1. Characterize Optical Beam in Outer Space

Beam Divergence:

The divergence of a beam describes the angular spread of the beam as it propagates. For a Gaussian beam:

$$\theta = \frac{\lambda}{\pi \omega_0}$$

Where:

- θ: Beam divergence (radians).
- λ: Wavelength of the beam (m).
- ω_0: Beam waist (radius at the narrowest point, m).

The Beam divergence is inversely proportional to the beam waist. Larger beam waists (e.g., for collimated beams) result in smaller divergence angles.

Beam Spread:

Beam spread quantifies how the beam's width increases with propagation distance. For a Gaussian beam:

$$w(z) = w_0 \sqrt{1 + \left(\frac{z}{z_R}\right)^2}$$

Where:

- $w(z)$: Beam radius at distance zzz from the beam waist.
- $z_R = \frac{\pi \omega_0^2}{\lambda}$: Rayleigh range, the distance over which the beam remains collimated.

Within the Rayleigh range ($z < z_R$), the beam spread is minimal. Beyond z_R, the beam spread increases approximately linearly.

Intensity Profile:

The intensity profile describes the distribution of optical power across the beam. For a Gaussian beam:

$$I(r, z) = I_0 \exp\left(-\frac{2r^2}{w(z)^2}\right)$$

Where:

- $I(r, z)$: Intensity at radial position rrr and distance zzz.
- I_0: Peak intensity at the beam center.
- $w(z)$: Beam radius at distance z.

Here, the intensity is highest at the center ($r = 0$) and falls off exponentially with r. Moreover, the intensity profile is radially symmetric.

Quantitative Example:

Parameters:

- λ: 500nm
- Beam waist (w_0): 1mm
- Distance (z): Up to 10m

Derived Quantities:

Divergence:

$$\theta = \frac{5 \times 10^7}{\pi \times 10^{-3}} \approx 1.59 \times 10^{-4}\, rad$$

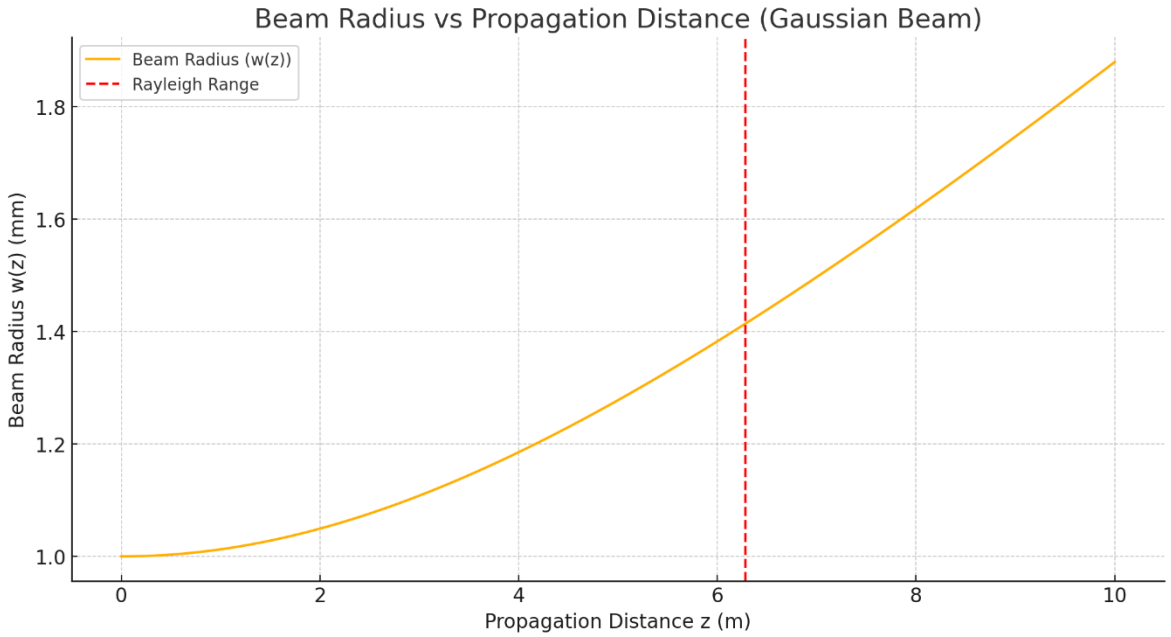
Beam Radius:

