Design and Simulation of Wireless Fading Channels

Yaxi Xie
dept. of Electrical Engineering
Chalmers University of Technology
yaxi@student.chalmers.se

Lise Aabel
dept. of Electrical Engineering
Chalmers University of Technology
aabel@student.chalmers.se

Shabir Kashif
dept. of Electrical Engineering
Chalmers University of Technology
shabirk@student.chalmers.se

***Abstract*—To contemplate the wireless channel effect on a signal, two different models for fading channels are studied during different circumstances. The main objective is to do a parameter analysis, and the results are presented through plots for different scenarios.**

I. Introduction

In wireless communication, understanding the free space channel effects on a transmitted signal is the key to successful signal restoration. Well developed channel models enable parameter studies, providing a better understanding of the effects on a signal. In this project, the Rayleigh and Rician channel fading models are studied in MATLAB. The Rayleigh fading model describes a scenario of a rich multipath environment where there is no dominating path between the transmitter and the receiver. The Rician fading model describes the scenario where there is a dominating path of a certain magnitude.

The report contains two parts, starting with narrow band, frequency flat channels and moving on to a scenario where the channel has a certain delay, depending on the number of signal paths. In that case the channel is varying in both time and frequency domain.

II. Part 1: Rayleigh and Rician Flat Fading

Generally, the time dependence of a signal is divided into two domains; time t and delay τ. These are properties that are profitably analyzed separately. A channel is considered narrow band if the symbol time T_s is much larger than the delay spread, i.e. that all the paths approximately arrive at the same time. This is equal to that the signal bandwidth is much smaller than the coherence bandwidth, i.e. frequency flat. Coherence bandwidth is the frequency interval for which the channel can be considered constant. For that scenario, the delay spread becomes negligible and the signal is analyzed in time t.

To generate a channel for analysis, two different methods are compared; the filter method and the spectrum method. The general difference is whether to mathematically operate in time or frequency domain. A brief theoretical introduction to both methods is described here, and the implemented MATLAB code for all tasks can be found in the appendix B and C of the report.

The auto-correlation function (ACF) of a narrow band, time varying channel is, according to Jakes' model, described by

$$A_c(\Delta t) = J_0(2\pi f_D \Delta t), \tag{1}$$

where J_0 is the zeroth order Bessel function of the first kind and f_D is the Doppler frequency, caused by movement in the channel. The ACF of the channel describes the relation between the channel at a moment in time t, and at the moment Δt later. According to Jakes' model, the time dependent channel will be uncorrelated at $\Delta t \approx 0.4/f_D$. After that, the channel will be independent of what it was at $t = 0$. A higher Doppler frequency means objects in the channel move faster relative to each other, hence the channel will become uncorrelated faster.

From the ACF, $A_c(\Delta t)$, the Doppler power spectral density (PSD) can be calculated. It is defined as the Fourier transform of $A_c(\Delta t)$ and is denoted $S_c(f)$. The Doppler PSD describes the power distribution for the channel between the different frequency components from $[-f_D, f_D]$. For a white Gaussian process the PSD is constant, since it is theoretically an even composition of all frequencies. Modelling wide-sense stationary random process as a white, Gaussian random process, consequently means that after filtering it with $g(t)$, the Doppler PSD of the output is equal to the absolute square of a filter $G(f)$:

$$S_c(f) = |G(f)|^2. \tag{2}$$

A. The filter method

Since we can define $S_c(f)$, we can calculate the filter in time domain, as

$$g(t) = \mathfrak{F}^{-1}[\sqrt{S_c(f)}]. \tag{3}$$

For a chosen number of discrete time channel samples N_s, sample time T_s and filter samples N, the discrete time domain filter can be generated as $g(nT_s)$ for $n = [-N, N]$. We generate a number of time samples $x(nT_s)$ for $n = [0, N_s - 1]$ as a complex Gaussian distribution. A time domain channel can then be created through the convolution $c(t) = g(t) \star x(t)$.

B. The spectrum method

In order to avoid the convolution operation, the channel can be generated in frequency domain by a multiplication such that $C(f) = G(f) \cdot X(f)$. The time domain channel is then found through the inverse Fourier transform.

C. Simulation task 1

Given parameters for the scenario: carrier frequency $f_c = 2$ GHz, sampling interval $T_s = 0.1$ ms, transmitter and receiver relative velocity $v = 30$ km/h. The number of channel samples was after some analysis chosen as $N_s = 10^5$, providing a complex Gaussian distribution that has a high statistic reliability, but keeping the simulation time of the convolution relatively low.

1) What is the largest value of T_s you can use? Does $T_s = 0.1$ ms meet this requirement?: According to the Nyquist theorem, the sample frequency $f_s = 1/T_s = 10$ kHz must fulfil $f_s > 2f_D$ at baseband, in order to avoid aliasing. f_D is the Doppler frequency caused by relative movement, defined as $f_D = v/\lambda_c = 55.56$ Hz. $f_s \approx 111Hz$ thus sets the limitation $T_s < 9$ ms, which is the largest possible sample time considering aliasing. In order to consider the channel narrow band, it is also required that $T_s >> delayspread$, such that the approximation of a frequency flat channel holds.

