Práctica B3 RSI

Desarrollo de un modelo empírico. Cálculo por regresión de un modelo de pérdidas de trayecto

Samuel John Suffern Sánchez

9 de mayo de 2021

Índice

Índice de figuras

1. Introducción

La señal recibida a través de un canal de propagación radio-móvil tiene superpuestas variaciones de tipo rápido (**instantáneas**, P) y lento (**media local**, P_r). Además existen unas variaciones mucho más lentas que dependen de la distancia entre transmisor y receptor (**perdidas de trayecto y su potencia nominal**, P_R)

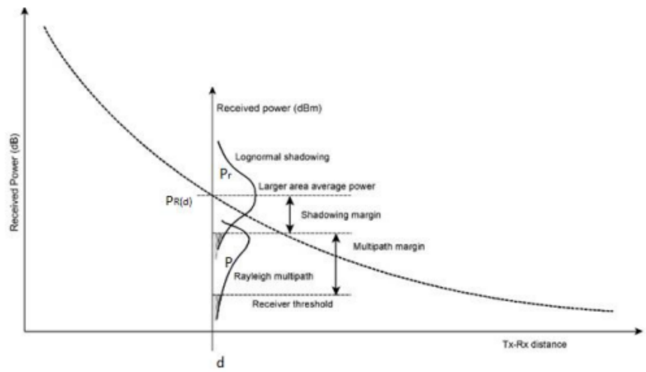

Figura 1: Slow and fast variations and associated margins

Normalmente la señal nominal decaerá de forma inversamente proporcional a la distancia elevada a un exponente α que estará en torno a 4, esto es, expresando las potencias en unidades lineales,

$$p_R(W) = p_R(d=1)/d^{\alpha} \qquad \text{(Potencia en unidades lineales)}$$

Es decir, normalmente, la señal caerá con la distancia de forma más acusada que para el caso de espacio libre donde $\alpha = 2$.
Al valor nominal para una distancia d, p_R, se llega removiendo, mediante filtrado, las variaciones rápidas en la potencia instantánea, p y las variaciones lentas, p_r, debidas a efecto sombra, mediante otro filtrado (media local).
Si expresamos anterior (Potencia en unidades lineales), en unidades logarítmicas:

$$p_R(dBW) = p_R(d=1) - 10\alpha log(d) \qquad \text{(Potencia en unidades logarítmicas)}$$

Si ahora calculamos **pérdidas de trayecto (path loss)**, o pérdidas nominales a una distancia d, vemos que tendrá una expresión similar.

$$L(dB) = L_1(d=1) + 10\alpha log(d) \qquad \text{(Path loss)}$$

Por último, para calcular la potencia nominal a una distancia d, utilizamos:

$$P_R(dBW) = EIRP(dBW) - L(dB) + G_R \qquad \text{(Potencia nominal a una distancia d)}$$

1.1. Objetivo

Tenemos como objetivo en esta práctica, a partir de datos experimentales, extraer los parámetros de un modelo de este tipo. Modelo que predice la potencia nominal a una determinada distancia d.

2. Inicio del entorno en Matlab

Lo primero que haremos será leer el archivo `data13.m` que proporcionan unos vectores que se corresponden con un conjunto de parejas de valores de **distancia d(km)** y **potencia media** $P_r(dBm)$

```
load data13
```

3. Análisis de los datos experimentales

Podemos observar la salida de las medidas tomadas, pero sin filtrar, no podremos distinguir nada.

```
figure, plot(distkm, Pr');
xlabel('Distance in km');ylabel('Measurement Pr')
```

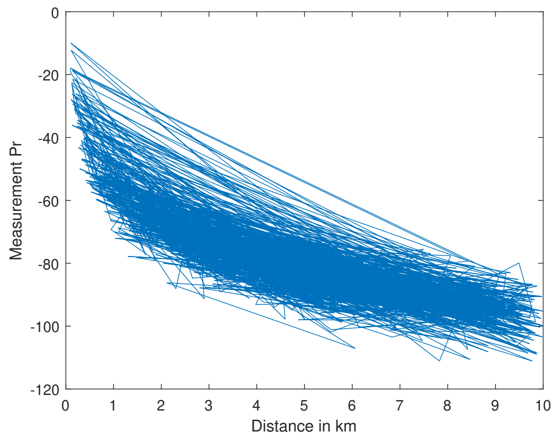

Figura 2: Plot Data13 values (P_r / distance in km)

