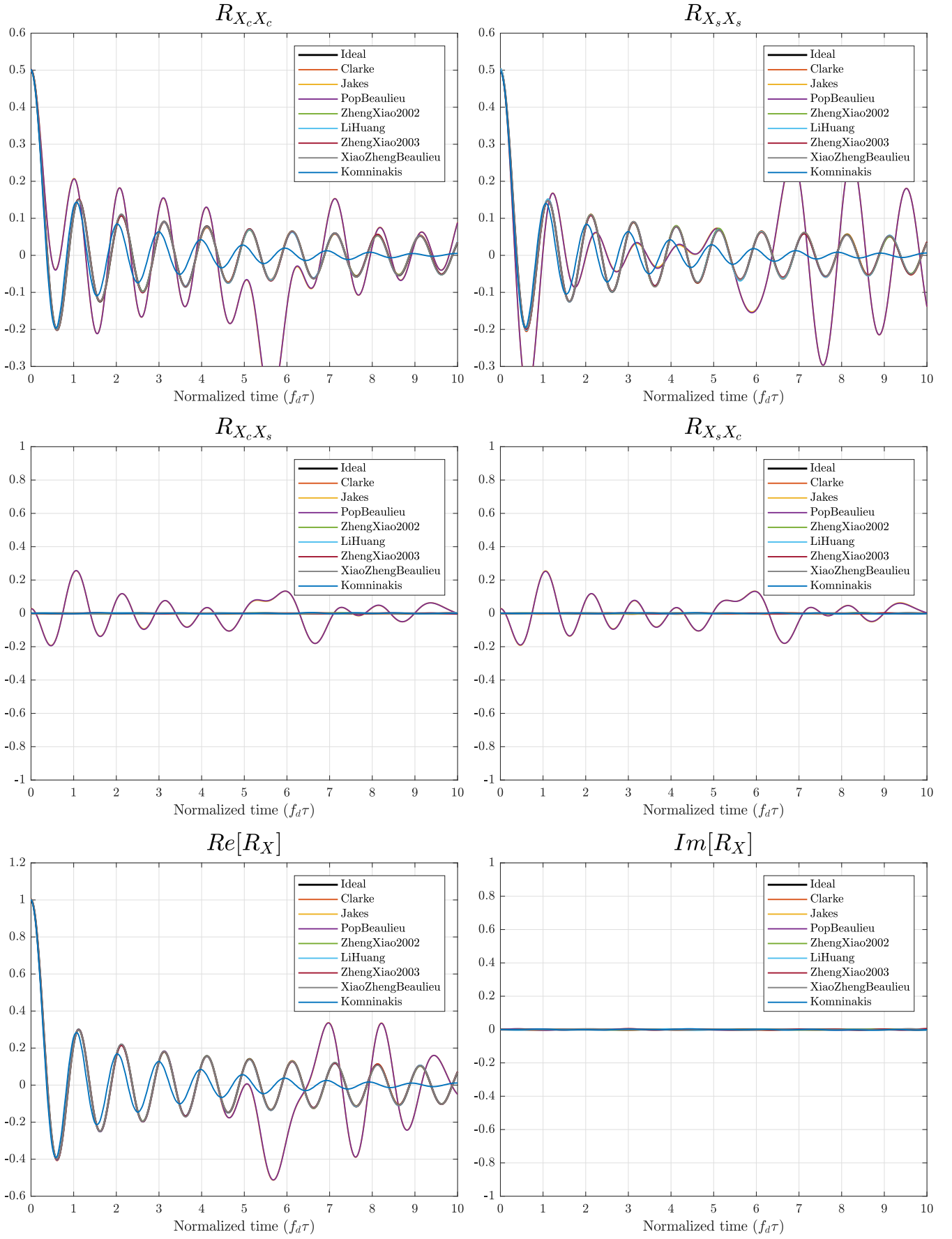


Figure 2: Cross correlations between real and imaginary components

stochastic process), the reason why this happen should be a poor approximation of such filter. I also recall that I simply took the filter's coefficients from [CB02] without further investigations. It's clear from Fig. 2, though, that both the shape (quick ripple attenuation) and the Doppler frequency  $f_d$  ("higher frequency" ripples) of the approximation are not really good.