2) 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) in the project description: The theoretical and estimated channel PSDs using the two channel generation methods are depicted in figure 1 and 2. The theoretical Doppler spectrum displays that the largest amount of power is allocated at $-f_D$ and f_D. The interpretation is that the Doppler shift of a received signal subject to this channel will predominantly be the maximum Doppler shift. For the estimated spectrum of the generated channel, the behaviour is similar, but with a more uneven distribution. This is due to the randomness of the simulated Rayleigh fading channel.

The two different methods display the same behaviour, which is expected since the method should not significantly affect the PSD.

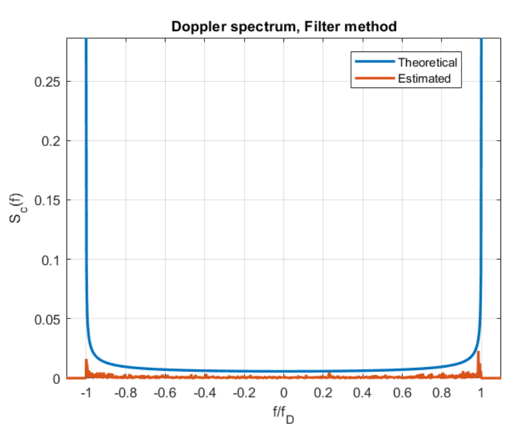

Fig. 1. Doppler PSD of a Rayleigh fading channel using the filter method.

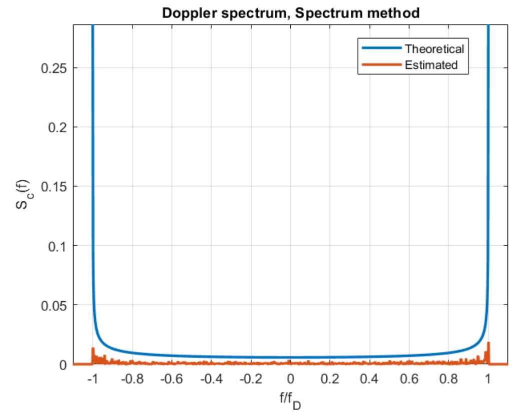

Fig. 2. Doppler PSD of a Rayleigh fading channel using the spectrum method.

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 auto-correlation function of $c(nT_s)$ and compare with the theoretical one: The Rician K-factor is described by

$$K = \frac{s^2}{2\sigma^2}, \tag{4}$$

where s^2 is the power of the LOS path and $2\sigma^2$ is the total power of the scattered components of the signal. The total power is thus $s^2 + 2\sigma^2$. We know that the difference between a Rayleigh and a Rician channel is that there is no LOS in Rayleigh, thus $s^2 = 0$. When we introduce a factor k_c to create the Rician fading channel in our model, based on the Rayleigh channel $c_{Rayleigh}$ created by using the filter or spectrum method. The factor k_c is added to the channel such that $c_{Rician} = c_{Rayleigh} + k_c$. The energy of a Rician signal is defined by

$$\begin{aligned} E_s = \sum_{n=0}^{N_s-1} |c_{Rician}(n)|^2 = \\ \sum_{n=0}^{N_s-1} |c_{Rayleigh}(n) + k_c|^2 = 2\sigma^2 + s^2. \end{aligned} \tag{5}$$

From this we can conclude that k_c^2 represents the power of the LOS component, meaning that $k_c^2 = s^2$.

The probability density function (pdf) and the cumulative distribution function (cdf) are used to describe the distribution of a random process. By the pdf and cdf plots in figure 3 to 8, we can see the effects of adding the k_c factor by using two methods. When $k_c = 0$, the Rician fading channel will reduce to the distribution of a Rayleigh fading channel. When the value of k_c is increased, the pdf of the Rician fading channel will move to the right along the x-axis and the maximum will decrease i.e. become narrower around the mean value. The rise in the cdf will accordingly move up along the x-axis. As the plots display, the simulations agree well with theory in terms of the behaviour in adding to the LOS component. The amplitude variations between the theoretical and estimated pdf may depend on characteristics of built-in functions in MATLAB, such as semi-randomness of variables and parameter definitions. There are almost no difference of the estimated pdf and cdf between the different methods, which is expected since they essentially should generate the same channel characteristics. The complexity may be different, but the estimated pdf and cdf shall be the same.

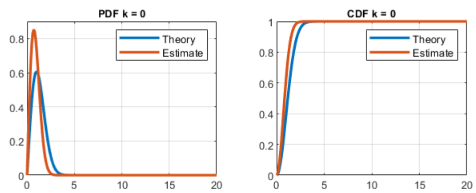

Fig. 3. PDF and CDF of the channel samples compared to theory for the filter method and kc = 0.

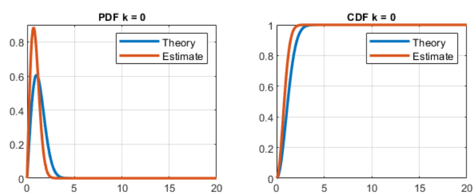

Fig. 4. PDF and CDF of the channel samples compared to theory for the spectrum method and kc = 0.

