Demo: Propagation and Rate Modeling

In this demo, we will illustrate how to perform simple path loss and rate models estimates in MATLAB. In doing this demo, you will learn to:

- Estimate the rate from a path loss using a free space model, as well as parameters such as the transmit power, noise figure, bandwidth, ...
- Use MATLAB atmospheric attenution models
- Create a simple 'drop' of users in a simulation
- Generate random path losses to the users using a 3GPP statistical model
- Compute the SNR and rate CDFs for the users

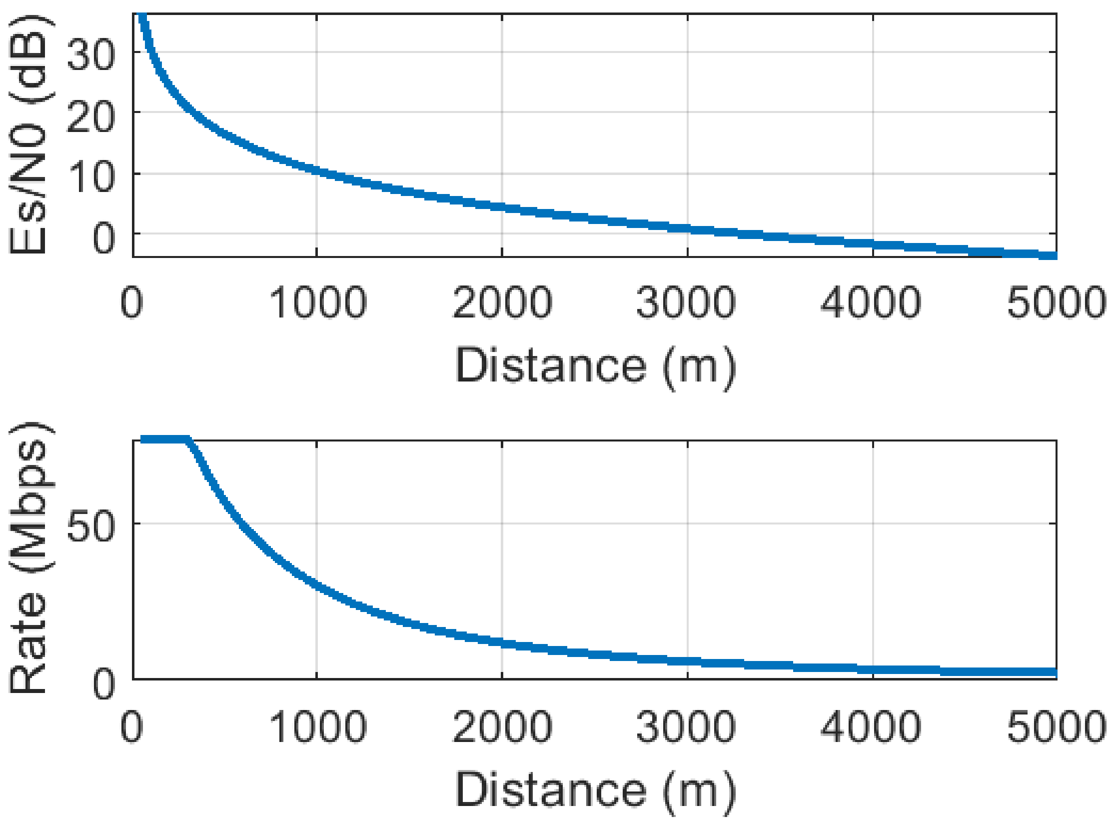
Contents

Estimate rate with free space propagation

We begin by showing how to estimate to the SNR and rate in a simple free-space scenario.

```
% Parameters.  We make some simple assumptions
B = 20e6;     % bandwidth
fc = 2.3e9;   % carrier
NF = 6;       % noise figure (dB)
snrLoss = 6; % loss from Shannon capacity
maxSE = 4.8; % max spectral efficiency
bwLoss = 0.2;    % bandwidth overhead
Ptx = 15;  % Power in dBm
dist = linspace(50,5000,100)';  % distance

