Fitting a time series to the Rayleigh plus lognormal (Suzuki) distribution

(v.1 January 22, 2023)

Fernando Pérez Fontán, Vicente Pastoriza Santos and Fernando Machado Domínguez
University of Vigo. Spain

fpfontan@uvigo.es, vpastoriza@uvigo.es, fmachado@uvigo.es

https://github.com/RadioPropagationChannel

Contents

1. Introduction 1
2. The Gaussian/normal distribution 2
3. Analyzing a time-series with shadowing and multipath 3
4. References 9
5. Software Supplied 9

1. Introduction

In this fascicle, we study how we could go ahead and analyze a series representing the measurement of an RF continuous, unmodulated wave, CW, where the distance from the transmitter, e.g. a macrocell BS, does not play a significant role. We assume that the path loss is constant and that we only observe slow variations due to shadowing/blockage and fast variations due to multipath.

For traveled distances of a few tens of wavelength, we can model the received signal at a mobile (if the direct signal is totally blocked) by means of a Rayleigh distribution. For longer traveled distances, the Rayleigh parameter (e.g. its rms value) shows some variability and thus, we can no longer use a single distribution but a combination of distributions, typically Rayleigh plus lognormal, the so called Suzuki distribution [1].

We reproduce how we would process the series and statistically analyze it. The data was generated with a simulator (discussed elsewhere on this site) that should perfectly follow a Rayleigh plus lognormal distribution. On top of that, there is no noise added, which is equivalent to having a very high Signal to Noise Ratio, SNR.

Here what we will be doing is separating out the fast and slow variations and model them separately. The Rayleigh distribution has already been studied and is presented somewhere on this site. As for the lognormal distribution, we are going to study it somewhere else when we discuss the Suzuki fading generator.

However, associated with the lognormal distribution is the normal or Gaussian distribution, thus, if we have a random variable x following a lognormal distribution, then variable $X = \ln x$ is normally distributed. In general, the normal distribution is much easier to handle and the parameters of the lognormal distribution are not given directly, in fact, the parameters of the associated normal distribution, mean and standard deviation, are provided instead.

The most common relation between the normal and lognormal distribution is when X is in Neper and x may be a voltage or a power. Other relationships are as follows,

$$X = \ln x \qquad P(\text{dB}) = 10 \log p \qquad V(\text{dB}) = 20 \log v \tag{1}$$

note that there is a simple conversion between Neper and dB, i.e., $\ln(x) = 10 \log x / 10 \log \text{e}$ and $\ln(x) = 20 \log x / 20 \log \text{e}$.

This introduction is to stress the fact that the slow variations both of the local average power or its associated voltage value will be analyzed in logarithmic units and thus we will be using the normal distribution, not the actual lognormal.

2. The Gaussian/normal distribution

This distribution describes continuous variables with positive values, negative or both [2]. When many random effects are added together they usually give rise to a Gaussian distribution. This distribution is often used, for example, in the variations in the received signal in mobile communications due to shadowing effects or, in tropospheric propagation under clear air conditions, for describing the fluctuations of a quantity around its mean value (scintillation). Of course, it is also used to describe the thermal noise in any communications system.

The Gaussian/normal pdf is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right] \tag{2}$$

where m and σ are the distribution's mean and standard deviation, respectively. Its cumulative distribution is

$$F(X) = \Pr(x < X) = \int_{-\infty}^{X} f(x)dx = 1 - Q\left(\frac{X-m}{\sigma}\right) \tag{3}$$

where Q is a function representing the right tail of the Gaussian distribution, see below.

It is helpful to normalize the Gaussian random variable x using its mean, m, and standard deviation, σ. The new, normalized Gaussian random variable, k, is defined as $k = (x - m)/\sigma$, where its mean is zero and its standard deviation equal to one.

The Gaussian pdf tail function is given by

$$Q(k) = \frac{1}{\sqrt{2\pi}} \int_{k}^{\infty} \exp\left(-\frac{\lambda^2}{2}\right) d\lambda \tag{4}$$

Function $Q(k)$ provides an easy way of calculating the probability that random variable x fulfills that $x > m + k\sigma$. This is equivalent to calculating the area under the tail of the pdf (Figure 1). Function Q is easily related to the error function or its complementary also available in Matlab (functions **`qfunc`**, **`erf`** and **`erfc`**) through

$$Q(k) = \frac{1}{2} - \frac{1}{2}\text{erf}\left(\frac{k}{\sqrt{2}}\right) = \frac{1}{2}\text{erfc}\left(\frac{k}{\sqrt{2}}\right) \tag{5}$$

Table 1 provides some practical values of the normalized Gaussian complementary CDF ($1 - F$).

