Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Simulating the SUI Channel Models**
Date Submitted	**2001-04-11**
Source(s)	Daniel S. Baum Information Systems Laboratory 272 David Packard Building 350 Serra Mall, Stanford University Stanford, CA 94305-9510, USA Voice: +1-650-724-3640 Fax: +1-650-723-8473 mailto: dsbaum@stanford.edu or daniel.baum@ieee.org
Re:	Call for Contributions: Session #10 Topic: Traffic, Deployment and Channel Models, dated September 15, 2000 (IEEE 802.16.3-00/13) This responds to the second item: Channel propagation model
Abstract	This document is an addendum to the channel models for fixed wireless applications. It is intended to help in the implementation of these models into a software channel simulator by elaborating on some issues that were not explained in detail and by providing code that can act as a core for more extensive simulations.
Purpose	This is for use by the Task Group to implement channel propagation models in software.
Notice	
Release	
Patent Policy and Procedures	

Simulating the SUI Channel Models

Daniel S. Baum
Stanford University

The SUI Channel Models

This submission is an addendum to the contribution [EKSB01], which presents Channel Models for Fixed Wireless Applications and which contains the definition of a set of 6 specific channel implementations known as SUI channels. This paper is intended to help in the implementation of these models into a software channel simulator by elaborating on some issues that were not explained in detail and by providing Matlab[1] code that can act as a core for more extensive simulations.

Definitions

The parameters of the 6 SUI channels, including the propagation scenario that led to this specific set, are presented in the referenced document. As an example, the definition of the SUI-3 channel is reproduced below:

SUI – 3 Channel				
	Tap 1	**Tap 2**	**Tap 3**	**Units**
Delay	0	0.5	1	µs
Power (omni ant.)	0	-5	-10	dB
K Factor (omni ant.)	1	0	0	
Power (30^o ant.)	0	-11	-22	dB
K Factor (30^o ant.)	3	0	0	
Doppler	0.4	0.4	0.4	Hz
Antenna Correlation:	$\rho_{ENV} = 0.4$			
Gain Reduction Factor:	GRF = 3 dB			
Normalization Factor:	F_{omni} = -1.5113 dB, $F_{30°}$ = -0.3573 dB			

Fig 1.) SUI-3 channel model definition

The set of SUI channel models specify statistical parameters of microscopic effects (tapped delay line, fading, antenna directivity). To complete the channel model, these statistics have to be combined with macroscopic channel effects such as path loss and shadowing (also known as excess path loss) which are common to all 6 models in the set.

Each set model also defines an antenna correlation, which is discussed in more detail later in this document. The gain reduction factor (GRF) has also been included in the tables to indicate the connection with the K-factor.

[1] MATLAB is a registered trademark of The Math Works, Inc.

Software Simulation

The aim of a software simulation can vary greatly, the range spans from statistical analysis to communication signal simulations. The approach to coding for a task can therefore differ significantly depending on the goal. For the purpose of demonstrating the model implementation, we concentrate on the core channel simulation, i.e. the result of our code is to produce channel coefficients at an arbitrary (channel) sampling rate.

Power Distribution

We use the method of filtered noise to generate channel coefficients with the specified distribution and spectral power density. For each tap a set of complex zero-mean Gaussian distributed numbers is generated with a variance of 0.5 for the real and imaginary part, so that the total average power of this distribution is 1. This yields a normalized Rayleigh distribution (equivalent to Rice with K=0) for the magnitude of the complex coefficients. If a Ricean distribution (K>0 implied) is needed, a constant path component m has to be added to the Rayleigh set of coefficients. The ratio of powers between this constant part and the Rayleigh (variable) part is specified by the K-factor. For this general case, we show how to distribute the power correctly by first stating the total power P of each tap:

$$P = |m|^2 + \sigma^2, \tag{1}$$

where m is the complex constant and σ^2 the variance of the complex Gaussian set. Second, the ratio of powers is

$$K = \frac{|m|^2}{\sigma^2}. \tag{2}$$

From these equations, we can find the power of the complex Gaussian and the power of the constant part as

$$\sigma^2 = P\frac{1}{K+1} \text{ and } |m|^2 = P\frac{K}{K+1}. \tag{3a/b}$$

From eqns. (3a/b) we can see that for K=0 the variance becomes P and the constant part power diminishes, as expected. Note that we choose a phase angle of 0° for m in the implementation.