% Compute the FS path loss
vp = physconst('lightspeed');  % speed of light
lambda = vp/fc;    % wavelength
pl = fspl(dist, lambda);

% Compute SNR.  For simplicity, we will ignore the antenna gain
kT = -174;
EsN0 = Ptx - pl - kT - NF - 10*log10(B);

% Compute rate
snrEff = 10.^(0.1*(EsN0-snrLoss));
rateMbps = B*(1-bwLoss)*min(log2(1 + snrEff), maxSE)/1e6;
```

```
% Plot results
subplot(2,1,1);
plot(dist,EsN0,'Linewidth', 3);
grid on;
set(gca,'Fontsize',16);
xlabel('Distance (m)');
ylabel('Es/N0 (dB)');
xlim([0, max(dist)]);

subplot(2,1,2);
plot(dist,rateMbps,'Linewidth', 3);
set(gca,'Fontsize',16);
xlabel('Distance (m)');
ylabel('Rate (Mbps)');
grid on;
xlim([0, max(dist)]);
```

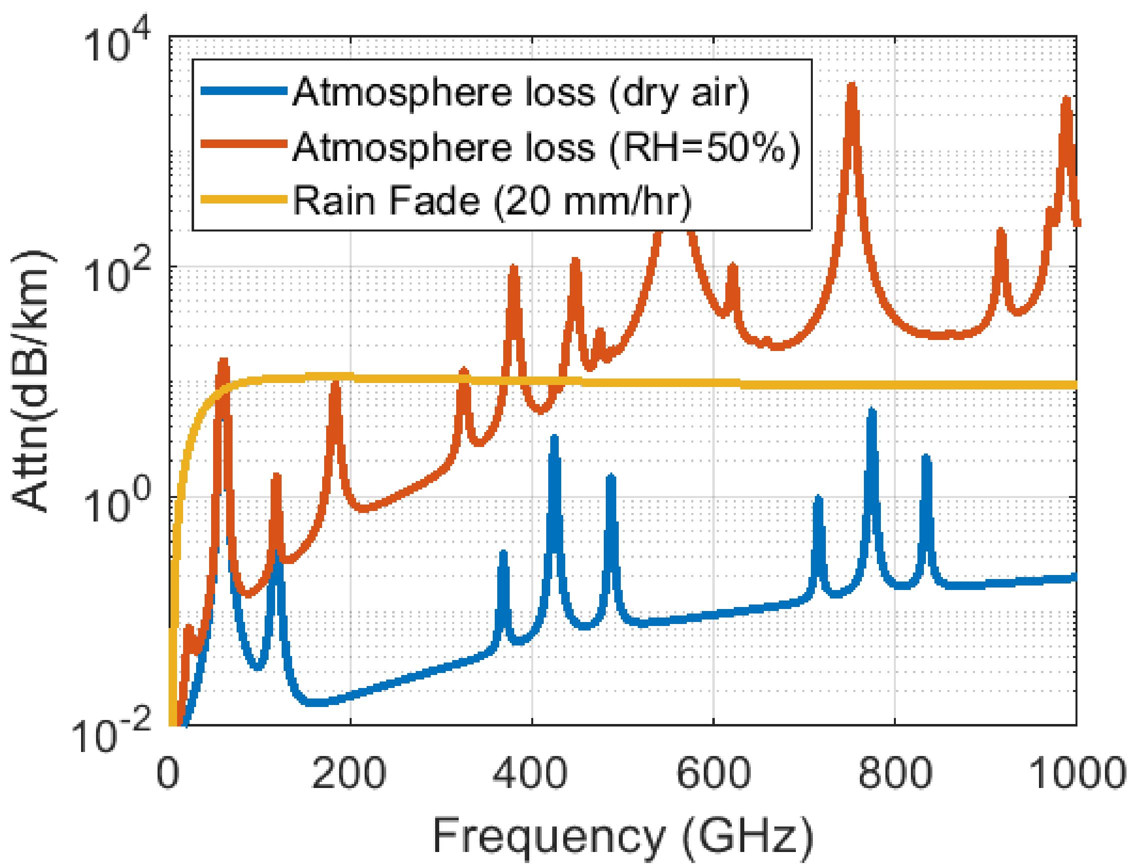


MATLAB attenuation models

Atmospheric attenuation due to water absorption is critical to model for mmWave links at distance. Matlab has many in-built excellent models for attenuation.

```
% Frequencies to test
freq = linspace(1,1000,1000)'*1e9;
range = 1000;  % Compute attenuation at 1 km
rr = 20;       % Rain rate
```

