# Experiment-4

**Aim:** To implement concept code for behaviour of power with distance and path loss exponent for Cellular Mobile Communication System.

**Activities:**

1. To implement and observe code for the simulation for the behaviour of power with distance and path loss exponent.
2. Consider transmitter is producing 100 W of power, it is applied to a unity gain antenna with 900 Mhz carrier frequency, find the receive power in dBm for a free space distance (d0) of 100 m and path loss exponent as n=1.
3. The same power (100 W) is applied to a unity gain antenna with a 900 Mhz carrier frequency, calculate the received power in dBm with programming with input parameter of a free space distance for 100 m, 200 m, 500 m, 1 km, 5 km, and 10 km. Also plot the distance Vs Received power.

Data to be consider, Transmitted Power (Pt ) = 100, initial distance (d0) = 100, Path loss exponent (n) = 2, 3, 4. Transmitter antenna gain (Gt , Gr) = 1, frequency of transmission (f) = 900 Mhz.

**Theory:**

In its simplest form, the Friis transmission equation is as follows. Given two antennas, the ratio of power available at the input of the receiving antenna, Pt , to output power to the transmitting antenna, Pt, is given by

Pr=PtGtGr ( λ/ 4 π R )^2 /L

Where Gt and Gr are the antenna gains (with respect to an isotropic radiator) of the transmitting and receiving antennas respectively, lambda is the wavelength, and R is the distance between the antennas. The inverse of the third factor is the so-called free space path loss. To use the equation as written, the antenna gain may not be in units of decibels, and the wavelength and distance units must be the same. If the gain has units of dB, the equation is slightly modified to:

log(Pr) =log(Pt \*Gt\* Gr ( λ/ 4 π R )^2) / L

log(Pr) =log(Pt)+log(Gt)+log(Gr)+log(( λ/ 4 π R )^2)+log(1/L)

Pr=Pt+Gt+Gr+20log( λ/ 4 π R ) -L

(Gain has units of dB, and power has units of dBm or dBW)

**Algorithm:**

1. **Received Power**

**Step1:** Enter the frequency of operation, transmitted power Pt, Gains of the Transmitter and receiver Antenna (Gt and Gr), Ref.

Distance R expressed in d0 and the path loss Exponent (n1 to n4).

**Step2:** Calculate received power

# Pr = (Pt\*Gt\*Gr\*lambda^2) / ((4\*pi\*d0)^2\*L)

# Pr\_dBm = 10\*log10(Pr / 10^-3)

1. **Received Power at different distance**

**Step1:** Enter the frequency of operation, transmitted power Pt, Gains of the Transmitter and recevier Antenna (Gt And Gr), Ref. Distance d0 and the path loss Exponent(n1 to n4).

**Step2:** Initialize **R,** *i.e.* [100,200,300,400,800,1200,1600,1800,2000]

**Step3:** Initialize **L,** *i.e.* [1,2,3,4].

**Step4:** Calculate received power at different distances

# Pr = (Pt\*Gt\*Gr\*lambda^2) / ((4\*pi\*d0)^2\*L)

# Pr\_dBm = 10\*log10(Pr / 10^-3)

1. **Power received at difference distance with reference distance point**

**Step1:** Enter the received power from(A) arPr, distance from antenna as (D), and path loss(L).

**Step2:** Initialize **D,** *i.e.* [100,200,300,400,800,1200,1600,1800,2000]

**Step3:** Initialize **L** *i.e.*[1,2,3,4].

Pr = Pr\*(d0/D)^L;

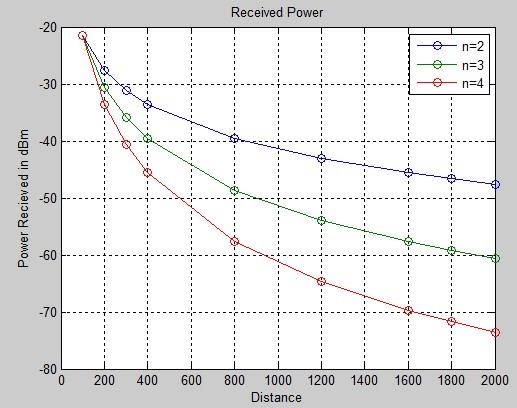
**Output:**

**(A)Received Power**

Pr = 7.0362e-006

Pr\_dB = -21.5266

**(B)Received Power at different distance**



**Activity A:**

**Simulation of Power with Distance and Path Loss Exponent**

**Code:**

# Step 1: Import necessary libraries

import numpy as np

import matplotlib.pyplot as plt

# Step 2: Define the parameters

frequency = 900e6 # 900 MHz in Hz

transmitted\_power = 100 #100 transmitter\_antenna\_gain = 1 receiver\_antenna\_gain = 1 reference\_distance = 100 # 100 m path\_loss\_exponent = 1

# Calculate wavelength (lambda) in meters wavelength = 3e8 / frequency

# Step 3: Calculate received power in dBm

received\_power = (transmitted\_power \* transmitter\_antenna\_gain \* receiver\_antenna\_gain \* wavelength\*\*2)

/ \

((4 \* np.pi \* reference\_distance)\*\*2 \* path\_loss\_exponent)

received\_power\_dBm = 10 \* np.log10(received\_power / 1e-3) # Convert to dBm

# Step 4: Print the result

print(f"Received Power (W): {received\_power}")

print(f"Received Power (dBm): {received\_power\_dBm}")