Doppler Spectrum

The random components of the coefficients generated in the previous paragraph have a white spectrum since they are independent of each other (the autocorrelation function is a Dirac impulse). The SUI channel model defines a specific power spectral density (PSD) function for these scatter component channel coefficients called 'rounded' PSD which is given as

$$S(f) = \begin{cases} 1 - 1.72 f_0^2 + 0.785 f_0^4 & |f_0| \le 1 \\ 0 & |f_0| > 1 \end{cases} \quad \text{where} \quad f_0 = \frac{f}{f_m}. \tag{4}$$

To arrive at a set of channel coefficients with this PSD function, we correlate the original coefficients with a filter which amplitude frequency response is derived from eqn. (4) as

$$|H(f)| = \sqrt{S(f)}. \tag{5}$$

We choose to use a non-recursive filter and the frequency-domain overlap-add method for efficient implementation. We also have to choose some filter length which determines how exact and smooth our transfer function is realized by the filter.

Since there are no frequency components higher than f_m, the channel can be represented with a minimum sampling frequency of 2 f_m, according to the Nyquist theorem. We therefore simply define that our coefficients are sampled at a frequency of 2 f_m.

The total power of the filter has to be normalized to one, so that the total power of the signal is not changed by it. The mean energy of a discrete-time process $x(k)$ is

$$E\left\{|x(k)|^2\right\} = \frac{1}{N}\sum_{k=1}^{N}|x(k)|^2 = \int_{-\pi}^{\pi} S_{xx}(e^{j\Omega})\,d\Omega \text{, where } \Omega = 2\pi f / f_s . \tag{6}$$

We note that the PSD function given in eqn. (4) is not normalized.

Antenna Correlation

The SUI channel models define an antenna correlation, which has to be considered if multiple transmit or receive elements, i.e. multiple channels, are being simulated. Antenna correlation is commonly defined as the envelope correlation coefficient between signals received at two antenna elements. The received baseband signals are modeled as two complex random processes $X(t)$ and $Y(t)$ with an envelope correlation coefficient of

$$\rho_{env} = \left| \frac{E\{(X - E\{X\})(Y - E\{Y\})^*\}}{\sqrt{E\{|X - E\{X\}|^2\}E\{|Y - E\{Y\}|^2\}}} \right| . \tag{7}$$

Note that this is not equal to the correlation coefficient of the envelopes (magnitude) of two signals, a measure that is also used frequently in cases where no complex data is available.

Since the envelope correlation coefficient is independent of the mean, only the random parts of the channel are of interest. In the following we therefore consider the random channel components only, to simplify the notation.

In the general case of frequency selective (delay-spread) propagation, the channel is modeled as a tapped-delay line:

$$g(t,\tau) = \sum_{l=1}^{L} g_l(t)\delta(\tau - \tau_l) \tag{8}$$

where L is the number of taps, $g_l(t)$ are the time-varying tap coefficients and τ_l are the tap delays.

