SSY135 – Wireless Communications MATLAB Project

Henk Wymeersch,
henkw@chalmers.se

***Abstract*—In this project, the performance of an OFDM communication system over a fading channel is evaluated. To this end, a communications engineer needs to understand the behavior of the channel, which drives the design of such a communication system. A wireless channel will be simulated and various properties of this channel will be studied in Part I. The knowledge gained from this wireless channel will be used in designing an OFDM communication system in Part II. Please read this entire document before starting the project.**

I. Part I: Design and Simulation of Wireless Fading Channels

The objective of Part I is to get a good understanding of wireless channels and the performance-limiting phenomena that occur in such channels. After completing Part I, you should be able to:

- Simulate a Rayleigh and Rician fading channel with a Clarke's (also known as Jakes') Doppler spectrum;
- Assess the quality of the simulated fading with respect to autocorrelation and power spectral density;
- State advantages and disadvantages of the methods used to generate the fading channel;
- Simulate a Rician channel with time and frequency-selectivity, with uniform power delay profile.
- Explain how delay spread and Doppler spread affect the variation of the time and frequency-varying channel.

The remainder of this section is structured as follows. First, we recap some basic concepts of Rayleigh fading channels. Then we describe two methods to generate flat-fading Rayleigh fading. Then, we extend these methods to generate WSS-US channels that vary over time and frequency. Finally, the simulation tasks are described.

A. Background information

A Rayleigh fading channel $c(t)$ according to Jakes' model is a unit-energy Gaussian process with mean 0 and autocorrelation function (ACF)

$$A_c(\Delta t) = \mathbb{E}[c^*(t)c(t+\Delta t)] = J_0(2\pi f_D \Delta t), \qquad (1)$$

where $J_0(x)$ is the Bessel function of the first kind of order 0, f_D is the Doppler shift in Hz and Δt is a time difference in seconds.[1] The Doppler spectrum (the power spectral density - PSD) is the Fourier transform of the ACF $A_c(\Delta t)$

$$S_c(f) = \mathcal{F}[A_c(\Delta t)] = \int_{-\infty}^{\infty} A_c(\Delta t) e^{-j2\pi f \Delta t} \mathrm{d}\Delta t \qquad (2)$$

$$= \begin{cases} \frac{1}{\pi f_D} \frac{1}{\sqrt{1-(f/f_D)^2}}, & |f| \le f_D, \\ 0, & \text{otherwise.} \end{cases} \qquad (3)$$

Since the channel Doppler spectrum is band-limited to $[-f_D, +f_D]$ we can simulate it in MATLAB by generating samples $c(nT_s)$, for a suitable sampling rate $1/T_s$.

We recall from linear systems theory that a wide-sense stationary random process $x(t)$ with PSD $S_x(f)$, filtered with a filter $g(t)$ leads to an output process $c(t)$ with PSD

$$S_c(f) = |G(f)|^2 S_x(f), \qquad (4)$$

where $G(f)$ is the Fourier transform of $g(t)$. Hence, when $x(t)$ is white (i.e., with a flat power spectral density and a Dirac delta autocorrelation function), $S_x(f) = 1$ and $S_c(f) = |G(f)|^2$. We denote by $\tilde{G}(f)$ the periodic extension of $G(f)$ with period $1/T_s$

$$\tilde{G}(f) = \sum_{m=-\infty}^{+\infty} G(f - m/T_s). \qquad (5)$$

B. Methods to generate a Rayleigh fading channel

1) The Filter Method: Choosing $G(f) = \sqrt{S_c(f)}$, we find that $c(t)$ has the Doppler spectrum from (3). It can be shown that[2]

$$g(t) = \mathcal{F}^{-1}[\sqrt{S_c(f)}] = \begin{cases} \dfrac{J_{1/4}(2\pi f_D |t|)}{\sqrt[4]{|t|}}, & t \neq 0, \\ \dfrac{\sqrt[4]{\pi f_D}}{\Gamma(5/4)}, & t = 0, \end{cases} \qquad (6)$$

where $\Gamma(x)$ is the Gamma function. Since $G(f)$ is bandlimited, it can also be sampled at rate $1/T_s$, greater than the bandwidth of $G(f)$. This leads to the following recipe[3] to generate $c(nT_s)$:

1) Choose T_s such that no aliasing occurs, i.e., $1/T_s > 2f_D$. Recall that aliasing means that periodic copies of the spectrum caused by sampling overlap in the frequency domain.