Table 1. Relevant values of $Q(k)$

k	$Q(k) = 1 - F(k)$	k	$Q(k) = 1 - F(k)$
0	0.5	1.282	10^{-1}
1	0.1587	2.326	10^{-2}
2	0.02275	3.090	10^{-3}
3	1.350×10^{-3}	3.719	10^{-4}
4	3.167×10^{-5}	4.265	10^{-5}
5	2.867×10^{-7}	4.753	10^{-6}
6	9.866×10^{-10}	5.199	10^{-7}
		5.612	10^{-8}

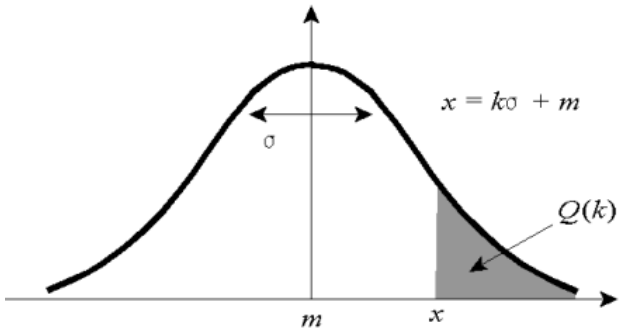

Figure 1 Function Q, area under the tail of the Gaussian distribution

3. Analyzing a time-series with shadowing and multipath

The following example is implemented in script **`fitSuzuki`**. The name comes from the fact that we are going to analyze a time series where there are fast fading effects (Rayleigh distributed) superposed on slow fading (lognormally distributed). Instead of studying the lognormal variations of the power or voltage in linear units, we will study their associated variations in dB which are normally distributed.

File **`SuzukiSeries1.mat`** (Figure 2) is provided which includes a simulated continuous wave (CW) signal measurement with carrier frequency, $f_0 = 2.0$ GHz. This file could be representative of a sector of a circular route at a constant distance from the transmitter so that the **nominal distance-dependent received power**, P_R, remains constant.

File **`SuzukiSeries1`** contains two column vectors, first the traveled distance in meters, **`dist_axis`**, and the second, the received signal in dBm, **`PdBm`**. Additionally, further details are also provided, that is,

```
V = 10;  % m/s, terminal speed
ds = 0.0150; % m, sample spacing
fGHz = 2;  % GHz, RF frequency
fs = 666.6667;  % Hz, sampling frequency
ts = 0.0015;  % s, sample interval
```

Note that here, the signal is sampled in the traveled distance domain.

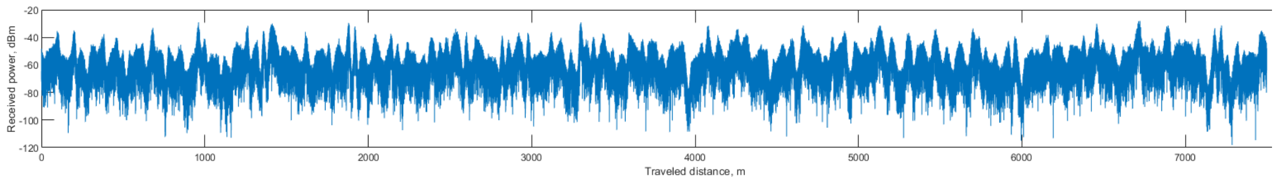

Figure 2 Plot of **`SuzukiSeries1`** where the abscissa is in length units and the ordinate shows the received power in dBm.

From a visual observation, it is clear that both, slow and fast variations, are present. **Separation** of these two effects must be performed by using a low- and high-pass filter. We have used a running mean filter for this purpose. This is equivalent to filtering the series with a rectangular window that is slid through the series.

It is left for the reader to use another filter for this purpose, for example a Hanning window (Matlab function **`hann`**). This has some advantages over the rectangular window used here.

Before we go on to discussing the filtering, we jump over to another script, **`shadowingPlusMultipathSpectrum`**, where we try and get a clearer picture of the variations rates involved in the provided data. We first analyze the spectrum of the series in Hz (or cycles per second) and then we picture it in spatial frequencies, that is in cycles per meter (or per wavelength).

