

```
# =====
# Near-Field Channel Modeling for Sub-THz
# Full Version with 5 Plots
# Ready for Jupyter Notebook
# =====
```

این قسمت فقط توضیح است و می‌گویند برنامه مدل کانال نزدیک میدان برای فرکانس‌های بالا است و آماده اجرای نمودارها در دفترچه یادداشت است.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.constants import c
from scipy.optimize import curve_fit
```

این خطوط کتابخانه‌های لازم برای محاسبات عددی، رسم نمودار، سرعت نور و برازش داده‌ها را وارد می‌کنند.

```
f = 140e9          # 140 GHz
wavelength = c / f
Pt = 1
d = np.linspace(0.05, 2.0, 300) # distance in meters
```

فرکانس موج، طول موج، توان فرستنده و فاصله‌ها را برای محاسبات مشخص می‌کند.

```
def friis_model(d, wavelength):
    return (wavelength / (4*np.pi*d))**2
```

این تابع توان دریافت در میدان دور را بر اساس مدل استاندارد محاسبه می‌کند.

```
Pr_far = Pt * friis_model(d, wavelength)
```

توان دریافت در میدان دور را با ضرب در توان فرستنده بدست می‌آورد.

```
def near_field_model(d, wavelength):
    return (wavelength / (4*np.pi*d))**2 * (1 + 0.4*np.cos(2*np.pi*d/wavelength))
```

این تابع توان دریافت در میدان نزدیک را با اثر نوسان‌های امواج محاسبه می‌کند.

```
Pr_near = Pt * near_field_model(d, wavelength)
```

توان دریافت میدان نزدیک را با ضرب در توان فرستنده محاسبه می‌کند.

```
np.random.seed(0)
noise = 0.05 * np.random.randn(len(d))
Pr_near_noisy = Pr_near * (1 + noise)
```

نویز تصادفی به داده‌های میدان نزدیک اضافه می‌شود تا شبیه داده واقعی باشد.

```
def fitted_model(d, a, b):
    return (a/d**2) * (1 + b*np.cos(2*np.pi*d/wavelength))
```

یک مدل تجربی برای برازش داده‌های میدان نزدیک تعریف می‌کند که شامل توان معکوس فاصله و نوسان است.

```
params, _ = curve_fit(fitted_model, d, Pr_near_noisy)
Pr_fit = fitted_model(d, params[0], params[1])
```

مدل تجربی را روی داده‌های نویزی برازش می‌کند و توان تخمینی برازش شده را محاسبه می‌کند.

```
plt.figure(figsize=(8,5))
plt.plot(d, 10*np.log10(Pr_far), label="Far-Field (Friis)")
plt.plot(d, 10*np.log10(Pr_near), label="Near-Field")
plt.xlabel("Distance (m)")
plt.ylabel("Received Power (dB)")
plt.title("Far-Field vs Near-Field")
plt.legend()
plt.grid(True)
plt.show()
```

نمودار مقایسه توان دریافت میدان دور و نزدیک رسم می‌کند و برچسب‌ها و شبکه نمودار را اضافه می‌کند.

```
plt.figure(figsize=(8,5))
plt.plot(d, 10*np.log10(Pr_far), label="Far-Field (Friis)")
plt.plot(d, 10*np.log10(Pr_near), label="Near-Field")
plt.xlabel("Distance (m)")
plt.ylabel("Received Power (dB)")
plt.title("Far-Field vs Near-Field")
plt.legend()
plt.grid(True)
plt.show()
```

نمودار داده‌های نویزی و مدل برازش شده میدان نزدیک را رسم می‌کند تا مطابقت آن‌ها دیده شود.

```
def gaussian_beam(d):
    return Pr_near * np.exp(-d/0.5) # simplified Gaussian decay

def bessel_beam(d):
    return Pr_near * (1 + 0.2*np.cos(2*np.pi*d/wavelength)) # simplified Bessel effect
```

دو مدل پرتوی متفاوت تعریف می‌کند: یکی با کاهش نمایی و دیگری با نوسان موجی.

```
Pr_gaussian = gaussian_beam(d)
Pr_bessel = bessel_beam(d)
```

توان دریافت برای هر یک از پرتوی گوسی و بسل محاسبه می‌شود.

```
plt.figure(figsize=(8,5))
plt.plot(d, 10*np.log10(Pr_gaussian), label="Gaussian Beam")
plt.plot(d, 10*np.log10(Pr_bessel), label="Bessel Beam")
plt.xlabel("Distance (m)")
plt.ylabel("Received Power (dB)")
plt.title("Gaussian vs Bessel Beam in Near-Field")
plt.legend()
plt.grid(True)
plt.show()
```

نمودار مقایسه‌ای توان دریافت برای دو نوع پرتو در میدان نزدیک رسم می‌کند.

```
plt.figure(figsize=(8,5))
plt.plot(d, 10*np.log10(Pr_near), label="Near-Field Clean")
plt.plot(d, 10*np.log10(Pr_near_noisy), label="Near-Field Noisy")
plt.xlabel("Distance (m)")
plt.ylabel("Received Power (dB)")
plt.title("Effect of Noise on Near-Field Channel")
plt.legend()
plt.grid(True)
plt.show()
```

نمودار نشان می‌دهد اضافه کردن نویز چقدر توان دریافت میدان نزدیک را تغییر می‌دهد.

```
angles = np.linspace(-30,30,50) # beam angle in degrees
D, A = np.meshgrid(d, angles)
Pr_heatmap = (wavelength/(4*np.pi*D))**2 * (1 + 0.3*np.cos(2*np.pi*D/wavelength)) * np.cos(np.radians(A))
```

برای زوایای مختلف پرتو، توان دریافت را محاسبه و یک شبکه دو بعدی برای نمودار حرارتی آماده می‌کند.

```
plt.figure(figsize=(8,5))
plt.pcolormesh(D, A, 10*np.log10(Pr_heatmap), shading='auto', cmap='viridis')
plt.colorbar(label="Received Power (dB)")
plt.xlabel("Distance (m)")
plt.ylabel("Beam Angle (degrees)")
plt.title("Heatmap of Near-Field Received Power vs Distance and Angle")
plt.show()
```

نمودار حرارتی توان دریافت میدان نزدیک را نسبت به فاصله و زاویه رسم می‌کند و نقشه رنگی اضافه می‌کند.

```
print("Fitted Parameters for Empirical Near-Field Model:")
print(f"a = {params[0]:.4e}")
print(f"b = {params[1]:.4f}")
```

پارامترهای برازش شده مدل تجربی میدان نزدیک را چاپ می‌کند.