[1] Sometimes it can be useful to change units, for instance express the Doppler shift in MHz and the time difference in ms, provided their product has no unit.

[2] This requires some tedious math.

[3] You should understand and be able to explain each step in this recipe!

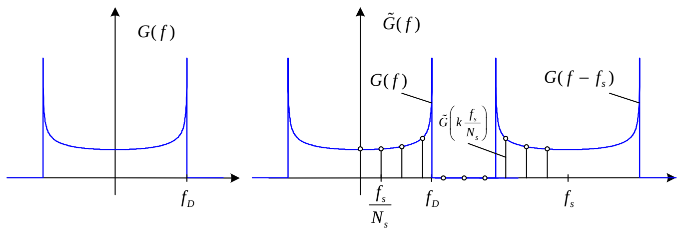

Fig. 1. Example of how $\tilde{G}(f)$ and its samples $\tilde{G}(kf_s/N_s)$ for $N_s = 10$ and $k = 0, 1, \ldots, N_s - 1$ are found from $G(f)$.

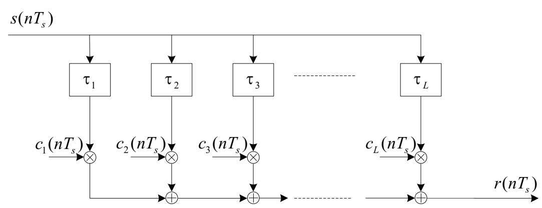

Fig. 2. Discrete-time model of a time and frequency-varying channel with L taps and sample period T_s. Here $\tau_i = (i-1)T_s$.

2) Choose how many samples of the channel to generate, say N_s.
3) Store $g(nT_s)$, for $n = -N_gT_s, \ldots, N_gT_s$ in a vector. Here N_g should be chosen carefully. Too small N_g will lead to artifacts in the PSD and undesired high-frequency components. Too large N_g will lead to high complexity of the method. We denote the filter by the vector $\mathbf{g}$. Make sure $\mathbf{g}$ is unit energy.
4) Generate independent samples $x(nT_s)$ from a zero-mean, unit-variance, complex Gaussian distribution,[4] for $n = 0, \ldots, N_s - 1$. We denote the samples by a vector $\mathbf{x}$.
5) Convolve $\mathbf{x}$ with $\mathbf{g}$, leading to $\mathbf{c}$. If you use the `conv()` operator, $\mathbf{c}$ will be of length $N_s + 2N_g$.
6) Discard transient samples: remove the first N_g samples and the last N_g samples. Check that the statistics are correct, including that $\mathbb{E}\{|c(nT_s)|^2\} = 1$, $\mathbb{E}\{c(nT_s)\} = 0$, and that $c(nT_s)$ has a complex Gaussian distribution.

2) The Spectrum Method: As an alternative to the filter method, we can work in the frequency domain. This avoids a costly convolution operator. The recipe[5] then becomes:

1) Choose T_s such that no aliasing occurs, i.e., $f_s = 1/T_s > 2f_D$.
2) Choose how many samples of the channel to generate, say N_s.
3) Store $\tilde{G}(kf_s/N_s)$ for $k = 0, \ldots, N_s - 1$ in a vector $\mathbf{G}$. Recall that $\tilde{G}(f)$ is the periodic version of $G(f)$ with period $1/T_s$. See Figure 1 for more elaboration.
4) Generate independent samples $X(nT_s)$ from a zero-mean, complex Gaussian distribution, for $n = 0, \ldots, N_s - 1$. We denote the samples by a vector $\mathbf{X}$.
5) Multiply $\mathbf{X}$ with $\mathbf{G}$ and denote the result $\mathbf{C}$.

[4] `x = (randn(Ns, 1) +j*randn(Ns, 1))*a;` where `a` is a constant for you to determine.

[5] Yes, you should again understand and be able to explain each step in this recipe!