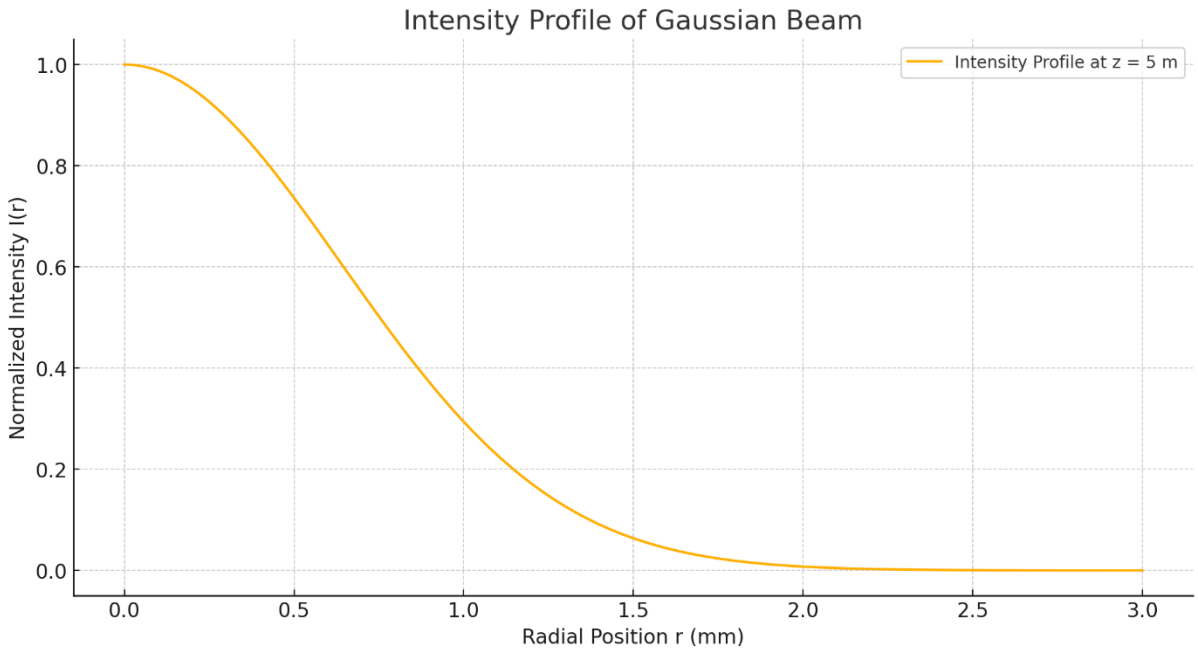
$w(z)$: Using $w_0 = 1mm, \lambda = 500nm$, compute for various z.

Intensity Profile:

Compute $I(r, z)$ for $r \in [0,3\omega(z)]$

Shown below is the visualization of the beam radius and the intensity profile for a Gaussian beam in Python.





It can be observed in the Beam Radius vs. Propagation Distance plot that:

- The beam radius $\omega(z)$ increases with propagation distance z.
- The beam growth is minimal within the Rayleigh range ($z < z_R$) and becomes linear for $z > z_R$.

It can be observed for the Intensity Profile plot at a Fixed Distance that:

- The intensity is highest at the center ($r = 0$) and decreases exponentially with the radial distance.
- The beam spreads wider at larger propagation distances, reducing peak intensity.

2. Characterize the Channel (Outer Space)

A wideband channel in Outer Space can simply be characterized using the Modified Saleh Valenzuela Channel Model. Herein, I refer to my own research paper:

[1] M. Nair, Q. Z. Ahmed and H. Zhu, "Hybrid Digital-to-Analog Beamforming for Millimeter-Wave Systems with High User Density," 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 2016, pp. 1-6, doi: 10.1109/GLOCOM.2016.7841879.

I provide a snippet in the page below (it can be easily adapted for Outer Space scenarios by setting the resolvable multipath $U = 0$, and the number of clusters $V = 0$. I am also happy to provide my original source code in Matlab):

chain is represented as $\boldsymbol{D}_i$ having dimensions $ML \times ML$. The complete digital beamformer $\boldsymbol{D}$ is given as:

$$\boldsymbol{D} = \text{diag}[\boldsymbol{D}_1, \boldsymbol{D}_2, \cdots, \boldsymbol{D}_N], \tag{1}$$

where $\boldsymbol{D}$ accounts for N-RF chains in the BS and is $NML \times NML$ dimensional.