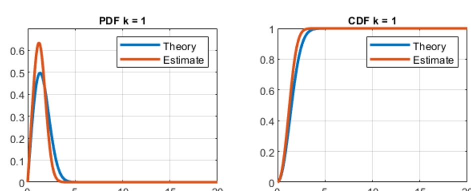

Fig. 5. PDF and CDF of the channel samples compared to theory for the filter method and kc = 1.

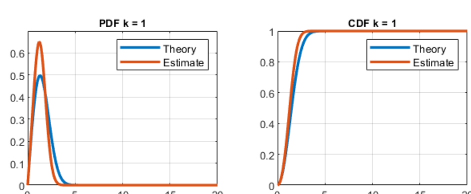

Fig. 6. PDF and CDF of the channel samples compared to theory for the spectrum method and kc = 1.

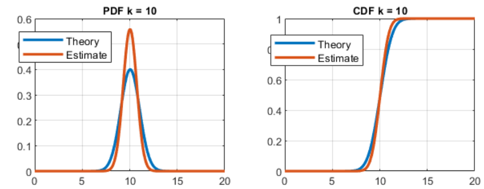

Fig. 7. PDF and CDF of the channel samples compared to theory for the filter method and kc = 10.

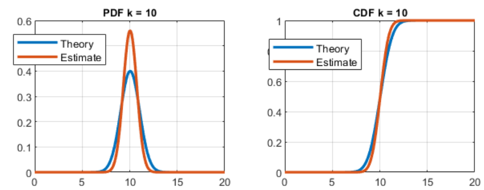

Fig. 8. PDF and CDF of the channel samples compared to theory for the spectrum method and kc = 10.

The ACF for the narrow band channel describes the correlation in time, that is how fast the channel changes, and is denoted as $A_c(\Delta t)$. The effect of adding the dominating path is presented in figure 9 to 14. According to equation 5 the power relates to k_c^2, explaining the values on the y-axis. A strong LOS component will dominate the received signal, and hence increase the ACF. When k_c increases, the estimated plot of ACF will be smoother. Considering the choice of method, the ACF plots differ slightly for the higher k_c values. The reason for this may be due to the set of random numbers.

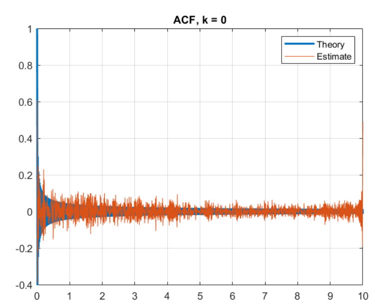

Fig. 9. Theoretical and estimated ACF for a Rayleigh fading channel (kc = 0), generated with the spectrum method.

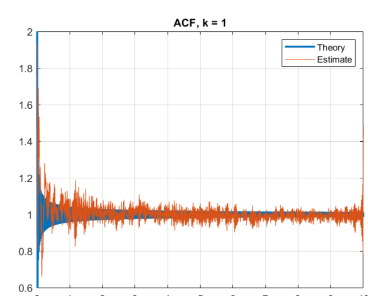

Fig. 10. Theoretical and estimated ACF for a Rician fading channel for kc = 1, generated with the spectrum method.

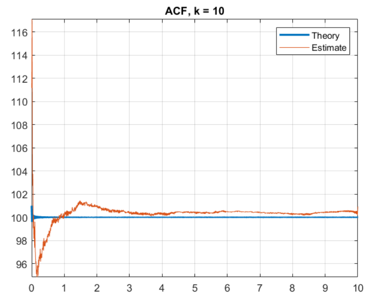

Fig. 11. Theoretical and estimated ACF for a Rician fading channel for kc = 10, generated with the spectrum method.

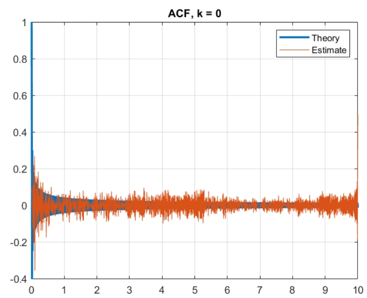

Fig. 12. Theoretical and estimated ACF for a Rayleigh fading channel (kc = 0), generated with the filter method.

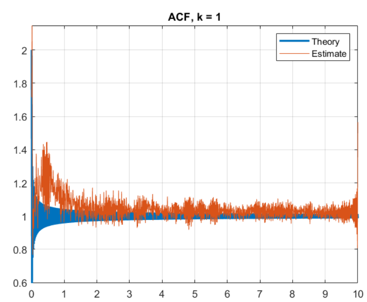

Fig. 13. Theoretical and estimated ACF for a Rician fading channel for kc = 1, generated with the filter method.

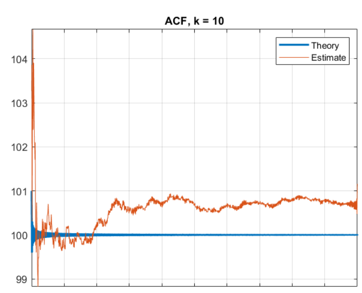

Fig. 14. Theoretical and estimated ACF for a Rician fading channel for kc = 10, generated with the filter method.