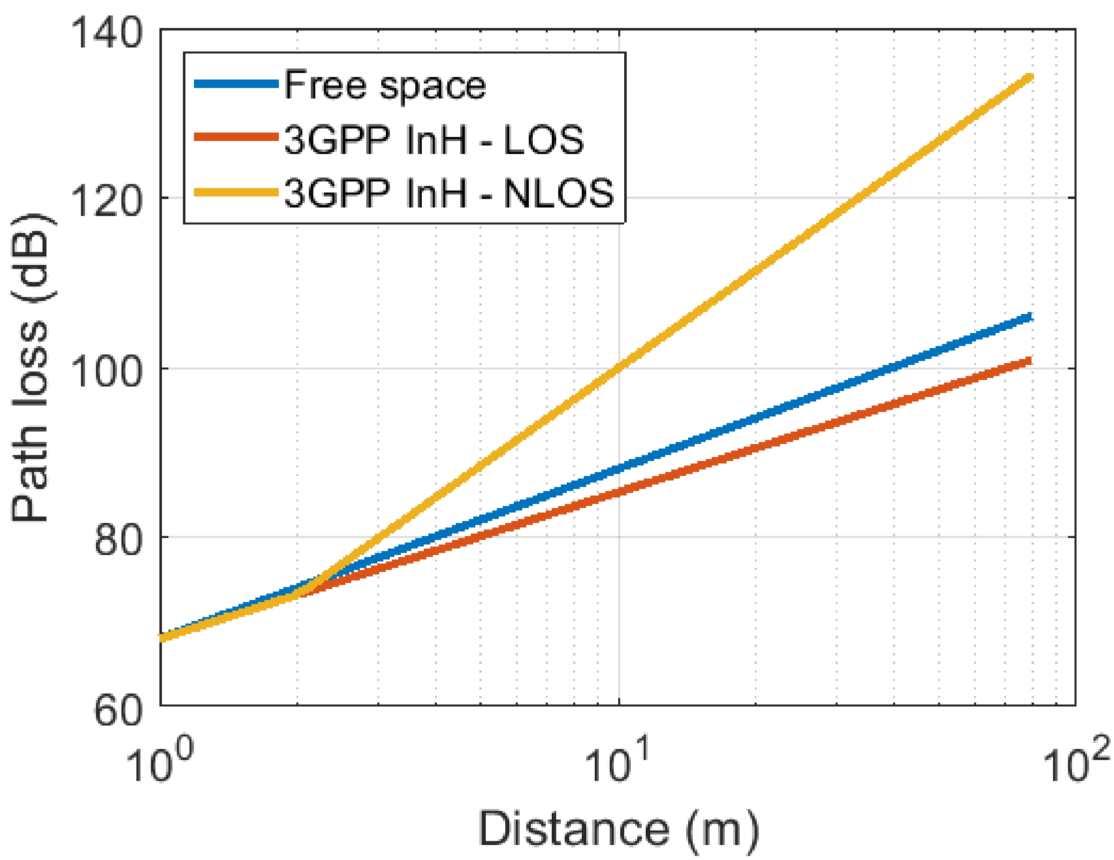
```
T = 15;          % temperature in C
P = 101300.0;    % atmospheric pressure
Wsat = 4.8;      % vapor density at saturation (g/m^3)
RH = 0.5;        % relative humidity
W = RH*Wsat;     % vapor density

% Compute attenuations
attn_dry = gaspl(range,freq,T,P,0)';
attn_humid = gaspl(range,freq,T,P,W)';
attn_rain = rainpl(range, freq, rr)';

subplot(1,1,1);
semilogy(freq/1e9, [attn_dry attn_humid attn_rain], 'Linewidth', 3);
grid on;
xlabel('Frequency (GHz)');
ylabel('Attn(dB/km)');
ylim([0.01, 1e4]);
set(gca,'Fontsize', 16);
legend(...
    'Atmosphere loss (dry air)', ...
    'Atmosphere loss (RH=50%)', ...
    'Rain Fade (20 mm/hr)', ...
    'Location', 'NorthWest');
```