A. 3D mm-Wave Channel Model

The 3D mm-Wave modified Saleh-Valenzuela (SV) channel impulse response (CIR) for the i-th RF chain and the m-th Tx antenna is given by [17], [18], [19]:

$$\begin{aligned} h^k_{i,m}(t) &= \sum_{v=0}^{V-1}\sum_{u=0}^{U-1} \alpha^k_{i,m,uv} h^k_{i,m,uv}\delta(t-\tau_v-\tau_{uv}) \\ &= \sum_{l=0}^{L-1} \alpha^k_{i,m,l} h^k_{i,m,l}\delta(t-l\tau), \end{aligned} \tag{2}$$

where $h^k_{i,m,l}$ is the k-th user CIR of l-th resolvable multi-path for the m-th Tx antenna of the i-th RF chain. V denotes the number of clusters, U the number of of resolvable multi-paths in one cluster, and $L = UV$ is the total number of resolvable multi-paths at the receiver. l is related to u and v by $l = vU+u$. In (2) $h^k_{i,m,uv} = |h^k_{i,m,uv}|e^{j\theta_{uv}}$ represents the fading gain of the u-th resolvable multi-path in the v-th cluster connecting the m-th antenna in the i-th RF chain to the k-th user. τ_v is the time-of-arrival (ToA) of the v-th cluster and $\tau_{uv} = u\tau$ denotes the ToA of the u-th resolvable multi-path in the v-th cluster. In our mm-Wave channel, it is assumed that the average power of a multi-path at a given delay is related to the power of the first resolvable multi-path of the first cluster through the following relationship [18], [19]:

$$P^k_{uv} = P^k_{00}\text{exp}\Big(-\frac{\tau_v}{\Psi}\Big)\text{exp}\Big(-\frac{\tau_{uv}}{\psi}\Big), \tag{3}$$

where $P^k_{uv} = P^k_l = |h^k_{i,m,uv}|^2$ represents the expected power of the u-th resolvable multi-path in the v-th cluster connecting the k-th user to the m-th antenna in the i-th RF BS chain. Ψ and ψ are the corresponding power delay constants of the cluster and the resolvable multi-path respectively. For the channel model to be generic, we assume that the delay spread, which is $(L-1)\tau$ of the mm-Wave channel spans $g \geq 1$ data bits, satisfying $(g-1)N_\tau \leq (L-1) \leq gN_\tau$, where N_τ is the number of time slots per symbol. Secondly, we assume that the L number of resolvable multipath components are randomly distributed, but they are the same over each symbol. Due to the wider bandwidth at mm-Wave, all the L multi-path components can be resolved at the Rx side [20], [21], and multi-path diversity will be exploited in the analog beamformer to significantly improve capacity in our proposed system.

The CIR experienced by the i-th RF chain and the k-th user is given as:

$$\boldsymbol{H}^k_i(t) = diag\left[\boldsymbol{H}^k_{i,0}(t), \boldsymbol{H}^k_{i,1}(t), \cdots, \boldsymbol{H}^k_{i,M-1}(t)\right], \tag{4}$$

where $\boldsymbol{H}^k_{i,m}(t)$ is the $L \times (2L-1)$ dimensional Block-Toeplitz temporal CIR convolution matrix associated with the i-th RF chain, m-th Tx antenna and the k-th user, given by (5) at the bottom of the page. Finally, $\boldsymbol{H}^k_i(t)$ will also be a temporal matrix be of dimension $ML \times M(2L-1)$.

The k-th user 3D BF gain $\alpha^k_{i,m,uv} = \alpha^k_{i,m,l}$ for every Tx antenna element of the i-th RF chain is given in (6). $F_{Rx,V}$ and $F_{Rx,H}$ are the Rx beam pattern for the vertical (V) and horizontal (H) polarizations, respectively. $F_{Tx,i,V}$ and $F_{Tx,i,H}$ are the Tx beam pattern for the i-th RF chain. $\phi^{VV}_l, \phi^{VH}_l, \phi^{HV}_l, \phi^{HH}_l$ are the initial random phases for vertical (VV), cross (VH, HV), and horizontal polarizations (HH) for the l resolvable multi-path. κ_m is the intra-cluster Rician K-factor associated with the m-th Tx antenna cluster [10]. ϑ_l and φ_l are the elevation and azimuth angle-of-arrival (AoA), respectively. Finally, $\theta_{l,m}$ and $\phi_{l,m}$ are the elevation and azimuth angle-of-departure (AoD) of the l-th resolvable multi-path and m-th Tx antenna in the i-th RF chain.