We now calculate the correlation coefficient between two receive signals $r_1(t)$ and $r_2(t)$, which are the result of a normalized, random, white transmitted signal $s(t)$ propagating through two channels with the channel impulse responses $g_1(t,\tau)$ and $g_2(t,\tau)$:

$$g_i(t,\tau) = \sum_{l=1}^{3} g_{il}(t)\delta(\tau - \tau_l)\text{, } i \in [1..2] \tag{9}$$

Note that in the SUI channel models the number of taps is L=3 and that the tap delays τ_l are fixed (independent of i). Assuming that equivalent taps in both channels have equal power:

$$\sigma_{1l}^2 = \sigma_{2l}^2 = \sigma_l^2 ,\ l \in [1..3] \quad (10)$$

and that taps with different delays are uncorrelated within a channel as well as between channels:

$$E\{g_{ik}(t) g_{jl}^*(t)\} = 0 ,\ \forall k \neq l \text{ where } k, l \in [1..3] ;\ i, j \in [1..2] \quad (11)$$

the antenna correlation coefficient becomes:

$$\rho_{env} = \left| \frac{\rho_1 \sigma_1^2 + \rho_2 \sigma_2^2 + \rho_3 \sigma_3^2}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2} \right| \quad (12)$$

where ρ_l are the correlation coefficients between each of the 3 pairs of taps $g_{1l}(t)$ and $g_{2l}(t)$:

$$\rho_l = \frac{E\{g_{1l}(t) g_{2l}^*(t)\}}{\sigma_{1l}^2 \sigma_{2l}^2} ,\ l \in [1..3] \quad (13)$$

From eqn. (12) we can see how the antenna correlation can be related to the individual tap correlations, where σ_l^2 are the individual tap gains. To obtain a simple solution for setting these tap correlations depending on the required antenna correlation, we can additionally demand all tap correlations to be equal. Then eqn. (12) simply states that all tap correlations have to be set to the antenna correlation. For the simulation of the SUI channel we recommend setting all tap correlations equal to the antenna correlation.

To generate a sequence of random state vectors with specified first order statistics (mean vector μ and correlation matrix R), the following transformation can be used:

$$\tilde{V} = R^{1/2} V + \mu , \quad (14)$$

where V is a vector of independent sequences of circularly symmetric complex Gaussian-distributed random numbers[2] with zero mean and unit variance. The correlation matrix R is defined and factored as:

$$R = E\{\tilde{V}\tilde{V}^H\} = \begin{bmatrix} 1 & r_{12} & \cdots \\ r_{12}^* & 1 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = R^{1/2} R^{H/2} = R^{1/2} R^{1/2} \quad (15)^3$$

In our simulation we demonstrate the implementation of antenna correlation by an example of two correlated channels (e.g. a 1 transmit by 2 receive antennas system). First two independent but equally distributed set of channel coefficients are created following the procedures given in the beginning of chapter 2. Then we use eqn. (14) and (15) to correlate the random signals of equivalent taps in these two channels, where V is a vector of size 2 by N in our case (N being the number of generated coefficients per channel). The correlation matrix R has only one correlation coefficient r_{12}, which is set to the specified antenna correlation.

Observation Rate

For some purposes it can be required or useful to have the SUI channel coefficients at an arbitrary chosen observation rate, i.e. the data rate of a communication system. This can be done easily by interpolating the channel data to the specific sampling rate.

[2] A circularly symmetric complex Gaussian random variable is a random variable $Z = X + jY \sim \boldsymbol{CN}(0,\sigma^2)$, where X and Y are i.i.d. $\boldsymbol{N}(0,\sigma^2/2)$.

[3] X^H denotes the conjugate transpose of X.

Matlab Coding

Based on the discussion in the previous chapter, the coding in Matlab is straightforward. In the following we will demonstrate this by an example implementation of the SUI-3 omni antenna channel. Apart from the basic simulation processing we also provide code to generate supplementary results which might be of interest and can help to improve understanding. This additional code is displayed in gray outlined boxes titled 'Additional Info'.