3.1. Ordenación de las muestras

Como primer paso vamos a ordenar las muestras en función de la distancia en sentido creciente.

```
[distkm,I] = sort(distkm);
Pr = Pr(I);
figure, plot(distkm, Pr, '+r');
xlabel('Distance from Tx, km'); ylabel('Low-pas filtered Received power, dBm')
```

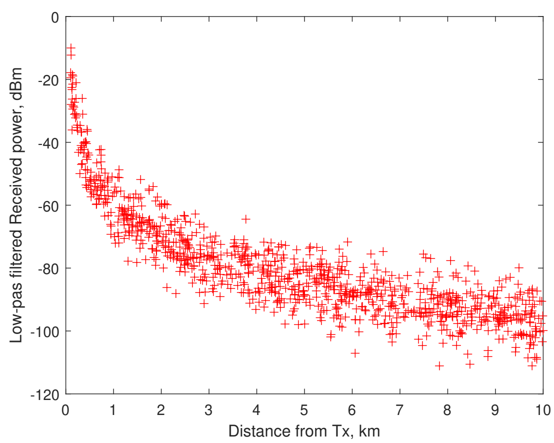

Figura 3: Gráfico de dispersión **scatter plot** de los datos en el fichero a analizar

En el modelo nos interesa obtener la distancia en unidades logarítmicas, así que aplicamos: $10log(d_{km})$. También aprovechamos para represental el cálculo de la **regresión lineal** con `polyfit` y `polyval`.

```
dlog = 10*log10(distkm); % Representaci n de la distancia en kil metros
figure, hold on;
plot(dlog, Pr, '+g') % Representaci n de la potencia recibida vs distancia en u. logar tmicas
N = 1;
P = polyfit(dlog,Pr,N);
Prfitted = polyval(P,dlog);
plot(dlog, Prfitted, 'r', 'LineWidth', 3)
xlabel('10 log_1_0 (d_k_m)')
ylabel('Received mean local power, dBm')
```

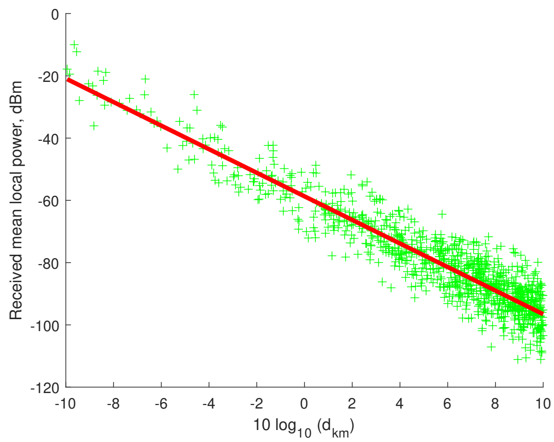

Figura 4: Eje de abscisas en unidades logarítmicas y ajuste de modelo lineal

El modelo/polinommio de **primer orden** resultante tiene los siguientes parámetros:

```
disp(['Stright line coefficients: a_1 : ',num2str(P(1)),' and a_0 : ',num2str(P(2))])
disp(' ')
```

```
>> Stright line coefficients: a_1 : -3.7852 and a_0 : -58.709
```

Es decir:

P = -3.7852 -58.709

Es decir, el modelo que queremos es:

$$P_R = A - Blog(d_{km}) = A - 10\alpha log(d_{km}) = -58.709 - 37.852log(d_{km}) \tag{Modelo}$$

Donde A es la potencia a distancia unidad. Si las distancias están en km entonces A es la potencia a 1 km. Si las distancias están en m entonces P_R es la potencia a 1 m. Si P_R está dada en dBm, entonces A también estará en dBm, si P_R está dada en dBW, A estará en dBW.

Sobre los valores de la señal nominal se superponen variaciones lentas de carácter gaussiano con media proporcionada por la ecuación anterior y desviación típica, $\sigma_L(dBm)$. La medida proporcionada, $P_r(dBm)$, en el archivo leído, representa la potencia media local, suma de la potencia nominal a una determinada distancia más las variaciones lentas, es decir,

$$P_r = P_R + X(0, \sigma_l) \tag{1}$$

donde $X(0, \sigma_l)$ se suele modelar como una **variable gaussiana** de media nula y desviación típica σ_L

Normalmente, el primer término del modelo resultante corresponde a la potencia recibida a la distancia de referencia, es decir, 1km en este caso.