B. Received Symbols of Hybrid D-A BF

The L samples of received signal at the k-th user from the i-th RF chain is expressed as:

$$\boldsymbol{y}^k_i(t) = \boldsymbol{H}^k_i(t)\boldsymbol{A}_i(t)\boldsymbol{D}_i\boldsymbol{x}^k_i(t) + \boldsymbol{n}^k_i(t), \tag{7}$$

where

$$\boldsymbol{x}^k_i(t) = \left[\boldsymbol{x}^k_{i,0}(t), \boldsymbol{x}^k_{i,1}(t), \cdots, \boldsymbol{x}^k_{i,M-1}(t)\right]^T \tag{8}$$

are the $ML \times 1$ dimensional transmitted uncorrelated data symbols to the k-th user from the i-th RF chain. $\boldsymbol{x}^k_{i,m}(t) = [x^k_{i,m}(t), x^k_{i,m}(t-1), \cdots, x^k_{i,m}(t-L+1)]$ are the L symbol samples for all the l resolvable multi-paths, corresponding to the m-th Tx antenna in the i-th chain. BF design will be discussed in detail in the following section. $\boldsymbol{n}^k_i(t)$ is modelled as independent and identical distributed (iid) complex Gaussian random noise with zero mean and a variance of σ^2_i for the k-th user. The signal to noise ratio (SNR) of the i-th RF chain is denoted by γ_i and is given as [22]:

$$\gamma_i(\boldsymbol{D}_i, \boldsymbol{A}_i(t), \boldsymbol{H}^k_i(t)) = \gamma_0 \frac{|\boldsymbol{H}^k_i(t)\boldsymbol{A}_i(t)\boldsymbol{D}_i\boldsymbol{x}^k_i(t)|^2}{\sigma^2_i}, \tag{9}$$

where γ_0 is the average input SNR. Maximizing this SNR will lead to improved system capacity (in bit per second per Hz) for

$$\boldsymbol{H}^k_{i,m}(t) = \begin{bmatrix} h^k_{i,m,0}(t) & h^k_{i,m,1}(t) & \cdots & h^k_{i,m,L-1}(t) & 0 & \cdots & 0 \\ 0 & h^k_{i,m,0}(t-1) & h^k_{i,m,1}(t-1) & \cdots & h^k_{i,m,L-1}(t-1) & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & h^k_{i,m,0}(t-L+1) & h^k_{i,m,1}(t-L+1) & \cdots & h^k_{i,m,L-1}(t-L+1) \end{bmatrix} \tag{5}$$

$$\alpha^k_{i,m,l} = \begin{bmatrix} F_{Rx,V}(\varphi_l, \vartheta_l) \\ F_{Rx,H}(\varphi_l, \vartheta_l) \end{bmatrix}^T \begin{bmatrix} e^{j\phi^{VV}_l} & \sqrt{\kappa_m^{-1}}e^{j\phi^{VH}_l} \\ \sqrt{\kappa_m^{-1}}e^{j\phi^{HV}_l} & e^{j\phi^{HH}_l} \end{bmatrix} \begin{bmatrix} F_{Tx,i,V}(\phi_{l,m}, \theta_{l,m}) \\ F_{Tx,i,H}(\phi_{l,m}, \theta_{l,m}) \end{bmatrix} \tag{6}$$

3. Power Analysis

The Friis transmission equation is given by:

$$P_r = P_t \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi d}\right)^2$$

Where,

P_r is the received power in Watts

P_t is the transmit power in Watts

G_t is the transmit gain

G_r is the receive gain

λ is the wavelength of the EM radiation

d is the Tx-Rx distance

Given parameters:

Tx power range: 100mW:10mW:1W, i.e., 100mW to 1W in steps of 10mW

Distances: 200Km:200Km:1000Km, i.e, 200Km to 1000Km in steps of 200Km

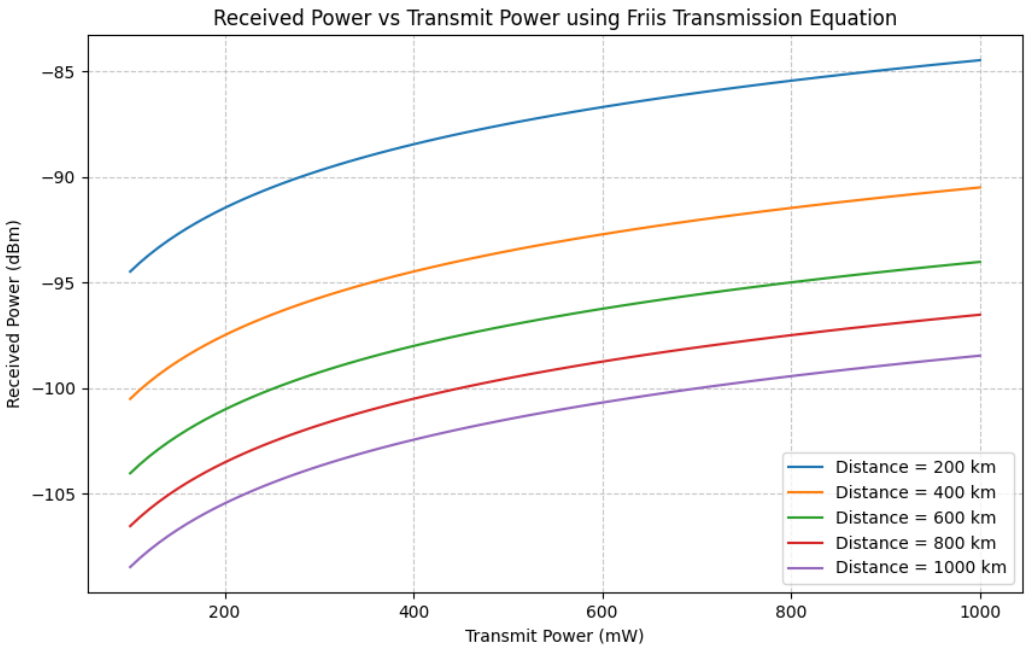
Assumptions:

$f = 2$GHz, i.e., L-band carrier frequency

$G_t = 30$dB, i.e., 1000 in linear scale with Tx beamforming array antenna

$G_r = 30$dB, i.e. 1000 in linear scale with Rx beamforming array antenna

Using the Friis Equation, we obtain plot the Rx power vs. Tx power for the given distances in Python as shown below:



It is observed that as the distance increases, the received power decreases significantly, following the inverse squared law. Moreover, the relationship between the Tx power and Rx power is linear on the Log scale for a fixed distance.

4. BER vs SNR for OOK Modulation

The BER for OOK is given by

$$BER = Q\left(\sqrt{2 \cdot \frac{E_b}{N_0}}\right)$$

Where,

$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$ is the Q-function

$\frac{E_b}{N_0}$ is the SNR, i.e., the ratio of symbol energy to the noise power spectral density

Given parameters:

Tx power: 200mW

Distances: 200Km:200Km:1000Km, i.e., 200Km to 1000Km in steps of 200Km

SNR: -2dB:1dB:7dB, i.e, -2dB to 7dB in steps of 1dB

Procedure:

1. Compute P_r using Friis transmission equation

2. Obtain E_b/N_0 in linear scale from the given SNRs in dB scale and compute the BER for each distance using the Q-function.

Assumptions:

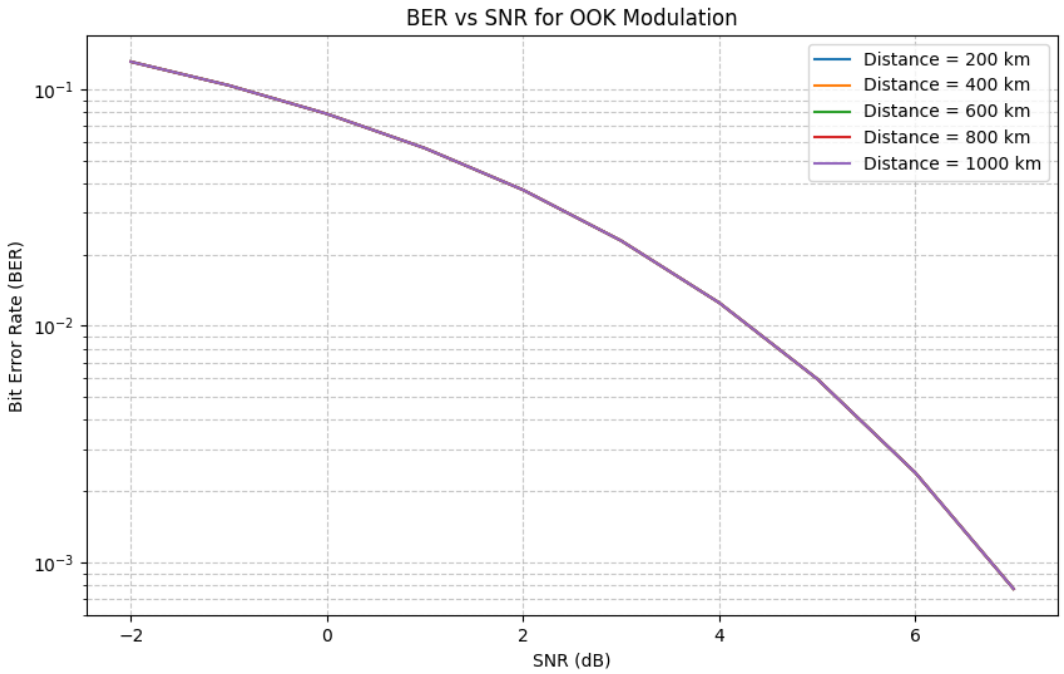
$f = 2$GHz, i.e., L-band carrier frequency