4) 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, auto-correlation functions, etc.: Generally, the filter method is computationally more complex than the spectral method, because of the required convolution. The convolution is performed such that two discrete signals are overlapped to find similarities between them. For a long signal, the data sets for processing become very large. As an indicator, for $N_s = 10^5$ the simulation time is $(0.027, 2.20)$ s and for $N_s = 10^6$ the simulation time is $(0.12, 227)$ s for the spectrum method and the filter method respectively.So when the signal is longer, the filter method will cost much more time than using the spectrum method.

When changing the different parameters, i.e. N_s, f_D and T_s, different effects are observed. As stated, the number of samples N_s has direct effect on the processing time. When N_s is small, the model will lose statistical reliability, since the number of random variables drawn from the defined distribution do not represent the entire population. Consequently, the randomness becomes questionable. The Doppler frequency f_D has a direct effect on the sampling frequency, since relative movement changes the experienced frequency at the receiving part. An increased frequency due to Doppler shift hence raises the requirement on f_s (consequently T_s).

III. Time and Frequency Varying Rician Fading Channels

A. Simulation task 2

Produce plots when $N = 300$ time samples and $M = 64$ frequency samples. Let the channels have $L = 1, 2, 3$ taps. Let the tap gains $c(nT_s)$ be Rician fading with Clarke's spectrum, with flat power delay profiles, and normalized Doppler frequency $f_D T_s = 0.1, 0.005$ and $k_c = 0, 1, 10$. According to the total 18 combinations of L, $f_D T_s$, and k_c, we have different results, which are shown in the appendix A of the report.

1) 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).: The number of taps, L, denotes how many resolvable paths that are used in the simulation model. This means we have now added a second measure in time domain; the *delay* (τ) that was considered negligible in the narrow band case.

The number of paths must be chosen such that the total time of the taps is longer than the delay spread, i.e. we need to satisfy that $LT_s \geq \tau_{DS}$, where τ_{DS} is the delay spread of the channel. For $L = 1, 2, 3$ and $T_s = 0.1$ ms, the limitation is thus $\tau_{DS} = 0.1, 0.2, 0.3$ ms respectively.

The two cases $f_D T_s$ correspond to two different velocities, that is 27 and 540 km/h determined by the Doppler effects, when $f_D Ts = 0.005$ and $f_D Ts = 0.1$ respectively. The average delay spreads can be calculated by the auto-correlation of the channel as a function of delay, such as

$$\mu_{Tm} = \frac{\int \tau A_c(\tau) d\tau}{\int A_c(\tau) d\tau}. \tag{6}$$

For the different cases, the following results are obtained.

L	$f_DT_s = 0.1$	$f_DT_s = 0.005$
1	0	0
2	14.3 μs	4.1 μs
3	41.2 μs	29.5 μs

TABLE I
AVERAGE DELAY SPREAD AND DOPPLER FREQUENCY

For $L = 1$ there is no delay (single path case), hence it is zero for both f_D. When taps are added, the delay spread should increase, as is the case. A larger Doppler frequency should also make the delay spread larger.

Compared with the different parameters of L, k_c and f_D.The magnitude of the time-varying response will be different.

In figure 15-17 $f_DT_s = 0.005$, $k_c = 0$ and L is varied. This means the Doppler spread is low and that the channel is of Rayleigh fading character. When $L = 1$ there is only one path, such that the channel is constant in frequency domain. When paths are added, the frequency dependence of the channel increases, introducing variations in both dimensions.

In figure 18-20 the same process is repeated but for a stronger LOS component, making the channel of Rician fading character. From figure 21-23 the k_c is further increased, for which it is very clear that the channel becomes much more static for a strong dominating path.

From figure 24-32 the same behaviour as previously is observed, but for a channel experiencing a larger Doppler frequency. In figure15-22, the Doppler shift is $f_D = 50$ Hz, since $f_DTs = 0.005$, while now it is $f_D = 1000$ Hz, with respect to $f_DTs = 0.1$. Comparing these two sections of figures, we can conclude that the channel is much more rapidly changing with high f_D, which is clearly visible in the time domain.

In figure 24-27 and 30, when we fix $f_DT_s = 0.1$ and $L = 1$, we can find that as k_c increases, the maximum value of the channel response will decrease, but there will be larger distribution of high values meaning that the yellow part will be more in the figures. This can be explained from the pdf of the channel. According to what we have achieved in task 1, as k_c increases, Rician distribution will be 'fatter' in time x axis but smaller in y axis, which is the same.

IV. CONTRIBUTION FROM GROUP MEMBERS

Lise Aabel: I have spent approximately 27 hours on writing the report and 40 hours writing the MATLAB code presented for both tasks.

Yaxi Xie : I have spent approximately 36 hours on trying to write the Matlab code. After asking the teacher assistant, I have spent 4 hours in troubleshooting and modifying my code. Then I talked about some questions in the code with the group members. For the report, I have spent about 12 hours in plotting figures and answering the questions in the simulation part in a basic version, which has been modified by Lise Aabel later.

Kashif Shabir: He has spent two hours starting on the report.