The highlights of the code are as follows,

```
SpW = abs(fft(pW,NFFT)).^2;
freq_axis = (1:NFFT)*fs/NFFT;  % axis in Hz
```

where we computed the power spectrum (Hz) of the series in W using NFFT points. The maximum power in the multipath part (envelope) is twice the maximum Doppler frequency, $f_m = V/\lambda$ (we discuss this somewhere else in this site). We can see that there is a DC component, the average, a slow variations spectrum up to about 1 Hz and the fast variations spectrum up to $2f_m$ (red line). After that, we may see the rectangular truncation effect and, if it was present, the noise.

We are interested here in the distance domain, the spatial frequency is calculated as follows,

```
space_freq_axis = freq_axis/V;
```

We can pay some attention to the approximate limit of the slow variations spectrum which is somewhere around 0.5 Hz (seconds per cycle) or, equivalently, 0.05 C/m, that is, reversing it, 20 meters/cycle. This number should give us some idea on the order of magnitude of the filter we need to use.

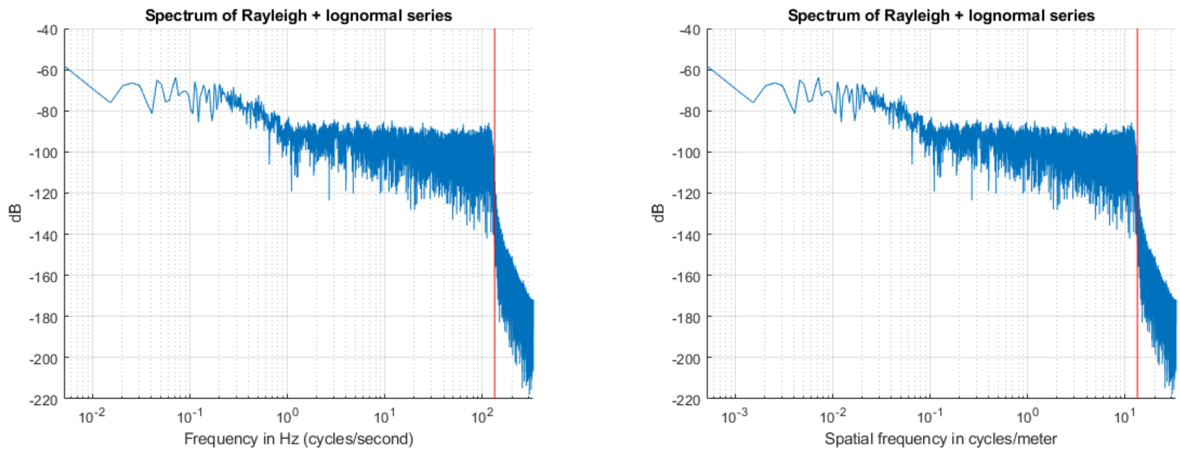

Figure 3 Spectrum of **`SuzukiSeries1`** in Hz and cycles/m. Red line indicates maximum Doppler spectrum of a Rayleigh series envelope.

Coming back to script **`fitSuzuki`**, Matlab function **`conv`** (convolution) can be used for carrying out the filtering. This process gives unreliable samples at the beginning and ending of the filtered series, which have to be discarded. We know that the convolution, **`C=conv(A,B)`**, produces a resulting vector of length

`MAX([LENGTH(A)+LENGTH(B)-1,LENGTH(A),LENGTH(B)]).`

If we use the function's parameter **`SHAPE`** in the convolution function, i.e., **`C=conv(A,B,SHAPE)`**, where the value of **`SHAPE`** is **`'same'`**, it returns the central part of the convolution that is of the same size as **`A`**. This solves our problem and we no longer need to clip off the unreliable samples at the beginning and ending of the convolution. Another alternative would be using function **filter**. We will be using this function elsewhere on this site.

The impulse response of the running mean or sliding window, W, used here as follows,

$$W[n] = h[n] = \frac{1}{N}\{\delta[n] + \delta[n-1] + \cdots + \delta[n-N]\}, \tag{6}$$

that is, a collection of N Dirac deltas with amplitude $1/N$, where N is the window length. Note that this is a FIR (Finite Impulse Response) filter. The frequency response of a rectangular window is a sinc function. This has some disadvantages given its high sidelobes which may lead to misestimating the Gaussian distribution parameters.

Since we are using a rectangular window as filter and its Fourier transform is a sinc function, it is good to remember some definitions. Thus, a **sinc function** is given by the expression

$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{7}$$

which is equal to unity at $x = 0$, and becomes zero at multiples of π, i.e., $x = \pm n\pi$ where n is an integer.