3GPP path models

3GPP has extensive sets of statistical models for a wide range of scenarios. In this demo, we will look at the 3GPP indoor office model from the specification 3GPP TR 38.901. First, we will plot the median path loss for NLOS and LOS cases.

```
% Parameters
fc = 60e9;  % Frequency
vp = physconst('lightspeed');  % speed of light
lambda = vp/fc;   % wavelength

% Compute LOS and NLOS path loss
dist = linspace(1,80,500)';
pllos = 32.4 + 17.3*log10(dist) + 20*log10(fc/1e9);
plnlos = 17.3 + 38.3*log10(dist) + 24.9*log10(fc/1e9);
plnlos = max(pllos, plnlos);
plfs = fspl(dist,lambda);

semilogx(dist, [plfs, pllos, plnlos], 'Linewidth', 3);
grid on;
legend(...
    'Free space', '3GPP InH - LOS', '3GPP InH - NLOS', ...
    'Location', 'NorthWest');
set(gca, 'Fontsize', 16);
xlabel('Distance (m)');
ylabel('Path loss (dB)');
```

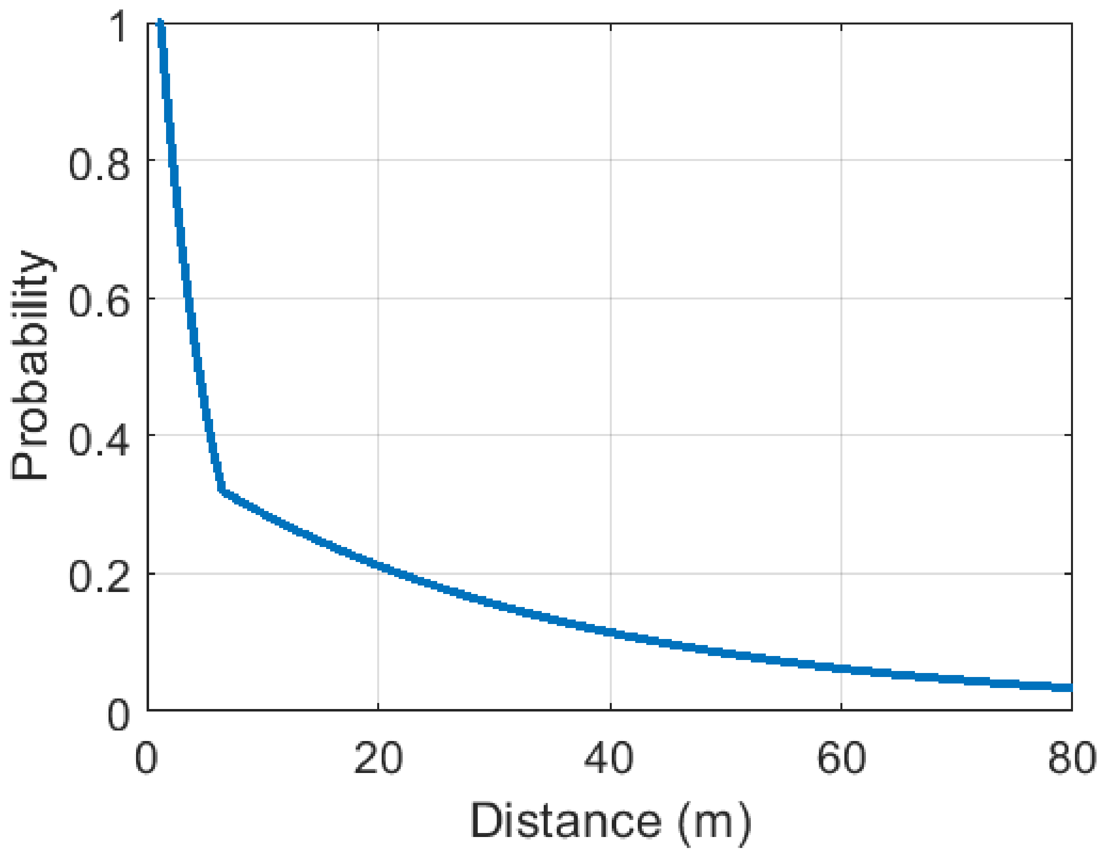


Plot the probability of LOS

In the 3GPP model, a link is in LOS with some probability that decreases with distance. We plot this probability

```
plos1 = min( exp(-(dist-1.2)/4.7), 1);
plos  = max(0.32*exp(-(dist-6.5)/32.6), plos1);
```

```
plot(dist, plos, 'Linewidth', 3);
grid on;
set(gca, 'Fontsize', 16);
xlabel('Distance (m)');
ylabel('Probability');
```