APPENDIX A

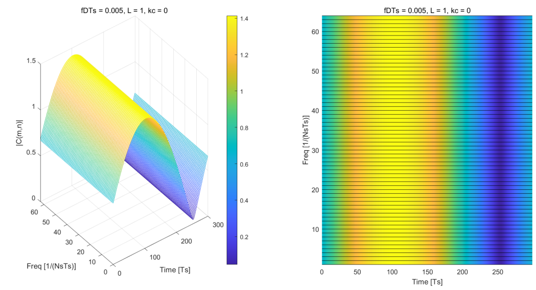

Fig. 15. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 1 kc = 0

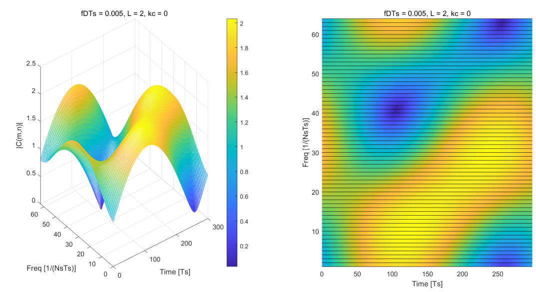

Fig. 16. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 2 kc = 0.

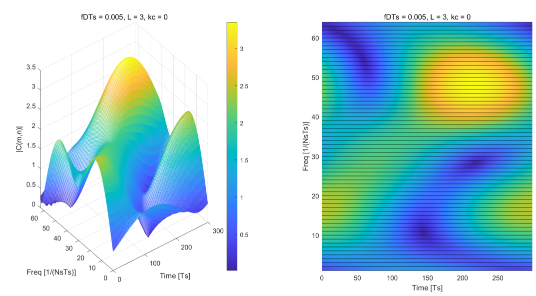

Fig. 17. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 3 kc = 0.

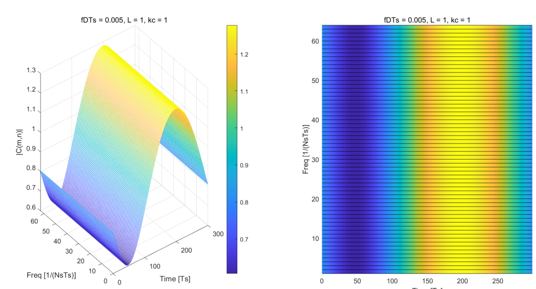

Fig. 18. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 1 kc = 1.

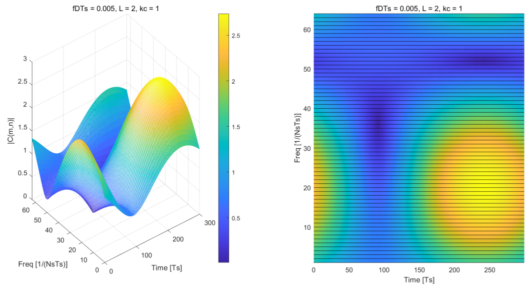

Fig. 19. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 2 kc = 1.

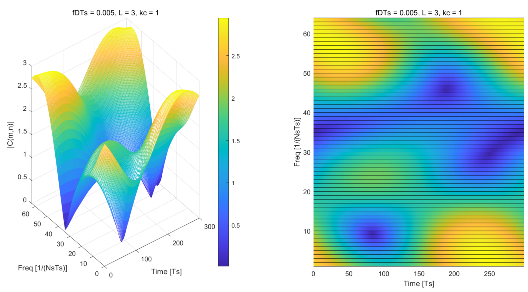

Fig. 20. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 3 kc = 1.

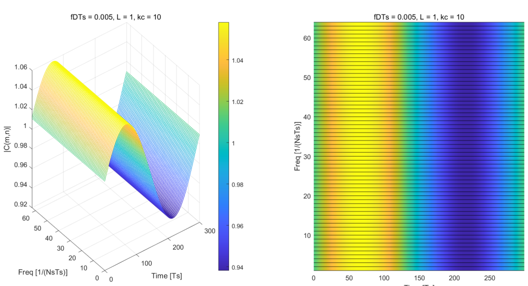

Fig. 21. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 1 kc = 10.

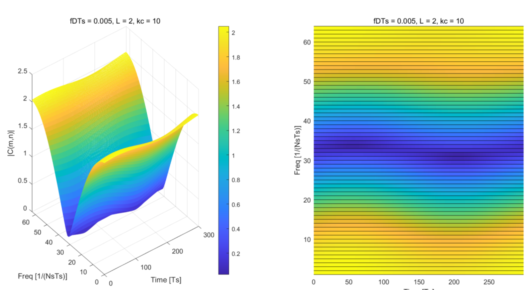

Fig. 22. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 2 kc = 10.

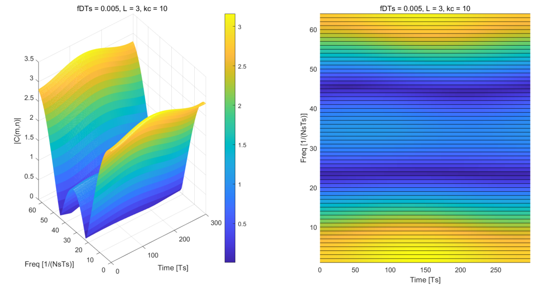

Fig. 23. The magnitude time and frequency-varying channel response for fDTs = 0.005 L = 3 kc = 10.