Let us first define the simulation parameters

`N          = 10000;`	number of independent random realizations
`OR         = 4;`	observation rate in Hz
`M          = 256;`	number of taps of the Doppler filter
`Dop_res    = 0.1;`	Doppler resolution of SUI parameter in Hz (used in resampling-process)
`res_accu   = 20;`	accuracy of resampling process

and the SUI channel parameters (SUI-3/omni used here):

`P   = [   0   -5  -10 ];`	power in each tap in dB
`K   = [   1    0    0 ];`	Ricean K-factor in linear scale
`tau = [ 0.0  0.5  1.0 ];`	tap delay in µs
`Dop = [ 0.4  0.4  0.4 ];`	Doppler maximal frequency parameter in Hz
`ant_corr  = 0.4;`	antenna correlation (envelope correlation coefficient)
`Fnorm     = -1.5113;`	gain normalization factor in dB

First we calculate the power in the constant and random components of the Rice distribution for each tap:

```
P       = 10.^(P/10);                             % calculate linear power
s2      = P./(K+1);                               % calculate variance
m2      = P.*(K./(K+1));                          % calculate constant power
m       = sqrt(m2);                               % calculate constant part
```

Additional Info: RMS delay spread

```
rmsdel  = sqrt( sum(P.*(tau.^2))/sum(P) - (sum(P.*tau)/sum(P))^2 );
fprintf('rms delay spread  %6.3f µs\n', rmsdel);
```

Now we can create the Ricean channel coefficients with the specified powers.

```
L       = length(P);                              % number of taps
paths_r = sqrt(1/2)*(randn(L,N) + j*randn(L,N)).*((sqrt(s2))' * ones(1,N));
paths_c = m' * ones(1,N);
```

Before combining the coefficient sets, the white spectrum is shaped according to the Doppler PSD function. Since the frequency-domain filtering function `FFTFILT` expects time-domain filter coefficients, we have to calculate these first. The filter is then normalized in time-domain.

```
for p = 1:L
   D     = Dop(p) / max(Dop) / 2;               % normalize to highest Doppler
   f0    = [0:M*D]/(M*D);                       % frequency vector
   PSD   = 0.785*f0.^4 - 1.72*f0.^2 + 1.0;      % PSD approximation
   filt  = [ PSD(1:end-1) zeros(1,M-2*M*D) PSD(end:-1:2) ]; % S(f)
   filt  = sqrt(filt);                          % from S(f) to |H(f)|
   filt  = ifftshift(ifft(filt));               % get impulse response
   filt  = real(filt);                          % want a real-valued filter
   filt  = filt / sqrt(sum(filt.^2));           % normalize filter
   path  = fftfilt(filt, [ paths_r(p,:) zeros(1,M) ]);
   paths_r(p,:) = path(1+M/2:end-M/2);
end;
paths   = paths_r + paths_c;
```

Now that the fading channel is fully generated, we have to apply the normalization factor and, if applicable, the gain reduction factor

```
paths   = paths * 10^(Fnorm/20);                % multiply all coefficients with F
```

Additional Info: average total tap power

```
Pest = mean(abs(paths).^2, 2);
fprintf('tap mean power level: %0.2f dB\n', 10*log10(Pest));
```

Additional Info: spectral power distribution

```
figure, psd(paths(1,:), 512, max(Dop));
```

In a multichannel scenario, the taps between different channels have to be correlated according to the specified antenna correlation coefficient. Assuming that two random channels have been generated in `paths1 = paths_r1 + paths_c1` and `paths2 = paths_r2 + paths_c2` following the procedures above, we can now apply the correlation

```
rho  = ant_corr;                                % desired correlation is ant_corr
SR  = sqrtm([  1    rho ; ...                   % factored correlation matrix
              rho'   1  ]);
V  = zeros(L,2,N);
V(:,1,:) = paths_r1;                            % combine paths
V(:,2,:) = paths_r2;
for l = 1:L
   V(l,:,:) = SR * squeeze(V(l,:,:));           % transform each pair of taps
end;
paths_r1 = squeeze(V(:,1,:));                   % split paths
paths_r2 = squeeze(V(:,2,:));

paths1 = paths_r1 + paths_c1;                   % add mean/constant part
paths2 = paths_r2 + paths_c2;
```