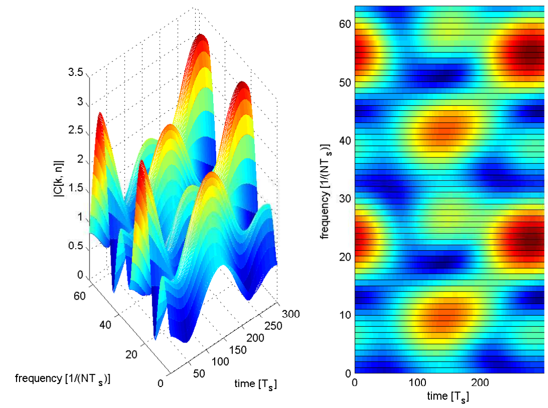

Fig. 3. Plot of the magnitude time and frequency-varying channel response for $L = 3$. The plot on the left is generated with `mesh(0:M-1, 0:N-1, abs(C));` and the one on the right with `surf(1:M, 0:N-1, abs(C), 'MeshStyle', 'row')` .

6) Apply an inverse DFT to $\mathbf{C}$ and denote the result $\mathbf{c}$. Check that $\mathbb{E}\{|c(nT_s)|^2\} = 1$.

C. Methods to generate a time and frequency-varying channel response

The methods above can easily be extended to generate channels that are varying in time and frequency, following the WSS-US model. The basic model is shown in Figure 2. Each of the taps $c_\ell(nT_s)$ is an independent time-varying process as generated in the previous sections. The recipe thus becomes:

1) Choose a sampling time T_s, such that no aliasing occurs.
2) Choose a number of taps L such that $LT_s \geq \tau_{\mathrm{DS}}$, where τ_{DS} is the delay spread of the channel.
3) For $\ell = 0, \ldots, L-1$ independently:
   a) Generate $c_\ell(nT_s)$, $n = 0, \ldots, N_s - 1$ according to the filtering method
   b) OR: Generate $C_\ell(kf_s/N)$, $k = 0, \ldots, N_s - 1$ according to the spectral method method and apply an IDFT
4) For $n = 0, \ldots, N_s - 1$
   a) Apply an M-point DFT to a zero-padded version of the channel at time nT_s:$[c_0(nT_s), c_1(nT_s), \ldots, c_{L-1}(nT_s), \mathbf{0}_{M-L}]$, where $\mathbf{0}_{M-L}$ is a vector of $M - L \geq 0$ zeros.
   b) The result is denoted by $C(f, nT_s)$, with entries at frequencies $f = 0, 1/(MT_s) \ldots, (M-1)/(MT_s)$.
5) Denote $C(m/(MT_s), nT_s)$ by $C[m, n]$. This leads to a matrix $\mathbf{C}$ with M rows and N_s columns.

The magnitude time and frequency-varying channel response can be viewed as a surface over the time-frequency plane (e.g., as in Figure 3).

D. Simulation Task

1) Rayleigh and Rician Flat Fading: Simulate a frequency-flat Rayleigh fading gain process for a carrier frequency of 2 GHz, a transmitter-receiver relative speed 30 km/h, and sample interval $T_s = 0.1$ ms for both the filter method and the spectrum method. For a Rician channel, set $c(nT_s) = c_{\text{Rayleigh}}(nT_s) + k_c$ and then renormalize so that $\mathbb{E}\{|c(nT_s)|^2\} = 1$. Here $k_c \geq 0$ is a constant and $c_{\text{Rayleigh}}(nT_s)$ is the Rayleigh part of the channel. You can choose filter or spectral method.

- What is the largest value of T_s you can use. Does $T_s = 0.1$ ms meet this requirement?
- Estimate the power spectral density of $c(nT_s)$ in MATLAB and compare with the theoretical one, i.e., $S_c(f)$ as defined in (3).
- For $k_c = 0, 1, 10$, verify the generated channel by evaluating its characteristics and comparing with theory for both methods. How does k_c relate to the Rician K-factor? Estimate the probability density function (pdf) and cumulative density function (cdf) of the samples $|c(nT_s)|$ and compare with theory. Estimate the autocorrelation function of $c(nT_s)$ and compare with the theoretical one, i.e., $A_c(\Delta t)$ as defined in (1).
- Explain the advantages and disadvantages of the different methods (filter method vs spectral method) when N_s (i.e., the length of the simulated fading vector), f_D, and T_s change. Think in terms of complexity, autocorrelation functions, etc.