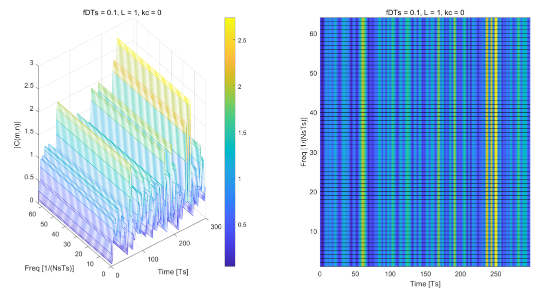

Fig. 24. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 1 kc = 0

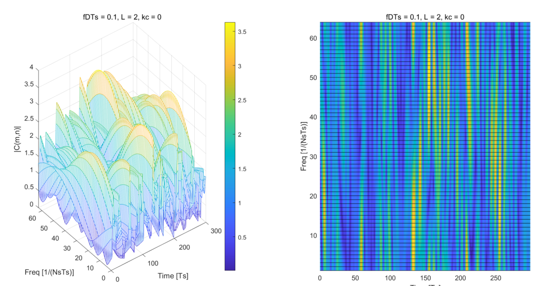

Fig. 25. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 2 kc = 0.

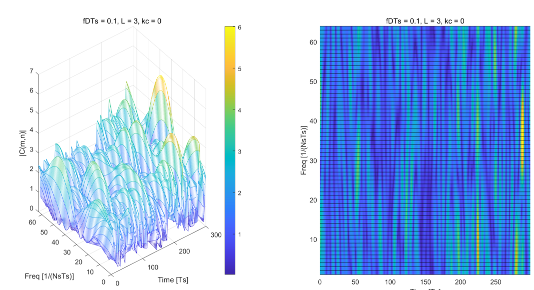

Fig. 26. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 3 kc = 0.

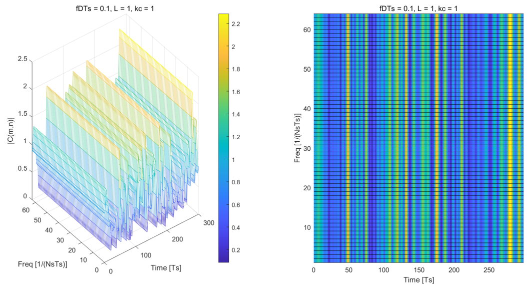

Fig. 27. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 1 kc = 1.

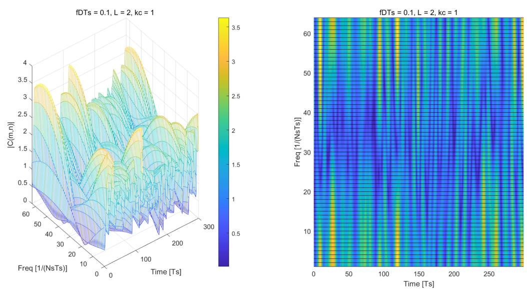

Fig. 28. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 2 kc = 1.

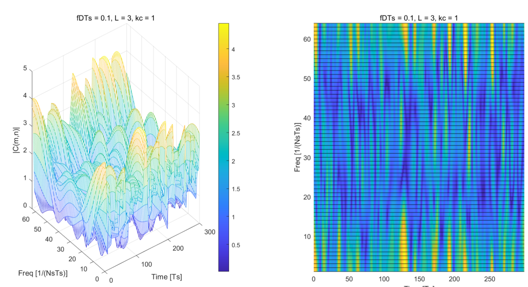

Fig. 29. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 3 kc = 1.

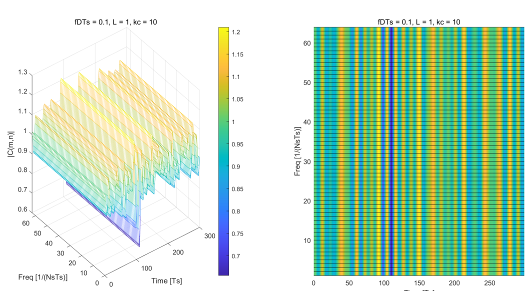

Fig. 30. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 1 kc = 10.

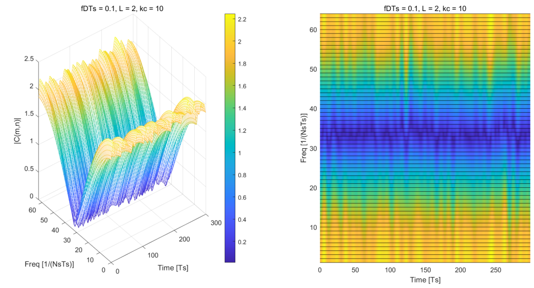

Fig. 31. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 2 kc = 10.