## **Output:**

Received Power (W): 7.036193308495678e-06

Received Power (dBm): -21.526622374835178

**Activity B:**

**Received Power at Different Distances and Path Loss Exponents**

**Code:**

# Step 1: Import necessary libraries (already imported in Activity A)

# Step 2: Define the parameters

distances = [100, 200, 500, 1000, 5000, 10000] # Distances in meters path\_loss\_exponents = [2, 3, 4]

# Create lists to store results

received\_powers\_dBm = []

# Step 3: Calculate received power at different distances and path loss exponentsg for distance in distances:

for path\_loss\_exponent in path\_loss\_exponents:

received\_power = (transmitted\_power \* transmitter\_antenna\_gain \* receiver\_antenna\_gain \* wavelength\*\*2) / \

((4 \* np.pi \* reference\_distance)\*\*2 \* path\_loss\_exponent) received\_power \*= (reference\_distance / distance)\*\*path\_loss\_exponent received\_power\_dBm = 10 \* np.log10(received\_power / 1e-3) # Convert to dBm received\_powers\_dBm.append(received\_power\_dBm)

# Step 4: Print the results and plot the distance vs. received power for i, distance in enumerate(distances): for j, path\_loss\_exponent in enumerate(path\_loss\_exponents):

index = i \* len(path\_loss\_exponents) + j print(f"Distance: {distance} m, Path Loss Exponent: {path\_loss\_exponent}") print(f"Received Power (dBm): {received\_powers\_dBm[index]}") print("---")

# Plot distance vs. received power for different path loss exponents for j, path\_loss\_exponent in enumerate(path\_loss\_exponents):

plt.plot(distances, [received\_powers\_dBm[i \* len(path\_loss\_exponents) + j] for i in range(len(distances))], label=f'n={path\_loss\_exponent}')

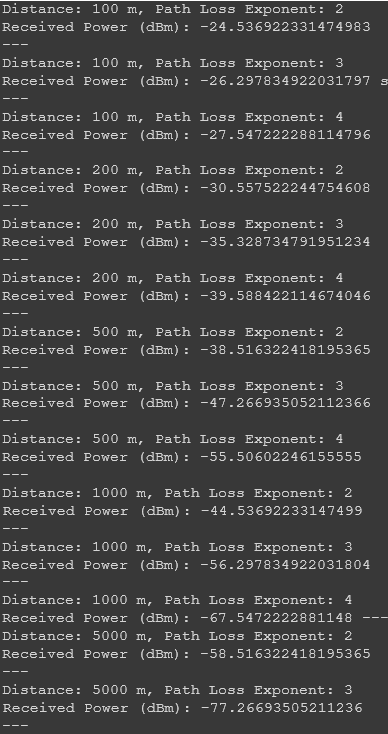
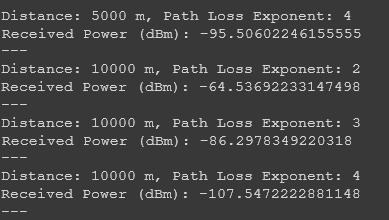
plt.xlabel("Distance (m)") plt.ylabel("Received Power (dBm)")

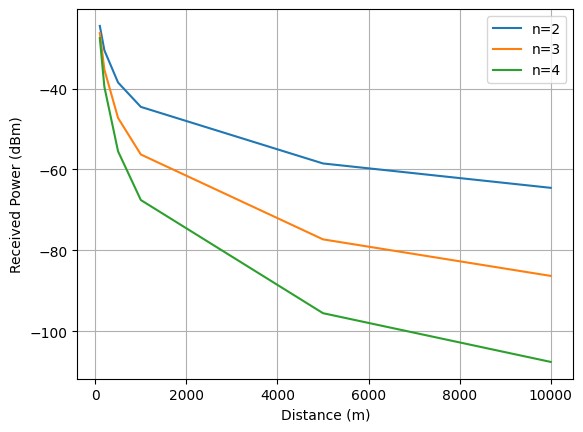
plt.legend()

plt.grid()

plt.show()

**Output:**

****



**Activity C:**

**Power Received at Different Distances with a Reference Distance**

# Step 1: Import necessary libraries (already imported in Activity A)

# Step 2: Define the received power from Activity A

received\_power = 7.0362e-006 # Replace with the actual received power from Activity A

# Step 3: Define the parameters

distances = [100, 200, 500, 1000, 5000, 10000] # Distances in meters path\_loss\_exponents = [1, 2, 3, 4]

# Step 4: Calculate received power at different distances with a reference distance point for distance in distances:

for path\_loss\_exponent in path\_loss\_exponents:

received\_power\_at\_distance = received\_power \* (reference\_distance / distance)\*\*path\_loss\_exponent

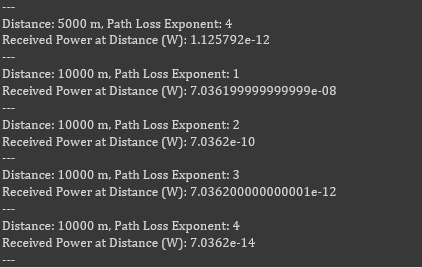
print(f"Distance: {distance} m, Path Loss Exponent: {path\_loss\_exponent}")

print(f"Received Power at Distance (W): {received\_power\_at\_distance}")

print("---")

**Output:**

****



**Conclusion:**

From this graphical visualization, we can say that for smaller distances, i.e., for distance < 100m the received power decreases significantly, after that, as distance increases the power received graph almost the same.

Using Colab, we can make various graphical representations and analysis where exceptions are occurring and solve them.