Additional Info: estimate envelope correlation coefficient

```
disp('estimated envelope correlation coefficients between all taps in both paths');
disp('matrix rows/columns: path1: tap1, tap2, tap3, path2: tap1, tap2, tap3');
abs(corrcoef([ paths1; paths2]'))
```

Finally, we interpolate the current rate to the specified observation rate. In order to use the Matlab polyphase implementation `resample`, we need the resampling factor F specified as a fraction $F = P/Q$.

```
SR = max(Dop)*2;                                    % implicit sample rate
m  = lcm(SR/Dop_res, OR/Dop_res);
P  = m/SR*Dop_res;                                  % find nominator
Q  = m/OR*Dop_res;                                  % find denominator
paths_OR = zeros(L,ceil(N*P/Q));                    % create new array
for p=1:L
   paths_OR(p,:) = resample(paths(p,:), P, Q, res_accu);
end;
```

The resampled set of channel coefficients for all the 3 taps are now contained in the matrix `paths_OR`. The total simulated observation period of the channel is now $SR \cdot N = OR \cdot \mathrm{ceil}(N \cdot P/Q)$, where $SR = 2 \cdot \max(Dop)$.

Some Exemplary Results

Now that we have implemented a core simulator for the SUI channels, we will use it to generate some plots about the statistical characteristics of these channels. All simulations use the settings as defined above (SUI-3/omni), if not otherwise stated.

To get a first impression, fig. 2 shows a signal magnitude plot of the channel coefficients over time.

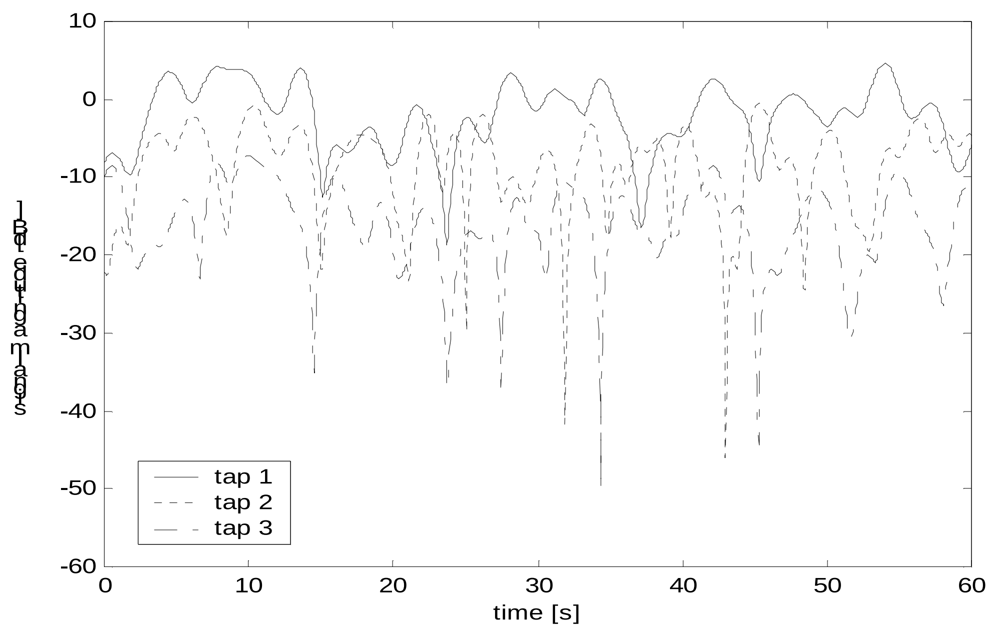

Fig 2.) Fading Plot (OR=20Hz)

For statistical purposes, the observation rate is of no significance to the result as long as it is higher than the original sampling rate. However, increasing the observation rate can sometimes help to smoothen distribution functions.