3.2. Residuos

Los residuos se corresponden con la diferencia entre los valores experimentales (`Pr`) con respecto a los valores obtenidos teóricamente (`Prfitted`).

Si representamos los residuos:

```
X = Pr - Prfitted;
figure, plot(dlog, X, 'v')
xlabel('10log(d(km))')
ylabel('Residuals, dB')

figure, plot(distkm, X, 'v')
xlabel('d (km)')
ylabel('Residuals, dB')
```

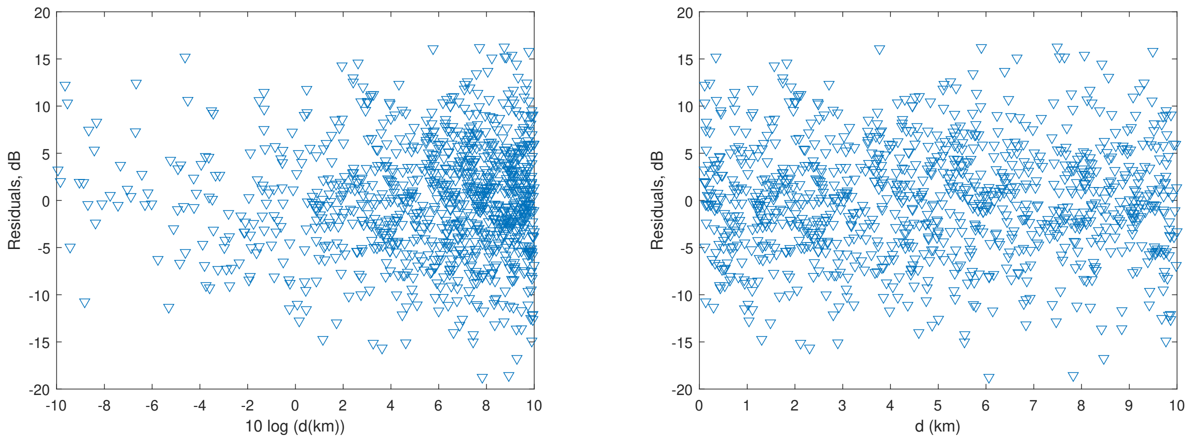

Figura 5: Residuos: Valores medidos - Modelo teórico

Podemos entonces calcular la media y desviación típica de los residuos.

```
meanX = mean(X);
sigmaL = std(X);
disp(['Mean and stadard deviation of random variable X:', num2str(meanX),'', num2str(sigmaL)])
```

```
>> Mean and stadard deviation of random variable X :-1.9568e-14  5.9937
```

Vamos a verificar que los residuos siguen un modelo gaussiano:

3.2.1. Modelo Gaussiano

Para ellos vamos a utilizar la siguiente función en Matlab rescatando el código utilizado de la práctica B2.

```
function compruebaNormal(signal)
    Pfilt = signal;
    % C lculo media y desviaci n t pica
    MM = mean(Pfilt);
    SS = std(Pfilt);
    % Display the los valores obtenidos
    disp(['Series mean : ', num2str(MM),' dBm'])
    disp(['Series std : ', num2str(SS),' dBm'])
    % --------------------- %
    N_bins = 80;
    [a,b] = hist(Pfilt,N_bins);
    histBin = b(2)-b(1);
    a = a/length(Pfilt);
```

```
        aa = cumsum(a);

        % Theoretical distribution
        Paxis = [min(Pfilt):max(Pfilt)];
        pdf = 1/(sqrt(2*pi)*SS)*exp(-0.5*((Paxis-MM)/SS).^2);
        fhist = pdf*histBin;
        figure, bar(b,a,'y'), hold on, plot(Paxis, fhist,'r', 'LineWidth',2)
        xlabel('Potecias medias, dBm')
        ylabel('Probabilidades')
        F = 1-qfunc((Paxis-MM)/SS);
        figure, semilogy(b,aa, 'y'), hold on, semilogy(Paxis, F, 'r','LineWidth',2)
        xlabel('Potecias medias, dBm')
        ylabel('Probabilidad de no exceder la abscisa')
    end
```

