Channel Modeling Notes

Charles Rino

September 2010

1 Background

Channel model, as introduced in the author's book *The Theory of Scintillation with Applications in Remote Sensing*

`http://www.wiley.com/WileyCDA/WileyTitle/productCd-047064477X.html,`

provides a framework for analyzing the effects of propagation disturbances on communication and remote sensing systems. A channel model imposes a random modulation on a signal that would otherwise be detected in a background of additive white noise. In book Chapter 5 the class of waveforms used for remote sensing and communication are organized by a hierarchy of time scales over which signal processing operations are applied. The impact of the random modulation depends on the frequency, spatial, and temporal coherence of the disturbance. For most signal processing applications, the temporal coherence is most important. In effect, individual waveforms are undistorted, but their amplitude and phase varies from waveform to waveform. The models described in book Chapter 5 provide the analysis tools needed for performance analysis in disturbed propagation environments. Book Chapter 5 also illustrates some basic signal processing operations with real data. Channel modeling and data processing applications are not built around a central utilities like PropCodes 1, 2, 3, and 4. Rather, specific applications draw on a collections of utilities. Waveform simulations are to derive realizations of channel transfer function.

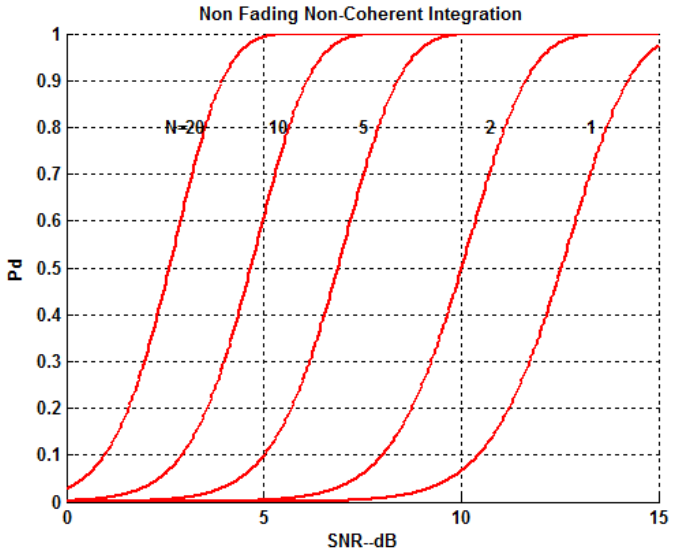
Running the script `SetPath4ChannelModel` in the subdirectory

`···\Examples`

will place the appropriate folders containing these utilities on the MATLAB path. The user will be prompted to identify the MATLAB path to the folder `\PropagationCode1`, which can be downloaded from

`http://www.mathworks.com/matlabcentral/fileexchange/28800-2-d-propagation-simulation.`

`PropCode1` is used to simulate the effects of a channel transfer function realization.