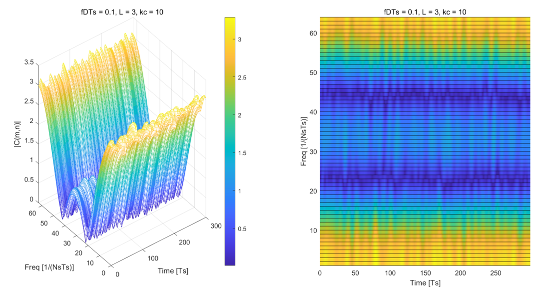

Fig. 32. The magnitude time and frequency-varying channel response for fDTs = 0.1 L = 3 kc = 10.

Appendix B

```
 clear all, close all

% Parameter definition
par.fc = 2e9; % Carrier [GHz]
par.lambda = 3e8/par.fc; % Wavelength [m]
par.v = 30e3/3600; % Relative velocity [m/s]
par.Ts = 0.1e-3; % Sampling rate [s]
par.fD = par.v/par.lambda; % Doppler spread [Hz], no aliasing; BW=10k > 2fD=110
par.method = 'Filter'; % Choose method {'Filter', 'Spectrum'}
par.kc = 10; % Rician factor for dominating path, set to k = 0 for Rayleigh
par.Ns = 1e5; % Number of samples for the channel
par.N = par.Ns*2+1; % Number of samples for the filter g(t)
par.fs = 1/par.Ts; % Sampling frequency

tic;
switch par.method

    case 'Filter'
        nTs = -par.N*par.Ts:par.Ts:par.N*par.Ts; % Time vector for filter

        % Define filter g(t)
        gt = besselj(1/4,2*pi*par.fD.*abs(nTs))./nthroot(abs(nTs),4); % For t~=0
        gt(par.N+1) = nthroot((pi*par.fD),4)/gamma(5/4); % For t=0
        gt = gt/sqrt(sum(abs(gt).^2)); % Normalise to unit energy
        Eg = sum(abs(gt).^2); % Check Energy of g(t)

        % Plot g(t) in time
        figure, plot(nTs.*1e3,gt);
        xlabel('Time [ms]')
        ylabel('g(t)')
        title('Time domain filter')

        % Generate Ns random complex distribution for the channel
        x = sqrt(1/2).*(randn(1,par.Ns)+1j*randn(1,par.Ns));
        var_x = var(x,[],2); % Check unit variance

        % Generate channel c(t)
        c = conv(x,gt,'same') + par.kc;
%          c = c./sqrt(mean(abs(c).^2)); % Renormalize to unit energy

        nTs = -par.Ns*par.Ts:par.Ts:par.Ns*par.Ts; % Time vector for the channel

    case 'Spectrum'
        f = -par.fD:par.fs/par.Ns:par.fD; % Defining the frequency interval
        nTs = -par.Ns*par.Ts:par.Ts:par.Ns*par.Ts; % Time vector for sampling

        % Calculate Sf and Gf
        Sf = 1./(pi*par.fD.*sqrt(1-(f./par.fD).^2));
        Sf(1) = Sf(end); % Get rid of the 'Inf' at -fD
        Gf = sqrt(Sf);

        % The periodic extension of G(f) for k = 0:1:par.Ns-1
        Gfp = [Gf(length(Gf)/2:end) zeros(1,par.Ns-length(Gf)) Gf(1:length(Gf)/2-1)];
        var_Gfp = var(Gfp,[],2);

        % Random complex distribution
        a = sqrt(1/2/var_Gfp);
        X = a*(randn(1,par.Ns)+1j*randn(1,par.Ns)); % Random complex
        var_X = var(X,[],2); % Check variance

        C = Gfp.*X;
        c = sqrt(par.Ns)*ifft(C,par.Ns,2);

        c = c + par.kc; % Add Rician k-factor
%          c = c./sqrt(mean(abs(c).^2,2)); % Renormalize to unit energy

end
time = toc;

% Plot channel distribution
```

```matlab
figure, histogram(abs(c),100);
title('Distribution of c(t), Narrow band')

% Calculate expected values of c
E_Ec = mean(abs(c).^2,2); % E(energy) = 1
E_c = mean(real(c),2); % E(c) = 0 at Rayleigh

% Theoretical and estimated PSD
fD_vec = -par.fD:par.fs/par.Ns:par.fD;
Sc = 1./(pi*par.fD.*sqrt(1-(fD_vec./par.fD).^2));
Sc(1) = Sc(end);
% Sc_est = pwelch(c,[],[],[],par.fs); % Estimate of PSD
Sc_est = abs(sqrt(1/par.Ns)*fft(c)).^2/par.Ns;
figure
h1 = plot(fD_vec./par.fD,Sc,'LineWidth',2,'DisplayName','Theoretical');
grid on
hold on
h2 = plot((-par.Ns/2:par.Ns/2-1)*par.fs/par.Ns/par.fD,fftshift(Sc_est),'LineWidth',2,'DisplayName','Estimated');
axis([-1.1 1.1 0 max(get(h1,'Ydata'))])
title(['Doppler spectrum, ' par.method ' method'])
xlabel('f/f_D'), ylabel('S_c(f)')
grid on
legend

% PDF and CDF, theoretical and estimates of |c(t)|
x = 0:0.1:20; % x-axis def
dist = makedist('Rician','s',par.kc,'sigma',1); % Theoretical distribution
PDF = pdf(dist,x);
CDF = cdf(dist,x);
dist_est = fitdist(abs(c).','Rician'); % Estimated distribution
PDF_est = pdf(dist_est,x);
CDF_est = cdf(dist_est,x);
figure
subplot(1,2,1)
plot(x,PDF,'LineWidth',2,'DisplayName','Theory'), hold on
plot(x,PDF_est,'LineWidth',2,'DisplayName','Estimate')
title(['PDF k = ' num2str(par.kc)])
legend, grid on
subplot(1,2,2)
plot(x,CDF,'LineWidth',2,'DisplayName','Theory'), hold on
plot(x,CDF_est,'LineWidth',2,'DisplayName','Estimate')
title(['CDF k = ' num2str(par.kc)])
legend, grid on


% Theoretical and estimated ACF
Ac = besselj(0,2*pi*par.fD.*nTs) + par.kc^2; % Theoretical Ac(dt)
[Ac_est,lags] = xcorr(real(c),'unbiased'); % Estimated Ac(dt)
lags = lags(lags>=0);
Ac_est = Ac_est(lags>=0);
% Ac_est = Ac_est./max(Ac_est); % Normalize to max correlation = 1
figure
plot(nTs(par.Ns:end),Ac(par.Ns:end),'LineWidth',2,'DisplayName','Theory')
hold on
plot(lags*par.Ts,Ac_est,'DisplayName','Estimate')
title(['ACF, k = ' num2str(par.kc)])
axis tight, grid on
legend
```