1 bonus point: For a more efficient implementation, use the overlap-add method[6] and compare with the spectrum method.

2) Time and Frequency-Varying Rician Fading Channels: Produce plots like right side of Figure 3 for $N = 300$ time samples and $M = 64$ frequency samples. Let the channels have $L \in \{1, 2, 3\}$ taps. Let the tap gains $c_\ell(nT_s)$ be Rician fading with Clarke's spectrum, with flat power delay profiles, and normalized Doppler frequency $f_D T_s \in \{0.1, 0.005\}$ and $k_c \in \{0, 1, 10\}$. In total 18 combinations of L, $f_D T_s$, and k_c should be plotted, added as appendices to your report. Explain how different parameters will affect the channel response along time and frequency axes. For each combination of L and f_D, state what is the corresponding delay spread (in ms) and user velocity (in km/h).

1 bonus point: Change your code to have an arbitrary power delay profile (PDP). For an exponentially decaying PDP, generate a time-frequency plot as before. Can you see any difference?

[6] See https://en.wikipedia.org/wiki/Overlap-add_method for more details.

II. Part II: OFDM Communication over Time-Varying Frequency-Selective Fading Channels

In Part II, you will design and execute an OFDM communication system over the time-varying frequency selective channel from Part I. After completing Part II, you should be able to

- Design and simulate an OFDM communication system to transmit over the simulated channel.
- Choose the design parameters based on channel properties and system requirements, including the number of subcarriers, the cyclic prefix, the energy per subcarrier.
- Evaluate the performance of an OFDM system through scatter plots and symbol error rate plots.

The remainder of this section is structured as follows. First we recap some basic concepts of OFDM, including the operation of the transmitter, channel and receiver. Then, the simulation tasks are described.

A. Background

The entire OFDM system is shown in Figure 4.

1) Transmitter: We generate QSPK sequences of length N. The m-th sequence of length N is denoted by $\mathbf{s}^{(m)}$, with entries $s_k^{(m)}$, satisfying an energy constraint $\mathbb{E}\{|s_k^{(m)}|^2\} = E$. Then we apply an IFFT to obtain the signal in the time domain, say

$$z_n^{(m)} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} s_k^{(m)} \exp\left(j2\pi \frac{nk}{N}\right), \qquad n = 0, 1, \ldots, N-1 \tag{7}$$

Note that in MATLAB this would be implemented as `z=sqrt(N)*ifft(s);` Then a cyclic prefix of length $N_{\text{cp}} \geq 0$ is added. This leads to a vector of length $N + N_{\text{cp}}$ with $z_{-k}^{(m)} = z_{N-k}^{(m)}$, for $k = 1, \ldots, N_{\text{cp}}$. We will not consider the D/A and A/D in this project and instead make the implicit assumption that the pulse shape involved is a square pulse of duration T_s. This means that the complex numbers $z_k^{(m)}$ ($k = -N_{\text{cp}}, \ldots, N-1$, $m = 0, 1, \ldots$) are sent sequentially over the channel at a rate $1/T_s$.

2) Channel: We use the channel generated in Section I-C. The channel during OFDM symbol m is denoted by $c_\ell^{(m)}(nT_s)$, $\ell = 0, 1, \ldots, L-1$ and $n = -N_{\text{cp}}, \ldots, N-1$. Here L is the number of channel taps, set such that LT_s is larger than the delay spread of the underlying physical channel. Note that

$$c_\ell^{(m)}(nT_s) = c_\ell\left((m-1)T_o + nT_s\right) \tag{8}$$

where $T_o = (N + N_{\text{cp}})T_s$ is the duration of 1 OFDM symbol. Convolve the sequence $z_k^{(m)}$ with the channel $c_\ell^{(m)}(nT_s)$ using MATLAB's `conv` operator. The OFDM system should be designed with the following constraints in mind:

- The channel should be close to constant during an OFDM symbol. That is, $c_\ell\left((m-1)T_o + nT_s\right) \approx c_\ell\left((m-1)T_o\right)$ or, equivalently $c_\ell^{(m)}(nT_s) \approx c_\ell^{(m)}(0)$. This is satisfied when $(N + N_{cp})f_D T_s \ll 1$.
- The channel should have a delay spread smaller than the cyclic prefix duration: $N_{\text{cp}} \geq L-1$.