All of the other simulators, instead, are almost indistinguishable from each other and from the ideal case, at least up to a normalized time  $f_d\tau = 10$ . I want to recall that this is obtained with only 8 oscillators for all of these simulators! Note that also the original *Clarke's* simulator is indistinguishable from the ideal case with these few oscillators, and keep in mind that it's only a fourth of the equivalent number of oscillators simulated by the other designs. This is to empirically confirm the statement from [B1] where *Clarke's* simulator is said to converge very quickly to the ideal case. This fact almost seems to suggest that *Jakes's* simulator might not have been even needed in the first place!

One last correlation to consider is  $R_{|X|^2}$ , shown in Fig. 3. Of all of the statistics shown up to now, this is definitely the most delicate one since it refers to powers (which is usually what we are really interested into) and in includes terms up to the fourth moment. Predictably it's the statistic that yields the worst results.

Once again, both *Jakes's* and *PopBeaulieu's* simulators have among the worst statistics. In this case, *ZhengXiao2003* yields quite bad statistics too. *ZhengXiao2002*, *LiHuang* and *XiaoZhengBeaulieu*, instead are definitely an improvement. *Komninakis* here is not bad, but has a strange ondulatory motion and, again, its oscillation seem to be too fast with respect to the ideal case. The

clear winner, this time, is *Clarke's* simulator: even though it falls a bit short around the zero lag (about 1.9 instead of 2), the rest of the correlation is by far the closest of them all, and that's just with 8 sinusoids!

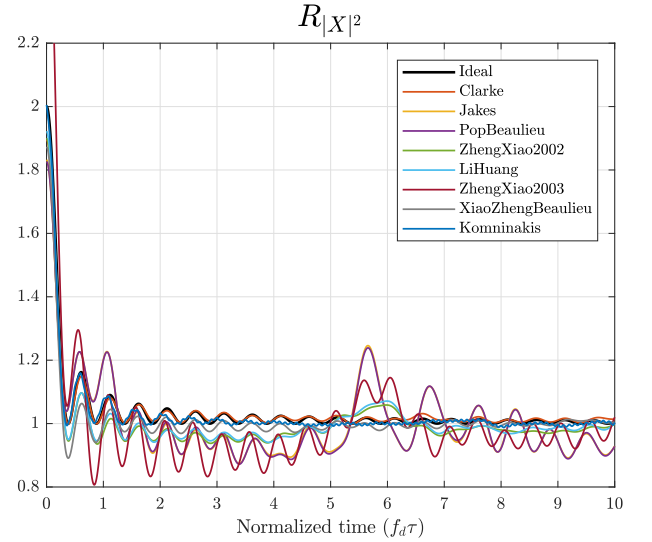


Figure 3: Correlations of the channels

### 3.3 Fading Statistics

The last statistics that I'm going to present here were not present in all of the reference papers, but i think they are really important if not crucial (for example in channel coding design). A short explanation and ideal formulas were given in Sec. 2.2, Eqs. 5,6.

Starting from *LCR*, it is possible to see from Fig. 4 how different values for  $f_dT$  (0.05, 0.01 and 0.002) yield very different results. In particular for low thresholds, the higher  $f_dT$  the earlier the deviation from the ideal formula begins. This can be seen in exactly the same way for all of the proposed simulators, hence