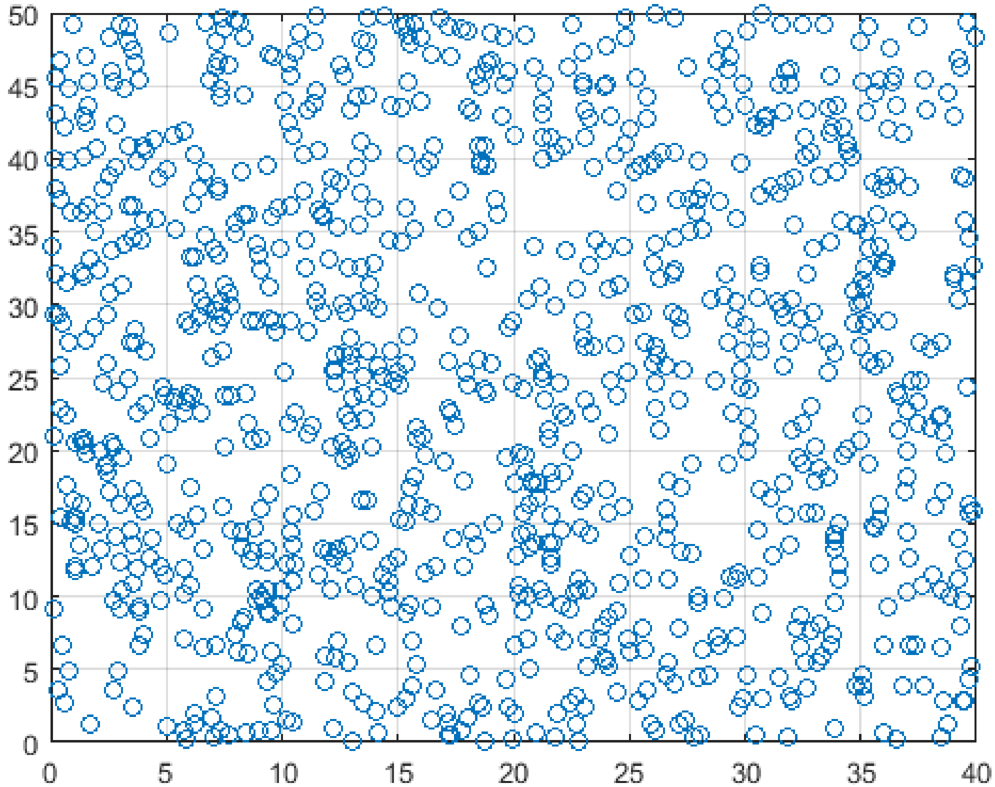


Simulating a simple indoor system: Dropping users

In many analyses of communication systems, we want to estimate the distribution of key parameters such as rate, SNR or delay under some statistical model for the locations of users, access points and path loss. Organizations like 3GPP have very elaborate procedures for running these simulations. But, here, to make things simple, we will do a very simple simulation where we assume a single RX is randomly distributed in some square region with a TX located at the origin.

The first part of such a simulation is to generate random 'drops' of the user.