2 Channel Modeling Utilities

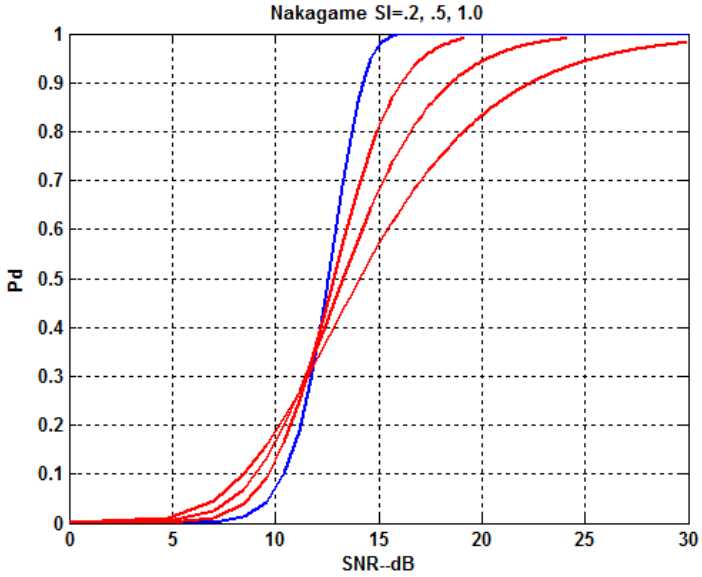
The folder

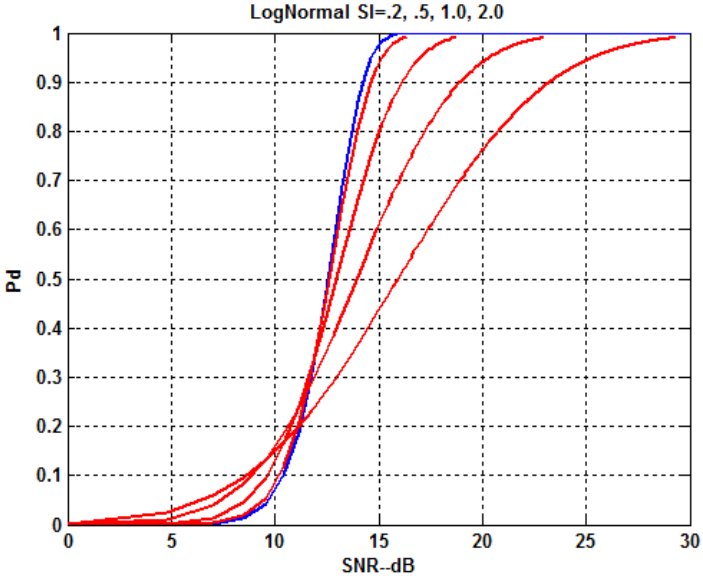
$$\cdots\backslash\texttt{PdComputation}$$

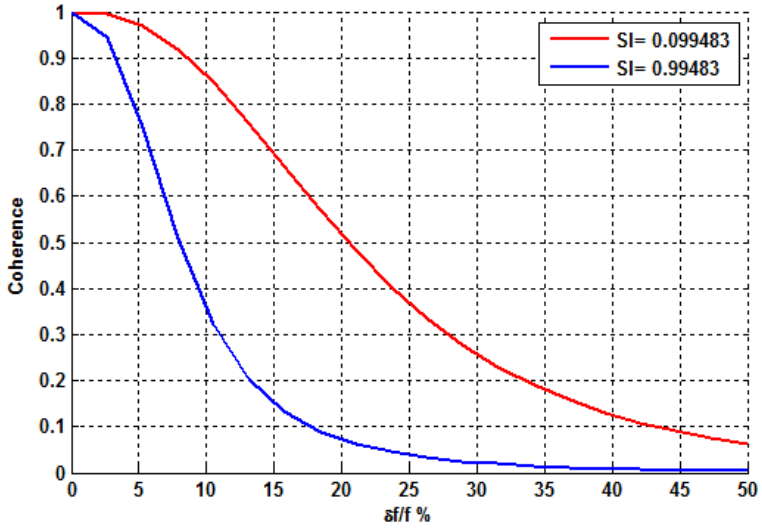
contains MATLAB scripts for computing probability density functions (PDFs), cumulative density functions (CDFs), and detection probabilities. The scintillation Pd computations are described in book Sections 5.2.1.1 and 5.2.1.2. Executing `DisplayPd_Richards` will reproduce Figure 2 below, which is book Figure 5.1. Similarly, executing `PdNakagami` and `PdLogNormal` will reproduce Figures 2 and 2 below, which are book Figures 5.2 and 5.3. Other utilities in the `PdComputation` folder will calculate standard radar Pd fading detection probabilities. Approximate Pd formulas can be found in Chapter 6.3 of *Fundamentals of Radar Signal Processing*, Mark Richards, McGraw-Hill, 2005. The MATLAB utilities in $\cdots\backslash$`PdComputation` compute the exact forms as described in references to papers by Swerling and Schneidman. These utilities are applicable to frequency-flat fading, which implies that the channel modulation is invariant over the duration of the waveform. The results also depend on hypothesized fade distributions, which are not accurate under strong scatter conditions. As discussed in the book, simulations provide the most accurate performance assessments.

2.1 Frequency Coherence

Non-dispersive (frequency independent) propagation is established by the channel coherence bandwidth. Electronic components and the ionosphere exhibit


Nakagame SI=.2, .5, 1.0
Pd
SNR--dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
30


LogNormal SI=.2, .5, 1.0, 2.0
Pd
SNR--dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
30