Another function of interest if the **rectangular function**, $\prod(f/B)$, equal to one from $-B/2$ to $B/2$, and zero elsewhere.

The following Fourier transform pairs are of interest,

$$\prod\left(\frac{t}{\tau}\right) \leftarrow F \rightarrow \tau\,\mathrm{sinc}(\tau f) \tag{8}$$

Note that $\mathrm{sinc}(\tau f)$ becomes zero at frequency multiples of $1/\tau$ except for $f = 0$. The other Fourier transform pair is

$$\mathrm{sinc}(Bt) \leftarrow F \rightarrow \frac{1}{B}\prod\left(\frac{f}{B}\right)$$

(9)

where $\mathrm{sinc}(Bt)$ becomes zero at time multiples of $1/B$ except for $t = 0$.

If we apply the first transform to our case, we can determine the length of our rectangular window that achieves the wanted null-to-null bandwidth. Thus,

$$BW = \frac{2}{\tau} \quad \longrightarrow \quad \tau = \frac{2}{BW} \tag{10}$$

Since $BW = 2 \times 0.5\ \mathrm{Hz}$, to account for the positive and negative parts of the spectrum, it means that the duration of the rectangular window should be $\tau = 2/(2 \times 0.5) = 2$ s, which is number of samples is equivalent to $\tau/t_s = 2/0.0015 = 1333$ samples. Also, 2 s duration is equivalent to a distance of $2 \times 10 = 20\ \mathrm{m}$.

Whit the above in mind, we proceed with a brief description of Script **SuzukiSeries** that loads the file, variable assignments and perform various transformations from logarithmic units to linear,

```
load SuzukiSeries1

d = dist_axis;
P = PdBm;
p = 10.^(P/10);                 % power in mW
```

A **window** size of 20 meters was estimated and is going to be used for separating out the fast and slow variations.

```
AveragingWindow = 20.0 ;       % Set the averaging window length in meters

windowLength = round(AveragingWindow/ds)
W = ones(windowLength,1);   % Create running mean window
W = W/windowLength;         % Normalize window
```

To separate the fast and slow variations, we need to convert the original series in logarithmic units to linear units and apply the running mean just discussed. Assuming that the received power series in linear units results from the product of two components, we get

$$p(x) = m(x) \cdot p_0(x) \qquad [\mathrm{W\ or\ mW}] \tag{11}$$

These two series vary at different rates. The first series represents the local mean power around route point x, while the second represents the superposed fast variations with normalized unit power.

The procedure followed to is to calculate the local mean of $p(x)$, $m(x)$, through low pass filtering, this is carry out using

```
pfilt = conv(p,W,'same');
```

this variable should contain the variations due exclusively to shadowing in linear (power) units. From this result we can extract the estimated parameters for the Gaussian distribution. First we convert back to dB and calculate the mean and standard deviation,

```
Pfilt = 10*log10(pfilt); % we now go back to dBm

MM = mean(Pfilt);
SS = std(Pfilt);
```

Once the estimated parameters are available, we can compare the theoretical and experimental distributions using

```
[~, ~, CDFx,CDFy, stepCDF] = fpdfCDFbins(Pfilt, 30);
[pdfX, pdfY, ~, ~, steppdf]= fpdfCDFbins(Pfilt, 20);
```

The number of bins for the CDF should be large while for the pdf small. Here, for the sake of plot readability we only chose 30 bins for the CDF. A lager number should produce a closer match.

As indicated in the fascicle on Rayleigh distribution fitting, we need to adapt the theoretical pdf to be able to compare it with the experimental pdf,

```
Paxis = (min(Pfilt):max(Pfilt));
pdf = 1/(sqrt(2*pi)*SS)*exp(-0.5*((Paxis-MM)/SS).^2);
fhist = pdf*steppdf;
```

Our comparisons will be purely qualitative through visual inspection. In the same fascicle mentioned we show how to perform a chi-square goodness-of-fit test to quantitatively verify the match.