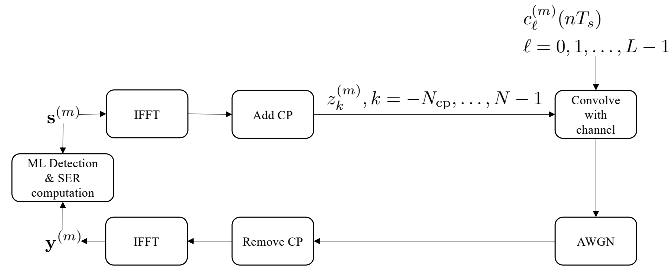

Fig. 4. Block diagram of uncoded OFDM system over a frequency-selective channel with QPSK modulation.

3) Receiver: In the receiver in Figure 4, the received signal is assumed to be filtered with an anti-aliasing lowpass filter with cutoff frequency $1/(2T_s)$ and then sampled with the sample rate $f_s = 1/T_s$. These aspects will be ignored in this project and we will work directly with the discrete-time signals. Add complex AWGN samples with variance N_0 to the received signal. Remove the samples corresponding to the cyclic prefix. Apply an FFT to the remaining N samples for the m-th OFDM symbol to obtain samples of the form

$$y_n^{(m)} = C_n^{(m)} s_n^{(m)} + w_n^{(m)}, \tag{9}$$

where $C_n^{(m)}$ is related to the FFT of $c_\ell^{(m)}(nT_s)$ and $w_n^{(m)}$ is the FFT of the noise. Finally, perform maximum likelihood detection to recover $s_n^{(m)}$, assuming the receiver knows the channel. Make sure to take care of any residual scalings and rotations.

B. Simulation Task

You will design a communication system for a fading channel. Assume that the tap gains $c_\ell(nT_s)$ are i.i.d. Rayleigh-fading with Clarke's spectrum and a flat power delay profile (so $\mathbb{E}\{|c_\ell(nT_s)|^2\} = \frac{1}{L}$, $\forall \ell$). The communication system is operating at 2 GHz carrier frequency with a bandwidth of 1 MHz. The noise spectral density receiver is $N_0/2 = 2.07 \times 10^{-14}$ μW/Hz. The wireless link is experiencing a path loss of 101 dB (we assume the shadow fading is negligible). The speed of receiver is 15 m/s and the delay spread is 5.4 μs. To generate the channel you can use the results from Part I or use the provided function Fading_Channel.p. (See Appendix III-B).

- Choose appropriate values for N and N_{cp}. Verify through calculation and simulation that the conditions from Section II-A2 are satisfied. Relate (though a mathematical expression) E to the average transmit power P (if the average transmit power P is 0.1 W, what is the value of E?).
- Generate independent QPSK data over all N subcarriers for multiple OFDM symbols and implement the transmitter and receiver. To test the system, run the transmitter and receiver over an AWGN channel with $c_0(nT_s) = 1, \forall n$ and $c_{\ell \neq 0}(nT_s) = 0, \forall n$. Make scatter plots of the received signal (9) on a few subcarriers. Explain how you implemented the different stages of the transmitter and receiver.

- Now include the time-varying frequency selective channel $c_\ell(nT_s)$ between the transmitter and the receiver. Make scatter plots of the received signal on a few subcarriers. Explain how you implemented the different stages of the transmitter and receiver. Derive and implement maximum likelihood detection for each subcarrier. Predict the SNR per subcarrier theoretically and compare with simulations. Explain your reasoning. By varying E, plot the average symbol error rate as a function of E/N_0 (choose a meaningful range of symbol error rates, e.g., between 0.5 and 0.001) and compare with theoretical symbol error rate. What is the effective data rate?
- What happens with the scatter plots when you choose N_{cp} too small? What happens to the symbol error rate?
- *2 bonus points:* Assume that the receiver is no longer moving but we keep the same N and N_{cp} as before. The data symbols of the first $Q > 0$ OFDM symbol are known to the receiver (so-called QPSK pilots), while the remaining OFDM symbols carry unknown symbols. Implement a channel estimation method that can use the known pilots and use the estimate to recover the data symbols. What value of Q would you recommend to choose and why?