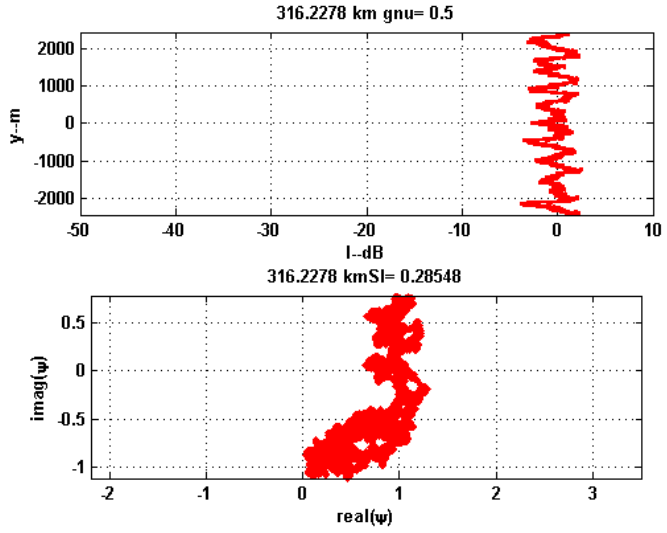


a measurable frequency dependence that usually can be compensated. However, scintillation structure can induce a random frequency decorrelation that distorts transmitted waveforms. A analytic formulas derived in book Chapters 3 and 4 provide estimates of the potential frequency decorrelation. Executing the MATLAB script `FreqCoherence` with the default parameters[1] will reproduce Figure 2.1, which is book Figure 5.4 discussed in book Section 5.3.2. The ordinate is the fractional frequency at 400 MHz. The abscissa is the correlation between two frequencies with the fractional frequency separation. The two curves are for weak (red) and strong (blue) scintillation. The assumptions under which the analytic formula applies are discussed in the book Chapters 3 and 4.

2.2 Temporal Coherence

Temporal coherence refers to the time scale of the random modulation propagation disturbances impart to transmitted waveforms. There is a natural tendency to think in terms of discrete random variables that are independent from transmission to transmission. This assumption allows analytic computation of performance bounds, but the modulation itself is continuous. Thus, realizations of the process applied to the actual data processing operations provide the most accurate evaluations of performance. For transionospheric propagation, `PropCode3` would provide the highest fidelity. However, this fidelity comes with a significant cost in setup and processing time. Thus, most simulations are performed using two-dimensional simulations as produced by `PropCode1`.

[1] Hit enter at any input request to use the default value.



In the folder

`...\Examples\TemporalCoherence`

executing the scripts `SetupPropCode1Ex1` followed by `PropCode1` and `SetupPropCode1Ex2` followed by `PropCode1` will generate two sets of output files for subsequent processing. These files can be processed by executing `MakeFigs` with appropriate GUI selection. The summary plots generated from the `SetupPropCode1Ex1` inputs generates a series of summary plots, which include Figures 2.2 and Figures 2.2 (book Figures 5.5 and 5.6) Executing `MakeFig57` will generate Figure 2.2 (book Figure 5.7), which shows the Doppler resolution that can be achieved in the presence of noise only as a reference. Figures 2.2 and 2.2 (book Figures 5.8 and 5.9) show the progressive effects of temporal coherence loss. Note that in 2.2 the Doppler peak is both reduced in amplitude and shifted in frequency.