Basta con llamar a la función pasando por argumentos la variable que contiene los residuos:

```
        compruebaNormal(X)
```

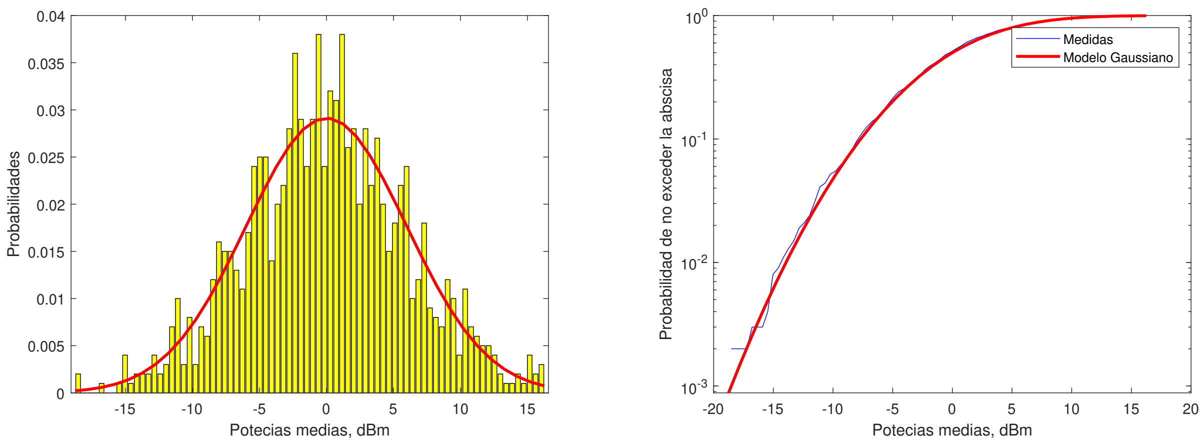

Figura 6: Funciones distribución y densidad de probabilidad teóricas vs medidas de los residuos

```
Series mean : -1.9568e-14 dBm
Series std : 5.9937 dBm
```

Como observamos en las gráficas obtenidas, los residuos siguen una distribución Normal.
Si observamos la función distribución de probabilidad con `normplot`:

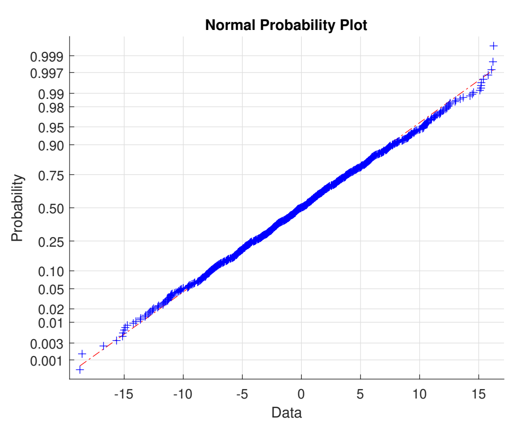

Figura 7: Función distribución con normplot mediante la Statistical Toolbox

4. Generación de datos aleatorios

Con las siguientes líneas vamos a intentar generar unos datos aleatorios y simular lo ya realizado.

```
EIRP=40; %dbm
EIRP_dbW= 10;
f= 2e9;
sigmal = 6;
nn = 3.8;
GRx= 0;
MaxDistKm= 10000;
MinDistKm = 100;
d= [MinDistKm:9.9:MaxDistKm];
d_km = d/1000;
L1= 20*log10(4*pi*1000*f/3e8); %perdidas por espacio libre
L= L1 + 10*nn*log10(d_km);
Pr_dbm = EIRP - L + GRx
```

El vector d es un vector 1x1001, se piden 1000 muestras, pero no hemos conseguido hacerlo de otra manera.