III. Appendices

A. MATLAB Tips

MATLAB have several in-built functions that may be useful in this project. Some useful functions are given below.

The `help` command is perhaps the most useful of all commands. When still in doubt after reading the help files, you can also read the source code (or search it online where you can also find several examples). The command `which cmd` gives a path to the m-file that implements the command `cmd` (if `cmd` is not a built-in function), and `type cmd` prints the source of the m-file in the command window.

command	**meaning**
`conv`	computes the convolution between two signals
`besselj`	computes the Bessel function of the first kind
`fft`	computes the discrete Fourier transform
`fftshift`	shift zero-frequency component of discrete Fourier transform to center of spectrum
`ifft`	computes the inverse discrete Fourier transform
`filter`	filter data with an infinite impulse response (IIR) or finite impulse response (FIR) filter
`gamma`	computes the Gamma function
`pwelch`	estimates the power spectral density using Welch's method
`randn`	draws i.i.d. random numbers from a Gaussian distribution
`reshape`	reformats the dimensions of a matrix. This is useful to implement a block interleaver.
`xcorr`	estimates the cross-correlation or autocorrelation. The 'unbiased' option is recommended in this assignment

B. How to use the Fading_Channel.p

Before calling this function, you need to download and copy "Fading_Channel.p" to your project folder. This function implements a time-varying frequency selective channel with tap delays specified in tau and power delay profile in P. The tap gains are Rayleigh fading processes with normalized Doppler frequency fdTs, where Ts is the sample interval. The inputs of the function are a vector $\mathbf{s}$ (of length, say L_s), a vector τ (of length, say L_τ) of delays, a scalar $f_D T_s$ representing the normalized Doppler frequency, the sampling interval T_s and $\mathbf{P}$, the power delay profile. The outputs are $\mathbf{r}$, which is the noise-less channel output (length = $L_s + \max(\tau)$) and $\mathbf{h}$, which is the channel, represented as a matrix of size $(L_s + \max(\tau)) \times L_\tau$.

Example:

```
s % the channel input
P = [0.5 0.5]; % Power delay profile
tau = [0 4]; % Path delays in samples
M = 300; % Number of time samples
fd = 100; % Doppler frequency
Ts = 1e-6; % Sample time
fdTs = 1e-4; % Normalized Doppler frequency
[r, h] = Fading_Channel(s, tau, fdTs, P)
```

When $(N + N_{\mathrm{cp}}) f_D T_s \ll 1$, the rows of $\mathbf{h}$ (the second output of Fading_Channel.p) are the same. Therefore, you can calculate the time domain response of the fading channel related to each OFDM symbol from the first row of $\mathbf{h}$. By defining `hm` as the time domain response of the mth OFDM symbol, you calculate $C_n^{(m)}$ `=tmp(n)` where `tmp=fft(hm,N)` (possibly up to a scaling).

C. Report Guidelines

The report for each part should be written in the IEEE Transactions format, following the IEEEtran LaTeX class (see http://www.ieee.org/documents/IEEEtran.zip for the package and a comprehensive example). To run LaTeX without installing any software and to write your report together, we recommend you use the online tools at overleaf.com. For each part of the project, the report may not exceed 5 double column pages (excluding figures) and should:

- include title and names of the authors;
- state the problem addressed and outline its importance;
- describe briefly the milestones of the project;
- include the results from your simulations, along with thorough discussion and interpretation;
- explain how you would modify/change the design for better performance;
- add conclusions.

Part I of the project deliverable should contain your explanation of how you implemented the two methods and what parameters you have used for verifying the generated channels. Part II of the project deliverable should include how the communication system was designed and implemented. All team members should contribute equally towards accomplishing the tasks of both parts of the project. A section in the report of Part I and Part II should clearly state the members' contributions, including writing the report and MATLAB simulations. Points may be deducted for cases in which figures are not properly labelled, results from simulations are not clearly explained, results are not compared with the theory.

For additional formatting and submission requirements, see the course PM. If there is a conflict between the course PM and this document, this document takes precedence.