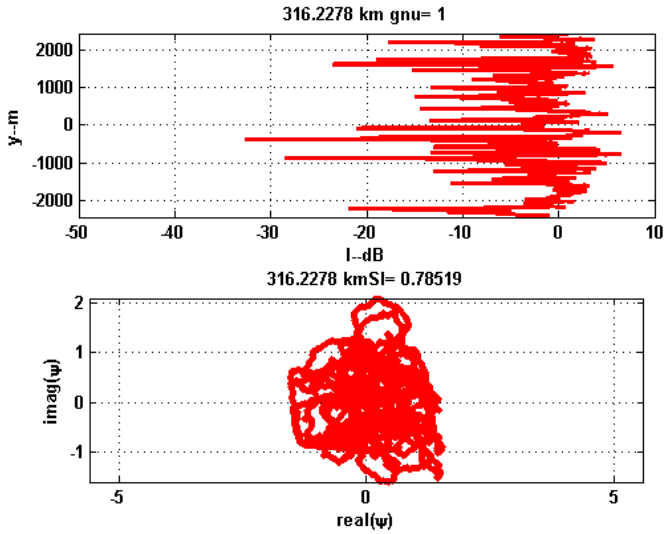
3 Digital Signal Processing Utilities

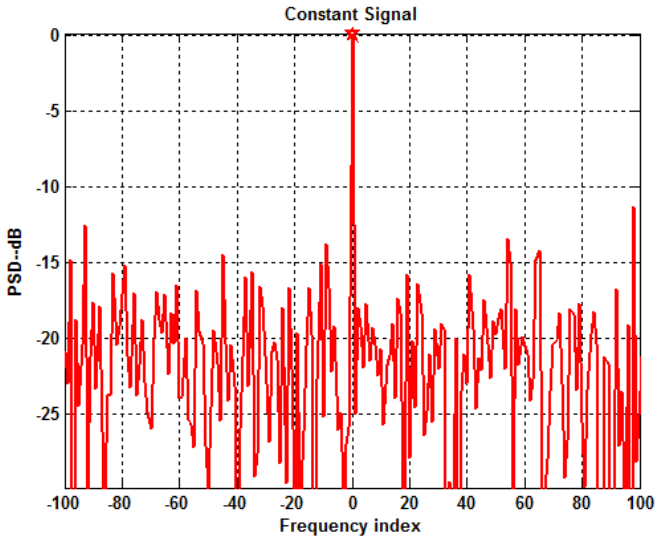
Under the best of conditions digital signal processing will be limited by background noise. In a well designed system the first low noise amplified (LNA) sets the noise level. For example, under the assumption that the signal intensity is captured without processing loss, the measured signal intensity fluctuations from sample to sample comprise both channel-induced fluctuations and noise.

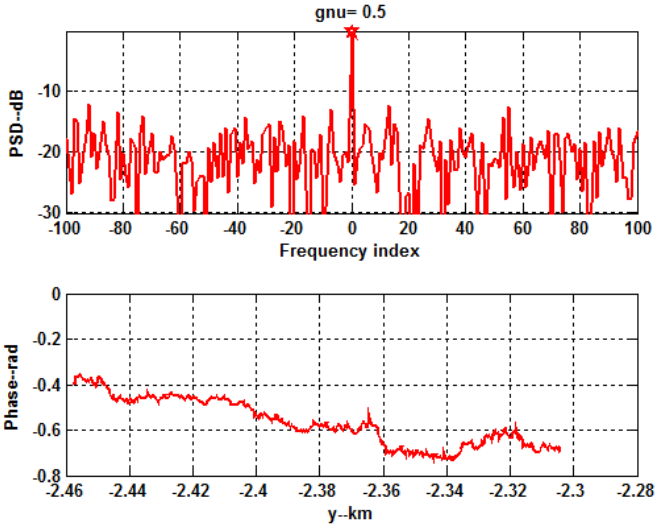
In the folder

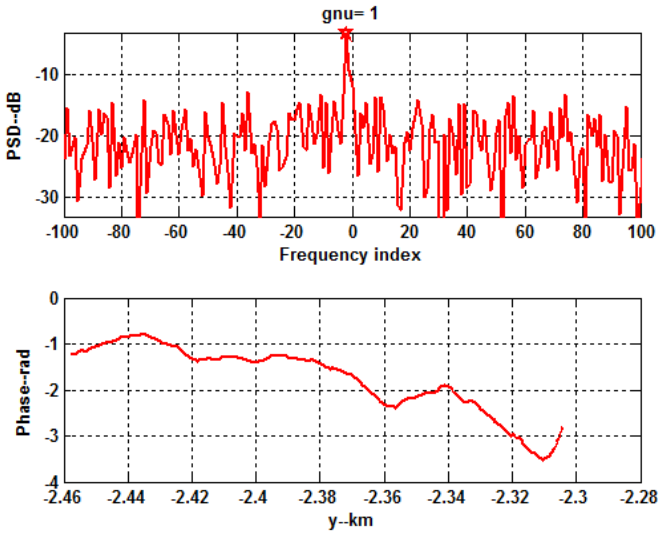
`...\ChannelModeling_Examples\SignalProcessing`

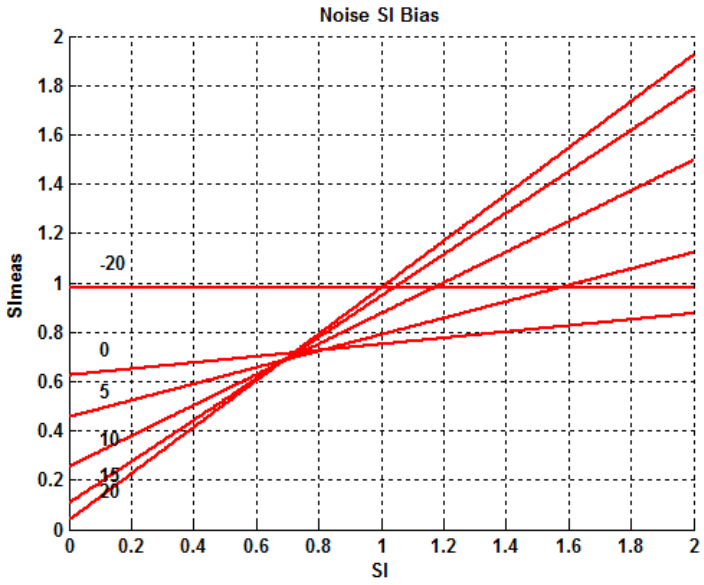
executing the script `SIvsSNR` will reproduce Figure 3 book Figure 5.10, which shows the expectation of the measured scintillation index versus the actual scin-