Falta añadir la aleatoriedad a potencia recibida. Sabemos que sigue una distribución normal de media 0 y desviación típica 6. Aplicando entonces la ecuación 1:

```
va = randn(1,1001)*sigmal+0;
Pr = Pr_dbm + va;
```

En este caso no es necesario ordenar el vector distancias con la potencia, ya que se ha creado de manera ordenada. Si hacemos el plot del vector potencia con respecto a la distancia, quedaría de la siguiente manera:

```
plot(dkm,Pr,'r');xlabel('Distancia from Tx, km');ylabel('Measurement, Pr')
```

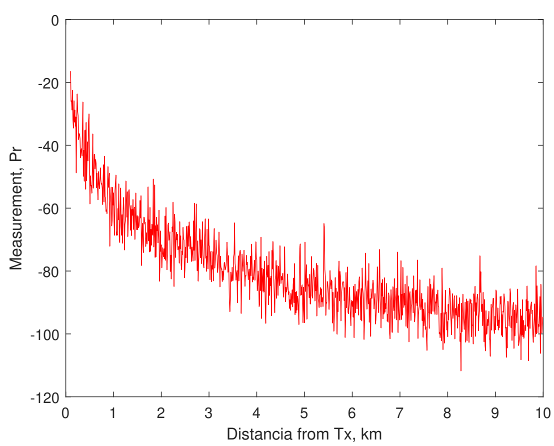

Figura 8: Valores de Potencia recibida con respecto a la distancia

Nos interesa la representación logarítmica de la distancia:

```
dlog = 10*log10(dkm); % Representaci n de la distancia en kil metros
figure, hold on;
plot(dlog, Pr_dbm, '+g') % Representaci n de la potencia recibida vs distancia en u.
    logar tmicas
N = 1;
P = polyfit(dlog,Pr_dbm,N);
Prfitted = polyval(P,dlog);
plot(dlog, Prfitted, 'r', 'LineWidth', 3)
xlabel('10 log_1_0 (d_k_m)')
ylabel('Received mean local power, dBm')
```

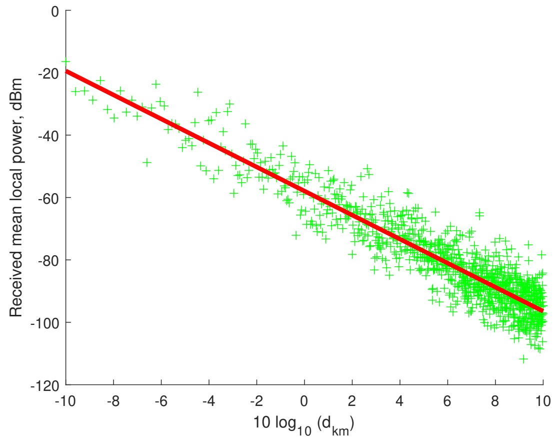

Figura 9: Eje de abscisas en unidades logar ıtmicas y ajuste de modelo lineal

A diferencia del apartado anterior, la regresión y los valores medidos se aproximan mucho. Observando los parámetros del polinimio de primer orden:

```
disp(['Stright line coefficients: a1 : ',num2str(P(1)),' and a0 : ',num2str(P(2))])
```

```
>> Stright line coefficients: a1 : -3.8511 and a0 : -57.9402
```

P = -3.8511 -57.9402

4.1. Residuos

Representando los residuos, al igual que en el apartado anterior:

```
X = Pr' - Prfitted;
figure, plot(dlog, X, 'v')
xlabel('10log(d(km))')
ylabel('Residuals, dB')

figure, plot(dkm, X, 'v')
xlabel('d (km)')
ylabel('Residuals, dB')
```