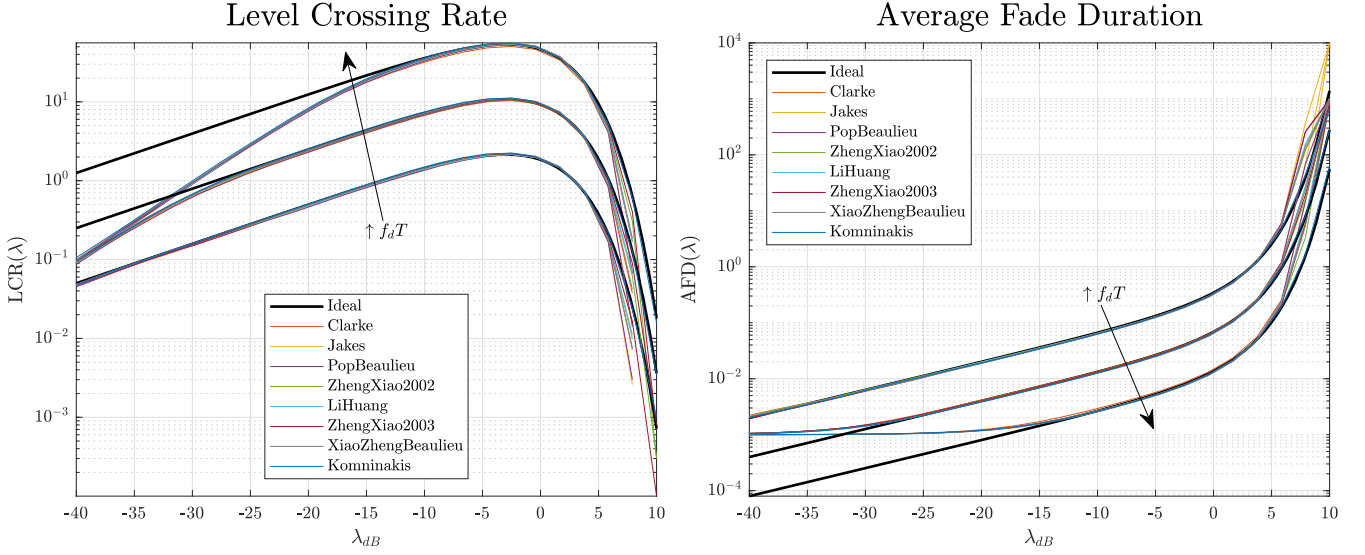


Figure 4: Plots for threshold-based statistics. Note that  $\lambda_{dB} \triangleq 20 \log_{10} \lambda$  since it's envelope related, not power related

it's probably caused by something more fundamental. Reference [A3] attributes this phenomenon to the fact that for fast fading (or low sampling rate, i.e. a high value of  $f_d T$ ), some very narrow troughs cannot be correctly sampled, thus giving an underestimate of the level crossing rate. As the fading slows down (or sampling rate increases, i.e.  $f_d T$  decreases) more and more fine troughs can be correctly sampled. This, of course, only postpones the problem, but if the critical threshold is low enough for your specific purpose, the model works just fine. For high thresholds, instead, there are some major differences among the simulators: here *Komninakis* is definitely the best one being almost identical to the ideal curve up to 10 dB. *ZhengXiao2003* and *ZhengXiao2002* are respectively the second and third best ones, but usually reaching the 10 dB threshold about an order of magnitude lower. All the other simulators do not even register any crossing above 8 dB. From 5 dB and below, instead, they all perform perfectly. Note that we are usually interested in low thresholds, since they are the ones responsible for errors, so the low precision towards higher levels may be considered on secondary importance.

Similar considerations can be done for the *AFD*: an increasing  $f_d T$  yields worst performance towards lower thresholds due to resolution problems: in this case, a constant sampling period  $T = 1ms$  was chosen, varying the Doppler spread as for the *LCR* test. Note that the fade duration obviously cannot go below  $1ms$ , and higher/lower values of  $f_d T$  make this problem arise sooner/later. Once again, *Komninakis* tend to perform almost perfectly while all the others overshoot even considerably the ideal statistics. But still, we are usually interested for how long the channel will return erroneous bits, meaning for how long it will stay below a (low) threshold. So, as for the *LCR*, also for the *AFD* high levels may be considered not important and, again, it should be checked whether the product  $f_d T$  is suitable for the particular problem.