316.2278 km gnu= 1
y--m
I--dB
316.2278 kmSI= 0.78519
imag(ψ)
real(ψ)


Constant Signal
PSD--dB
Frequency index


gnu= 0.5
PSD--dB
-10
-20
-30
-100
-80
-60
-40
-20
0
20
40
60
80
100
Frequency index
Phase--rad
0
-0.2
-0.4
-0.6
-0.8
-2.46
-2.44
-2.42
-2.4
-2.38
-2.36
-2.34
-2.32
-2.3
-2.28
y--km


gnu= 1
PSD--dB
-10
-20
-30
-100
-80
-60
-40
-20
0
20
40
60
80
100
Frequency index
Phase--rad
0
-1
-2
-3
-4
-2.46
-2.44
-2.42
-2.4
-2.38
-2.36
-2.34
-2.32
-2.3
-2.28
y--km



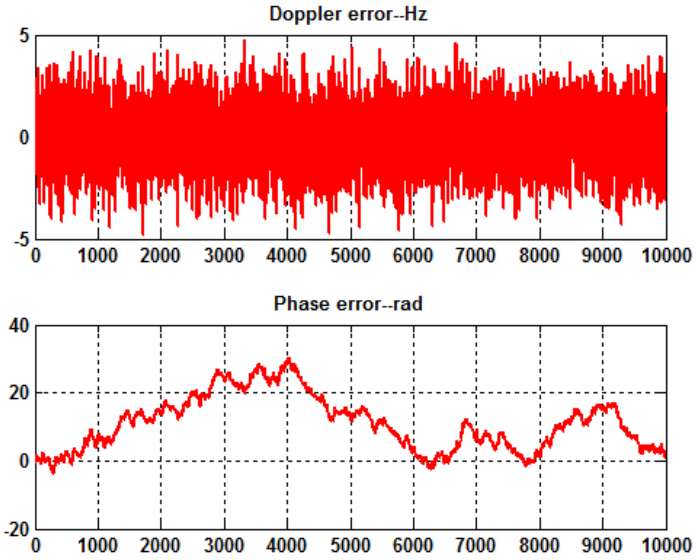
tillation index. Recall that background noise is Rayleigh distributed with a unity scintillation index. Thus, as very low SNRs, the measured scintillation index is close to the noise-dominated value $SI = 1$. To drive the noise bias to a negligible value requires and SNR in excess of 20 dB. At moderate SNRs, corrections can be made if the SNR is known.

The folder

`...\DataAnalysis`

contains a number of utilities, some of which have already been introduced. For example, `boxcar_avg` will perform a sliding average over symmetric window of a specified size. Near the ends of the intervals it uses the available samples. The script `computeSI` computes a local SI estimate using `boxcar_avg`. It also returns the local mean intensity, which can be used for detrending the intensity $I \leftarrow I/\langle I \rangle_N$, where $\langle I \rangle_N$ is the centered boxcar average over N samples, where N is odd. The script `specDen` has already been introduced. It computes a sliding PDF estimate normalized to preserve the variance over the data interval. It will apply a weighting function, but for parameter estimation in power-law environments windowing is not appropriate because it distorts the large scale structure. The script `DOPsim` has also been introduced.

Signal processing with digital-receiver data requires continuous Doppler estimation or frequency tracking. An algorithm based on frequency hypothesis testing is described in the book. The MATLAB frequency tracking algorithm `signalTracker` included in the DataAnalysis folder. Because of receiver specific details that are not relevant to the basic principles, the frequency tracking algorithm is demonstrated here with synthetic data following the discussion in



book Section 5.4.3. Executing the script `DemoFrequencyTrackingAlgorithm` will generate a synthetic quadratic phase signal embedded in white noise. Once the synthetic signal is generated it is processed to track the frequency. Figure 3, which is book Figure 5.12, shows the absolute frequency and phase errors. The frequency errors are uniform, while the phase errors exhibit a low-frequency meander. As discussed in book Section 5.4.3, this is a direct consequence of the integral relation between phase and frequency. Integration colors the white noise, which makes it behave like Brownian motion.