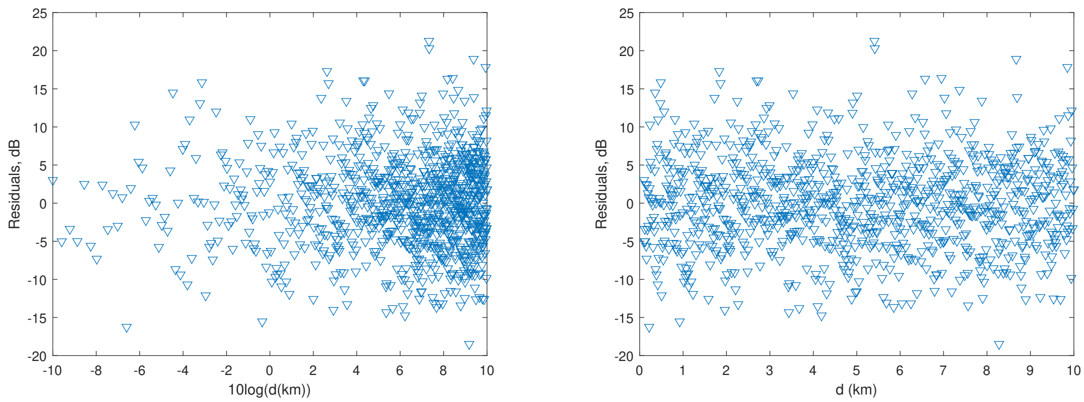

Figura 10: Residuos: Valores medidos - Modelo teórico

4.1.1. Modelo Gaussiano

Faltaría comprobar si estos residuos siguen una distribución normal. Para ello, al igual que en el apartado anterior, utilizaremos la función creada `compruebaNormal()`

```
compruebaNormal(X);
```

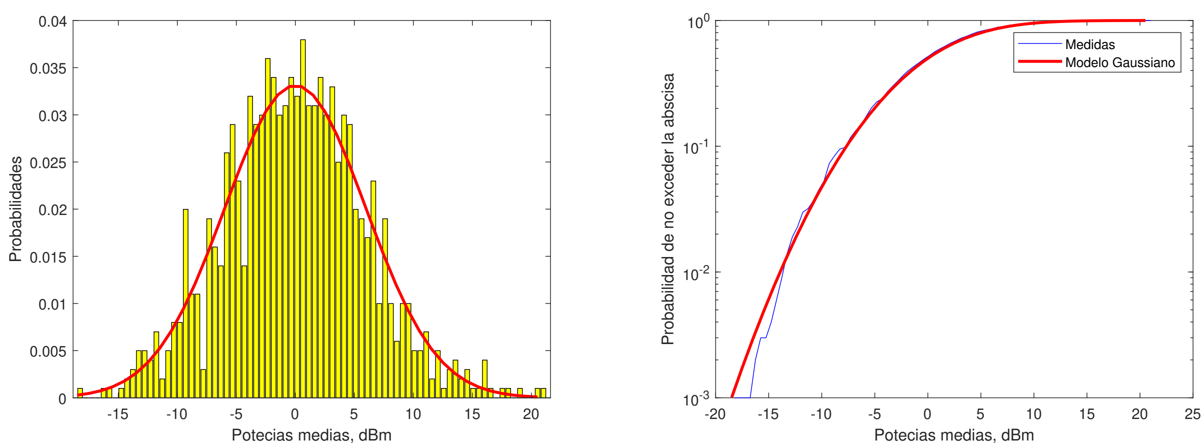

Figura 11: Funciones distribución y densidad de probabilidad teóricas vs medidas de los residuos

```
Series mean : -1.4403e-14 dBm
Series std : 5.9855 dBm
```

Observamos que la media es cero y la desviación típica coincide con 6.

Por último podemos observar la función distribución de probabilidad con `normplot`:

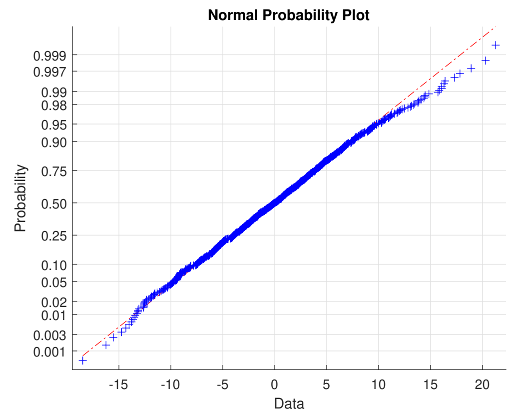

Figura 12: Función distribución con normplot mediante la Statistical Toolbox

5. Conclusión

Para una potencia recibida con variación con la distancia, es posible separar las muestras residuales de la propia muestra de interés.

Sabiendo que las muestras residuales siguen una distribución gaussiana, somos capaces de separar dichas muestras simplemente restando a la potencia recibida los valores obtenidos teóricamente mediante un modelo de regresión lineal.

El resultado de la resta será cuanto es de diferente el valor recibido con el valor real del modelo que debería tener. Esto es precisamente la definición de (**residuos**).

Con lo aprendido en la práctica B2, seremos capaces de comprobar si la distribución de estos residuos, de las variaciones lentas, es gaussiana, utilizando en este caso, un filtro rectangular.