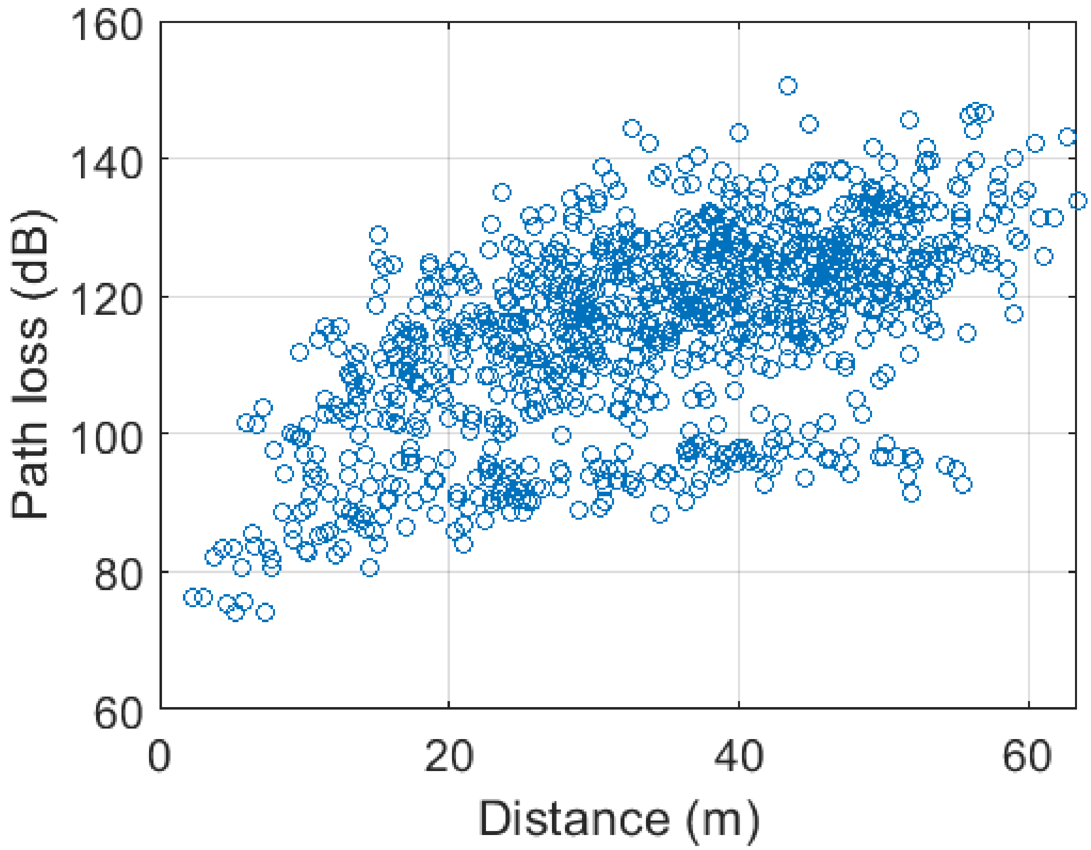
```
% Parameters
len = 40;  % length of region in m
wid = 50;  % width in m
nx = 1000;  % number of random points

% Generate random points in a square
x = rand(nx,2).*[len wid];

% Plot the random points
plot(x(:,1), x(:,2), 'o');
grid on;
```



Generate random path losses

We will now generate random path losses using the 3GPP model.