Fading Distributions

The distribution of power levels at the output of a channel is one of the important properties of a channel. Depending on the assumed input signal spectral power distribution, we can define 3 basic cases for the ‘channel power’:

narrowband signal: PSD: $S(f) = \delta(f_C)$

flat bandpass signal: PSD: $S(f) = 1/BW \cdot rect_{BW}(f - f_C)$

flat wideband signal: PSD: $S(f) = \lim_{BW \to \infty} \left[1/BW \cdot rect_{BW}(f - f_C)\right]$

The higher the bandwidth of a channel, the less likely a deep fade occurs for the total channel power (frequency diversity effect). Fig. 3 illustrates this effect by showing the probability of the signal magnitude falling below a certain level. Note that channel power distributions are generally not Ricean, except for the narrowband case.

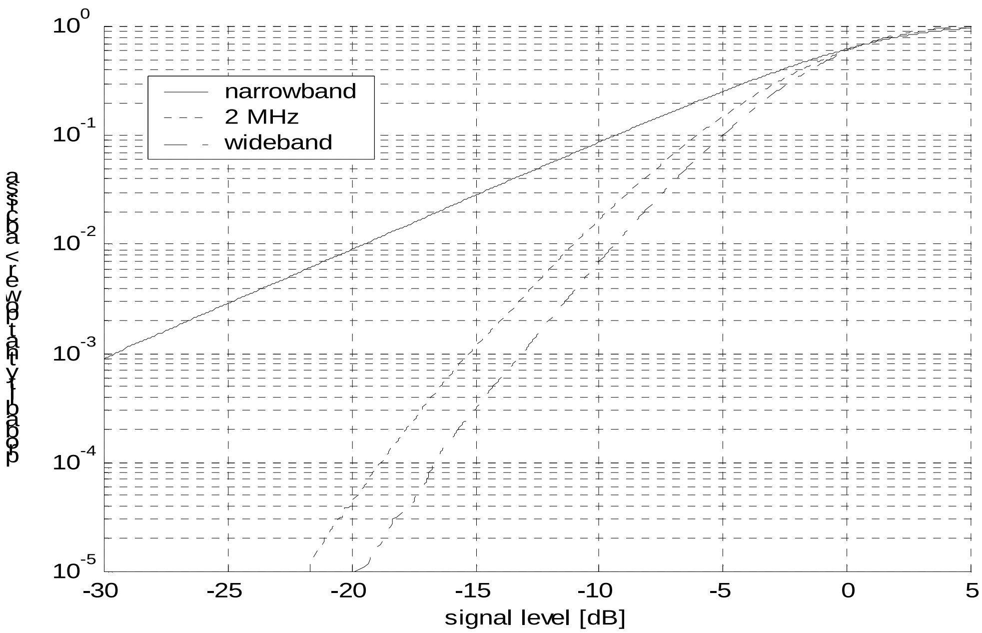

Fig 3.) CDF of channel power (N=200000, OR=4Hz)

Level Crossing Rates / Average Duration of Fades

The level crossing rate (LCR) and average duration of fade (ADF) functions are commonly used for examining the combined effect of fade distribution and Doppler spectrum. The LCR function shows the rate of the signal magnitude dropping below a certain level and the ADF function shows the average duration below that level. These graphs can be useful for choosing and evaluating channel codes for this particular channel.