To compute the theoretical distribution, we can use

```
F = 1-qfunc((Paxis-MM)/SS);
```

To check the correlation length of the extracted slow variations, we can calculate the autocorrelation function as follows,

```
rho_Pfilt = xcov(Pfilt,'coeff');
rho_axis = (1:length(Pfilt)*2-1) - length(Pfilt); % axis in samples
rho_axis = rho_axis*ds;          % axis in meters
```

and estimate the correlation length with respect to a threshold,

```
corrThresh = exp(-1); % 0.1  % correlation threshold, tipically 1/e
maxPoint = find(rho_Pfilt == 1);
II = find(rho_Pfilt(maxPoint:end) < corrThresh);
corrLength = rho_axis(II(1)+maxPoint);
```

Finally, in order to separate out the fast variations, we could implement a high-pass filter. Alternatively, we can operate in logarithmic units and subtract from the overall (instantaneous) series the newly extracted local mean value, i.e.,

$$P_0(x) = P(x) - M(x) \tag{12}$$

and, finally, convert the result back to linear, that is,

```
P0 = P - Pfilt;           % normalized Rayleigh series in dB units

p0 = 10.^(P0/10);
mean_p0 = mean(p0)
p0norm = p0/mean_p0;
vnorm = sqrt(p0norm);    % generate vnorm
```

Variable **vnorm** should contain the isolated fast variations, that are superposed on a constant local level.

One last step to facilitate the analysis of the isolated fast variations, we can export them to a file to be processed for example with **fitRayleigh** as discussed in another fascicle,

```
time_axis = d/V;
PdBm = P0;

save reminderSeries time_axis PdBm
```

In Figure 4-Figure 7 we present some of the obtained results. As for numerical values we get the parameters of the Gaussian distributed slow variations: mean value -**55.6467** (dBm), standard deviation **6.8806** (dBm) and correlation length **20** m.

From the comparison between the measured pdf and CDF and the theoretical ones shown in Figure 5 we can see that the match is fairly good. We would need to quantify this via a goodness-of-fit test, left for the reader.

In spite of the reasonably good visual fit, if we look at the results with some attention by zooming in on the series, for example, it can be shown that the separation of the fast and the slow variations achieved is far from perfect. This has led to the fast variations leaking into the slow variations and vice versa. The not so perfect analysis may be due to the window size used or to the filter. Maybe with a Hanning window filter we would be getting better results. It is left to the reader to perform a simple trial and error iteration changing one of the analysis parameters at a time.

An example of the not so good results can be observed if we compute the mean for sections of the overall fast variations instantaneous power series, **p0**. In principle the overall mean should be equal to 1 but also shorter sections of the series should yield the same value, indicating that there is no trace of the longer term variations.

It would be good to verify whether the isolated slow and fast variations follow their corresponding distribution by performing a chi-square test.

Finally, another series is provided for analysis in file **SuzukiSeries2** which has different parameters to the one just analyzed. It is left to the reader to analyze this new series.

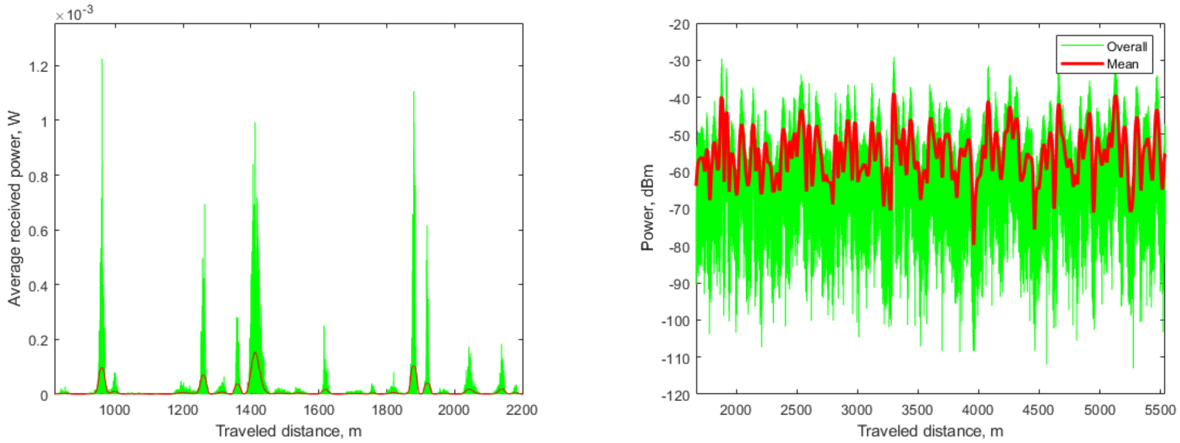

Figure 4 left. Zoomed in section of instantaneous power series and local means in linear units. Right. Zoomed in section of Instantaneous power series and local means in dBm.