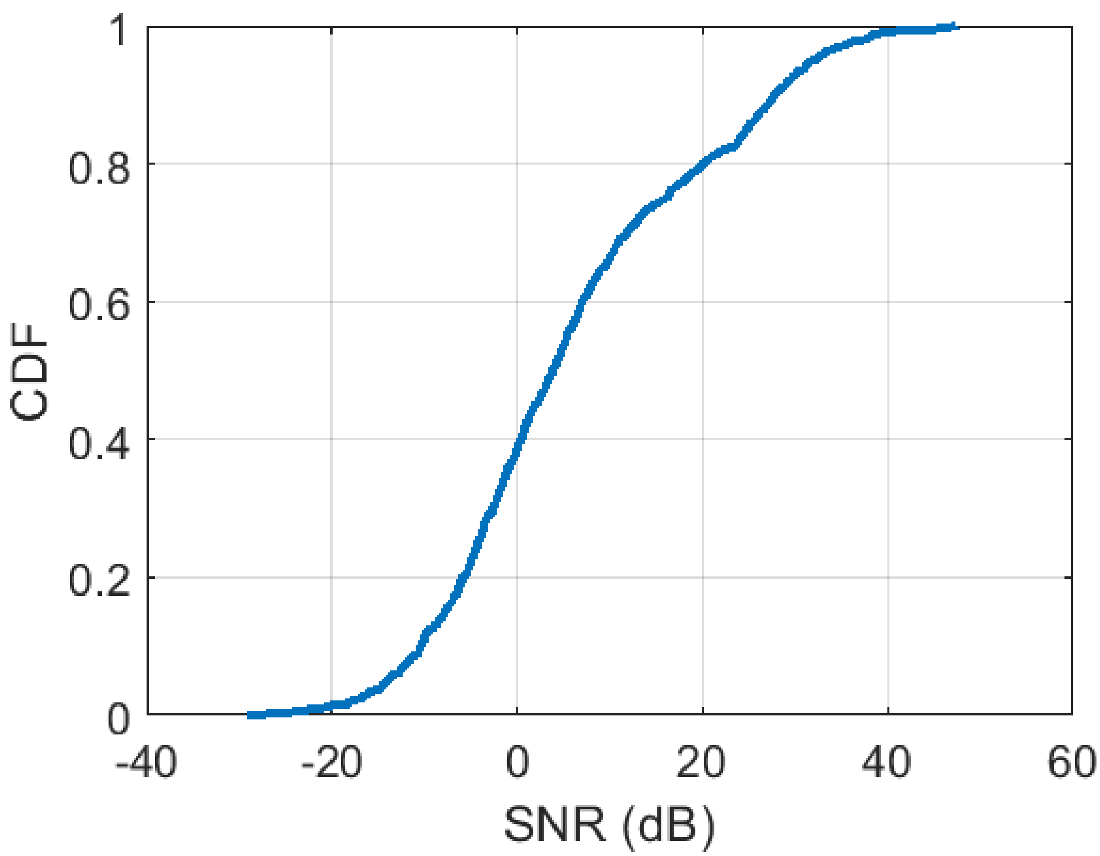
```
% We will make some simple assumptions for a wifi-like system
fc = 60e9;

% Compute the distances
dh = 1;  % Distances in height
dist = sqrt(sum(x.^2,2) + dh^2);

% We next generate random path losses to each
pl = pathLoss3GPPInH(dist, fc);

% Plot a scatter plot of the PL vs. distance
plot(dist, pl, 'o');
grid on;
xlabel('Distance (m)');
ylabel('Path loss (dB)');
set(gca, 'Fontsize', 16);
```



Compute the SNR CDF

We next compute the SNR at each point. We will assume powers of an 802.11ad-like system. Note the high directivity in our assumptions. We will show later with beamforming how such directivity can be obtained.

```
% Parameters
ptx = 20;      % transmit power
bw = 1.76e9;   % sample rate
nf = 6;        % noise figure
kt = -174;     % thermal noise
gaintx = 16;   % antenna gain
gainrx = 10;

% Compute SNR
snr = ptx - pl - nf - 10*log10(bw) - kt + gainrx + gaintx;

% Plot the SNR CDF
p = (1:nx)/nx;
plot(sort(snr), p,'Linewidth', 3);
grid on;
xlabel('SNR (dB)');
ylabel('CDF');
set(gca, 'Fontsize', 16);
```



Compute the rate CDF

Finally we compute the rate based on some simple simulatins

```
snrLoss = 6;
bwLoss = 0.2;
maxSE = 4.8;
rate = bw*(1-bwLoss)*min(log2(1 + 10.^(0.1*(snr-snrLoss))), maxSE);
rate = rate/1e6;

p = (1:nx)/nx;
semilogx(sort(rate), p,'Linewidth', 3);
grid on;
xlabel('Rate (Mbps)');
ylabel('CDF');
set(gca, 'Fontsize', 16);

function pl = pathLoss3GPPInH(dist,fc)
    % pathLoss3GPPInH:  Generates random path loss
    %
    % Samples the path loss using the 3GPP-InH model

    % Compute the median path losses for LOS and NLOS
    pllos = 32.4 + 17.3*log10(dist) + 20*log10(fc/1e9);
    plnlos = 17.3 + 38.3*log10(dist) + 24.9*log10(fc/1e9);

    % Add shadowing
    w = randn(size(dist));
    pllos = pllos + 3*w;
```

```
    plnlos = plnlos + 8.03*w;

    % Compute probability of being LOS or NLOS
    plos = min( exp(-(dist-1.2)/4.7), 1);
    plos = max( 0.32*exp(-(dist-6.5)/32.6), plos);

    % Select randomly between LOS and NLOS path loss
    u = (rand(size(dist)) < plos);
    pl = u.*pllos + (1-u).*plnlos;

end
```