APPENDIX C

```matlab
% L no taps
clear all, close all

% Parameter definition
par.fc = 2e9; % Carrier [GHz]
par.lambda = 3e8/par.fc; % Wavelength [m]
par.Ts = 0.1e-3; % Sampling rate [s]
par.Ns = 300; % Number of samples for the channel
par.fs = 1/par.Ts; % Sampling frequency
par.M = 64; % Zero-padding (oversampling in delay domain)
par.L = 2; % {1 2 3} Number of taps
par.kc = 0; % {0 1 10} Rician factor for dominating path, set k=0 for Rayleigh
par.fDTs = 0.005; % {0.1 0.005} For Clarke's spectrum, 0.4/fD = decorrelation time
par.fD = par.fDTs/par.Ts; % Doppler spread [Hz]

% Preallocation of c
cMN = zeros(par.M,par.Ns);

% ---------------Channel generation through Spectrum method----------------
f = -par.fD:par.fs/par.Ns:par.fD; % Defining the frequency interval
nTs = -par.Ns*par.Ts:par.Ts:par.Ns*par.Ts; % Time vector for sampling

% Generate G(f)
Sf = 1./(pi*par.fD.*sqrt(1-(f./par.fD).^2));
Sf = Sf(2:end-1); % Get rid of the 'Inf' at -fD and fD (not valid)
Gf = sqrt(Sf);

% The periodic extension of G(f) for k = 0:1:par.Ns-1
Gfp = [Gf(floor(length(Gf)/2):end) zeros(1,par.Ns-length(Gf)) Gf(1:floor(length(Gf)/2)-1)];
var_Gfp = var(Gfp,[],2);

% Random complex distribution
a = sqrt(1/2/var_Gfp); % Choose a such that the E[|c(t)|] = 1 ("eliminate the variance of G")
X = a*(randn(par.L,par.Ns)+1j*randn(par.L,par.Ns));
var_X = var(X,[],2); % Check unit variance

Cl = Gfp.*X; % Create channel in f domain
cl = sqrt(par.Ns)*ifft(Cl,[],2); % convert to t domain

cl = cl + par.kc; % Add Rician LOS factor
% cl = cl./sqrt(mean(abs(cl).^2,2)); % Renormalize to unit expected energy
E_cl = mean(abs(cl).^2,2);

cMN(1:par.L,:) = cl; % Zero padding
CMN = fft(cMN,[],1); % FT in delay domain
% -------------------------------------------------------------------------
tau = 0:par.Ts:(par.L-1)*par.Ts;
[Ac_est,lags] = xcorr(real(cMN(:,1)),'unbiased'); % Estimate Ac(tau)
Ac_est = Ac_est(lags >= 0); % Get the tau >= 0 part
DS_avg = sum(tau.*(abs(Ac_est(1:par.L)).^2)')/sum((abs(Ac_est(1:par.L)).^2)'); % avg delay spread in s
B_coh = 1/DS_avg;
T_coh = 1/(2*par.fD);

v = par.fD*par.lambda*3.6; % Relative velocity in kph

% figure
% mesh(0:par.Ns-1,0:par.M-1,abs(CMN))
% xlabel('Time [Ts]')
% ylabel('Freq [1/(NsTs)]')
% zlabel('|C(m,n)|')
% title(['fDTs = ' num2str(par.fDTs) ', L = ' num2str(par.L) ', kc = ' num2str(par.kc)])
figure
surf(0:par.Ns-1,1:par.M,abs(CMN), 'MeshStyle', 'row'), view(2), axis tight
xlabel('Time [Ts]')
ylabel('Freq [1/(NsTs)]')
zlabel('|C(m,n)|')
title(['fDTs = ' num2str(par.fDTs) ', L = ' num2str(par.L) ', kc = ' num2str(par.kc)])
```