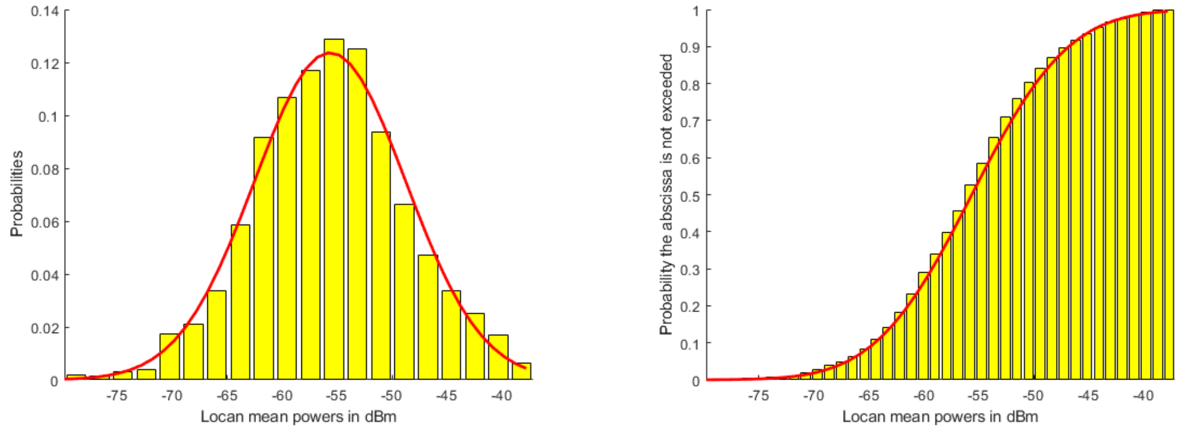

Figure 5 left. Measured pdf (histogram) and theoretical Gaussian pdf. Mean **MM = -55.6467** (dBm) and standard deviation **SS = 6.8806** (dBm). Right. Measured CDF (histogram) and theoretical Gaussian CDF.

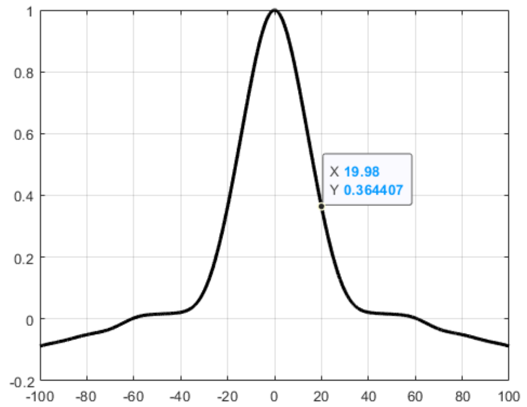

Figure 6 Autocorrelation function of `Pfilt`. The correlation length at 1/e resulted in about 20 m.

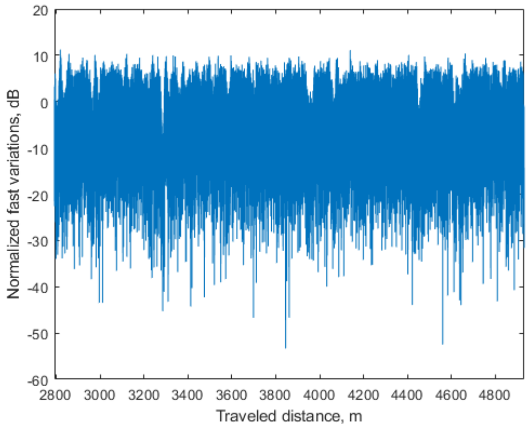

Figure 7 Fast signal variations separated out from the overall variations. The value of its average power is normalized to unity. Zoom in on the first 50 m of terminal's route.

4. References

[1] H.Suzuki. A Statistical Model for Urban Radio Propagation. IEEE Transactions on Communications, Vol. Com-25, No. 7, July 1977, Pp. 673-680

[2] Rec. ITU-R P.1057-6. Probability distributions relevant to radiowave propagation modelling. 2019. International Telecommunication Union. Radiocommunication Sector.

5. Software Supplied

In this section, we provide a list of Matlab functions and scripts implementing the examples mentioned.

FUNCTIONS
fpdfCDFbins

-

SCRIPTS
fitSuzuki
shadowingPlusMultipathSpectrum

Additionally, the following time series are supplied:

SERIES
SuzukiSeries1
SuzukiSeries2