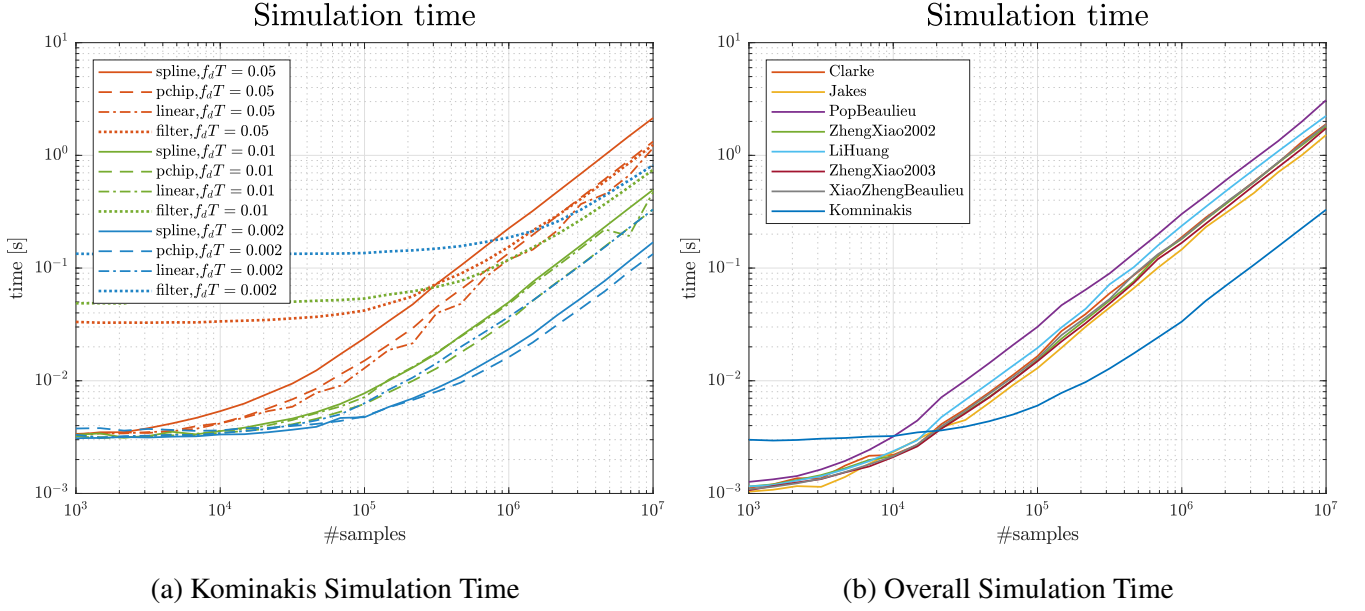


Figure 5: Simulation time for current implementations

### 3.4 Performance Evaluation

One final test that I decided to do is a performance evaluation. This is quite delicate, though: a fine optimization should be carefully done for each one of the 8 simulators implemented in order to obtain meaningful results. Also, the results are machine and probably even CPU architecture dependent. In order to squeeze every last bit of performance, Matlab's proprietary profiler has been exploited over and over at the best of my capabilities for every simulator. For what concerns the architecture, Matlab R2017a has been used, running over a 64-bit Windows 10 machine equipped with an Intel Core i5-2500K (overclocked @4.11 GHz) and 8 GB of RAM. Each test has been carefully designed in order to maximize the memory usage without exceeding it (which would have artificially hurt performance). Finally, timings were taken through Matlab using the `tic-toc` instructions around a call to the function `createChannel`. Multiple simulations are then performed (at least 10) up until the 95% *Confidence Interval* falls below a 10% error from the mean.

For the first test, I had to decide which type of interpolation for *Komninakis'* simulator and  $f_dT$  product to use for all of the previous tests. In Fig. 5a how different  $f_dT$  affect performance over a broad number of simulated samples. I want to recall that *Komninakis'* filter is set to work at a fixed  $f_dT = 0.1$ , hence

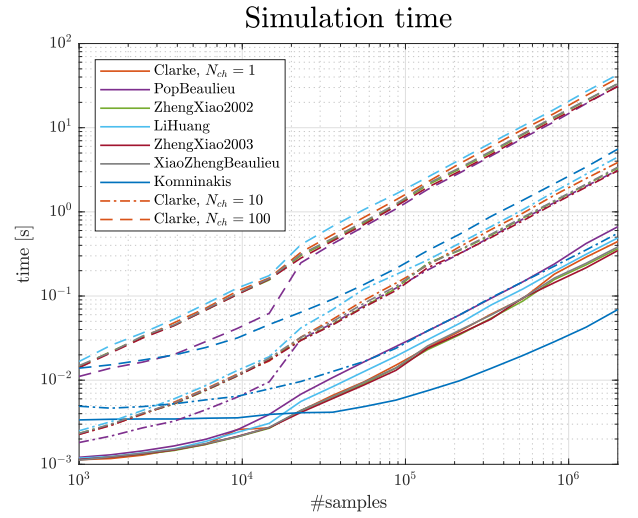


Figure 6: Simulation time with different number of channels

for lower values of  $f_d T$  a lower number of complex gaussian random variables will have to be created, a higher interpolation factor will be need and, thus, a narrower lowpass interpolation filter will have to be designed. This is perfectly reflected in the figure: for the filter interpolation, a much higher initial cost is needed in order to design the lowpass filter, and this cost raises the narrower the filter. For the polynomial interpolation, instead, no such cost is required thus yielding better performance when less random variables are needed (polynomial interpolation is low cost). Furthermore, the *spline* interpolation, as expected, is the most expansive one, while *pchip* is a bit more efficient. I expected *linear* interpolation to be by far the most efficient one but apparently it is not so. In any case, statistics for this simple interpolation were quite bad, whereas *spline* and *pchip* were comparable. For the high initial cost, then, I didn't even considered the *filter* method but it may be a possibility for a very high number of samples. For these reasons in the end I decided to use *pchip* as the main interpolation method.