$G_t = 30$dB, i.e., 1000 in linear scale with Tx beamforming array antenna

$G_r = 30$dB, i.e. 1000 in linear scale with Rx beamforming array antenna

Utilizing the procedure given above, we can obtain the BER vs SNR plot for OOK modulation in Python for the given distances.

OO



Observation:

It seems that the BERs overlap irrespective of the distance. This is because of how the SNR in linear scale, Noise Power (N_0), and the symbol to noise ratio (E_b/N_0) are computed in the Python script.

For example, the SNR (linear) table is as follows:

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

Where as the noise power computed as $P_r./SNR$ is as follows:

7.8410e-16 6.2283e-16 4.9473e-16 3.9298e-16 3.1216e-16 2.4795e-16
1.9696e-16 1.5645e-16 1.2427e-16 9.8712e-17

1.9602e-16 1.5571e-16 1.2368e-16 9.8245e-17 7.8039e-17 6.1988e-17
4.9239e-17 3.9112e-17 3.1068e-17 2.4678e-17

8.7122e-17 6.9203e-17 5.4970e-17 4.3664e-17 3.4684e-17 2.7550e-17
2.1884e-17 1.7383e-17 1.3808e-17 1.0968e-17

4.9006e-17 3.8927e-17 3.0921e-17 2.4561e-17 1.9510e-17 1.5497e-17
1.2310e-17 9.7780e-18 7.7669e-18 6.1695e-18

3.1364e-17 2.4913e-17 1.9789e-17 1.5719e-17 1.2486e-17 9.9181e-18
7.8783e-18 6.2579e-18 4.9708e-18 3.9485e-18

And finally, the symbol to noise ratio (E_b/N_0) computed as $P_r/noisepower$ is as follows which is the same as the SNR linear:

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

0.6310 0.7943 1.0000 1.2589 1.5849 1.9953 2.5119 3.1623 3.9811 5.0119

Thus, the plot does seem to be counterintuitive, i.e., the BERs remain unchanged irrespective of the distance. However, if we observe the definition of the Q function for OOK modulation, it is given by

$$BER = Q\left(\sqrt{2 \cdot \frac{E_b}{N_0}}\right)$$

Thus, the BER only depends upon the receive symbol-to-noise ratio. This explains the paradox.