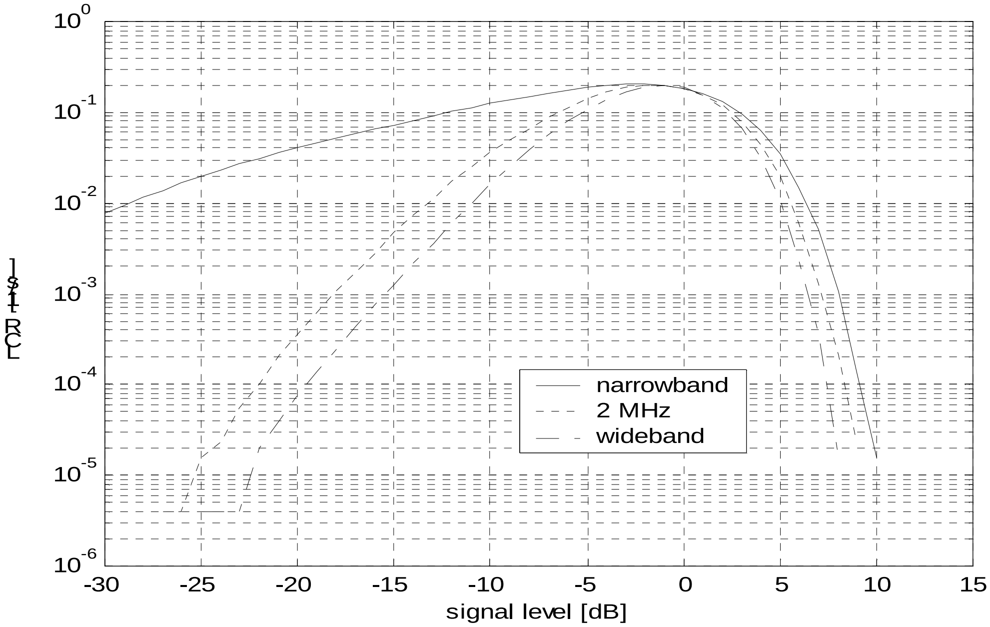

Fig 4.) LCR (N=200000, OR=10Hz)

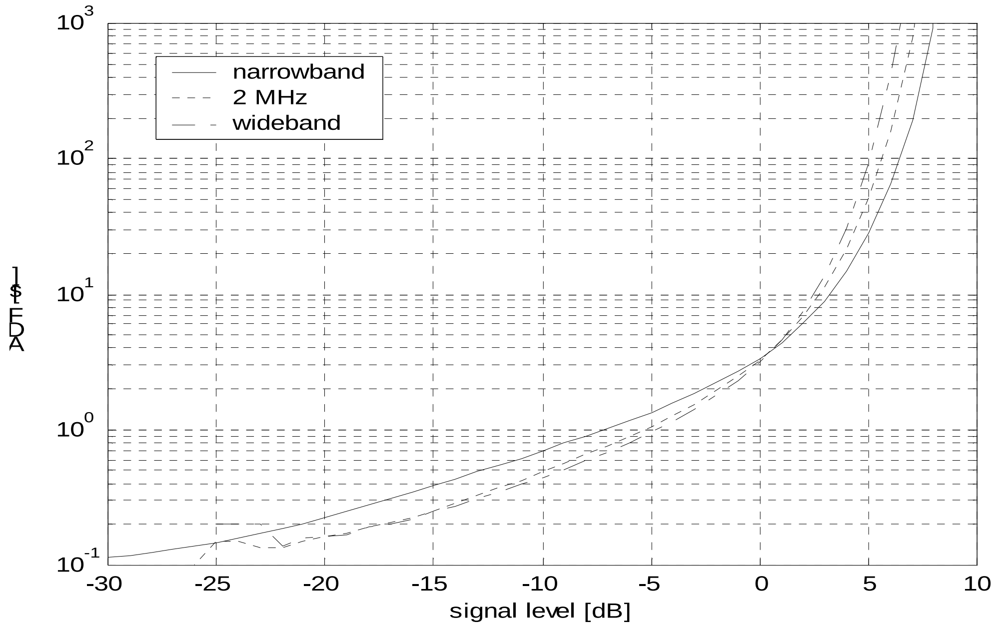

Fig 5.) ADF (N=200000, OR=10Hz)

References

[EKSB01] V. Erceg, K.V.S. Hari, M.S. Smith, D.S. Baum et al, “Channel Models for Fixed Wireless Applications”, IEEE 802.16.3c-01/29r1, 23 Feb. 2001