Looking, now, at Fig. 5b, the results for a single channel simulation, we clearly see two things at a glance: the *Komninakis* simulator starts higher than the others but quickly surpasses them all, while they stay quite close to each other. The former is partially expected since it requires a much higher number of complex gaussian random variables while apparently *pchip* interpolation is much faster than computing trigonometric functions, justifying the title of [A3]. The latter is also quite expected, but maybe not this much close together. A few things to notice: *Jakes'* simulator, fortunately, is the fastest one but *PopBeaulieu* is by far the slowest one. Note also that *Clarke's* simulator doesn't perform so bad compared to the others.

One last curiosity that I had was about how simulators would behave with a higher number of independent channels. Due to the deterministic nature of *Jakes'* simulator, I did not include it in this comparison since it would have simply made a repetition of the first channel. I was able, though, to keep the same colors for all the other simulators in order to have a better and unified visual presentation (Fig. 6). The strangest result is certainly given by *PopBeaulieu*: with only one channel it performs the worst while for 10 and 100 is actually the best one among the *SOS* type of simulators, even with exceptional results up to  $15 \cdot 10^3$  samples, then suddenly it aligns with the rest of the group. Here it can be fully appreciated the computational efficiency of *Komninakis'* simulator: it can improve up to a full order of magnitude the simulation time!

## 4 Conclusions

In the end, 8 different Rayleigh channel simulators have been implemented and both their statistical and computational performance have been tested. Excluding *Jakes'* and *PopBeaulieu's* simulators, all the others have very similar performance, and usually very close to the ideal case. Different simulators

shined for particular reasons: while *Komninakis* has outstanding performance for what concerns both computational complexity and fading statistics, it also has quite poor correlation statistics; on the other hand *Clarke*'s original model actually yields very good performance all around without resulting as slow as it may have been thought. One thing to say, though, is that while for all the other *SOS* simulators further optimization could (possibly) be done, it would be much more difficult for the latter since no symmetries whatsoever are present.

Thanks to this project I had the possibility to try out many new MATLAB's programming techniques that I recently learned from a few Mathwork's courses. Now my code performs better, is more robust, and also well commented, documented and organized. Furthermore, I now have many simulators ready to use if ever needed and a full performance characterization for all of them.

## References

- [A1] M. F. Pop and N. C. Beaulieu. "Limitations of Sum-of-Sinusoids Fading Channel Simulators". In: *IEEE Transactions on Communications* 49.4 (Apr. 2001).
- [A2] Y. R. Zheng and C. Xiao. "Simulation Models With Correct Statistical Properties for Rayleigh Fading Channels". In: *IEEE Transactions on Communications* 51.6 (June 2003).
- [A3] C. Komninakis. "A Fast and Accurate Rayleigh Fading Simulator". In: *IEEE GLOBECOM* (Dec. 2003).
- [B1] C. Xiao, Y. R. Zheng, and N. C. Beaulieu. "Novel Sum-of-Sinusoids Simulation Models for Rayleigh and Rician Fading Channels". In: *IEEE Transactions on Wireless Communications* 5.12 (Dec. 2006).
- [C1] Y. Li and X. Huang. "The Simulation of Independent Rayleigh Faders". In: *IEEE Transactions on Communications* 50.7 (Sept. 2002).
- [C2] Y. R. Zheng and C. Xiao. "Improved Models for the Generation of Multiple Uncorrelated Rayleigh Fading Waveforms". In: *IEEE Communications Letters* 6.6 (June 2002).
- [CB02] G. Cherubini and N. Benvenuto. *Algorithms for Communications Systems and their Applications*. John Wiley & Sons Inc., 2002. ISBN: 0-470-84389-6.
- [Cla68] R. H. Clarke. "A statistical theory of mobile-radio reception". In: *Bell Syst. Tech J.* (July 1968), pp. 957–1000.
- [Jak74] W. C. Jakes. *Microwave Mobile Communications*. Reissued in 1994. Piscataway, NJ: IEEE Press, 1974.