Appendix A: Python Script for Characterization of Optical Beam in Outer Space

```
import numpy as np

import matplotlib.pyplot as plt


# Parameters

lambda_ = 500e-9  # Wavelength (m)

w0 = 1e-3  # Beam waist (m)

z_values = np.linspace(0, 10, 500)  # Propagation distance (m)

r_values = np.linspace(0, 3 * w0, 500)  # Radial positions (m)

zR = np.pi * w0**2 / lambda_  # Rayleigh range


# Calculate beam radius w(z)

w_z = w0 * np.sqrt(1 + (z_values / zR)**2)


# Calculate intensity profile I(r, z) at a specific distance z (e.g., z = 5 m)

z_specific = 5  # Distance (m)

w_at_z = w0 * np.sqrt(1 + (z_specific / zR)**2)

I_0 = 1  # Assume normalized peak intensity

I_r = I_0 * np.exp(-2 * (r_values**2) / w_at_z**2)


# Plotting the beam radius over distance

plt.figure(figsize=(12, 6))

plt.plot(z_values, w_z * 1e3, label="Beam Radius (w(z))")

plt.axvline(zR, color="r", linestyle="--", label="Rayleigh Range")

plt.title("Beam Radius vs Propagation Distance (Gaussian Beam)")

plt.xlabel("Propagation Distance z (m)")

plt.ylabel("Beam Radius w(z) (mm)")

plt.grid(True, linestyle="--", alpha=0.7)

plt.legend()
```

```
plt.show()


# Plotting the intensity profile
plt.figure(figsize=(12, 6))
plt.plot(r_values * 1e3, I_r, label=f"Intensity Profile at z = {z_specific} m")
plt.title("Intensity Profile of Gaussian Beam")
plt.xlabel("Radial Position r (mm)")
plt.ylabel("Normalized Intensity I(r)")
plt.grid(True, linestyle="--", alpha=0.7)
plt.legend()
plt.show()
```

Appendix B: Python Script for Power Analysis

```
import numpy as np
import matplotlib.pyplot as plt

# Constants
c = 3e8  # Speed of light (m/s)
f = 2.0e9  # Frequency (Hz)
lambda_ = c / f  # Wavelength (m)
Gt = 1000  # Transmitter antenna gain
Gr = 1000  # Receiver antenna gain

# Transmit power range (100 mW to 1 W in 10 mW steps)
tx_power_mw = np.arange(100, 1001, 10)  # mW
tx_power_w = tx_power_mw / 1000  # Convert to W

# Distances (in km converted to m)
distances_km = [200, 400, 600, 800, 1000]
distances_m = np.array(distances_km) * 1000

# Calculate received power for each distance
received_power = []
for d in distances_m:
    Pr = tx_power_w * Gt * Gr * (lambda_ / (4 * np.pi * d))**2
    received_power.append(Pr)

# Plotting
plt.figure(figsize=(10, 6))
for i, d in enumerate(distances_km):
    plt.plot(tx_power_mw, 10 * np.log10(received_power[i]), label=f'Distance = {d} km')

# Adding labels and legend
plt.title("Received Power vs Transmit Power using Friis Transmission Equation")
plt.xlabel("Transmit Power (mW)")
plt.ylabel("Received Power (dBm)")
plt.grid(True, linestyle='--', alpha=0.7)
plt.legend()
plt.show()
```

Appendix C: Python Script for BER vs SNR for OOK Modulation

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import erfc

# Constants
c = 3e8  # Speed of light (m/s)
f = 2.0e9  # Frequency (Hz)
lambda_ = c / f  # Wavelength (m)
Gt = 1000  # Transmitter antenna gain
Gr = 1000  # Receiver antenna gain

# Parameters
tx_power_w = 200 / 1000  # Transmit power: 200 mW converted to Watts
distances_km = [200, 400, 600, 800, 1000]  # Distances in kilometers
distances_m = np.array(distances_km) * 1000  # Convert distances to meters
snr_db = np.arange(-2, 8, 1)  # SNR range in dB
snr_linear = 10 ** (snr_db / 10)  # Convert SNR from dB to linear scale

# BER computation
ber_results_separated_fixed = []  # Store BER arrays for each distance

# Loop through each distance to calculate and store BER
for d in distances_m:
    Pr = tx_power_w * Gt * Gr * (lambda_ / (4 * np.pi * d))**2  # Received power
    ber_distance = []  # Store BERs for this distance
    for snr in snr_linear:
        noise_power = Pr / snr  # Noise power for this SNR
        eb_n0 = Pr / noise_power  # Calculate Eb/N0
        ber = 0.5 * erfc(np.sqrt(eb_n0))  # Compute BER using Q-function
        ber_distance.append(ber)  # Append BER for this SNR
        ber_results_separated_fixed.append(ber_distance)  # Add BER array for this distance

# Plotting the BER vs SNR
plt.figure(figsize=(10, 6))
for i, d in enumerate(distances_km):
    plt.semilogy(snr_db, ber_results_separated_fixed[i], label=f'Distance = {d} km')  # label

# Adding labels, grid, and legend
plt.title("BER vs SNR for OOK Modulation")
plt.xlabel("SNR (dB)")
plt.ylabel("Bit Error Rate (BER)")
plt.grid(True, which='both', linestyle='--', alpha=0.7)
plt.legend()
plt.show()
```