5.2. SIMULACION DE UN CANAL RADIO.

El sistema para el cual vamos a realizar la simulación es el sistema IS-95. Éste se encuentra actualmente en servicio en Norteamérica, siendo un sistema de segunda generación basado en CDMA. Este sistema de segunda generación, junto con GSM en Europa, PDC en Japón o US-TDMA en América, está pensado para que sean desarrollados en una red de celdas macros, a pesar de existir estructuras que permiten ofrecer soluciones de capacidad en espacios más reducidos. Sin embargo, y hay radica la mayor diferencia con los sistemas de tercera generación, estos no están pensados para ofrecer elevadas tasas de bits, aspecto que incorpora UMTS y que esta definiendo el desarrollo e implementación de las comunicaciones en la actualidad.
En este capitulo se va a implementar una cadena de bloques que simulen el comportamiento de un canal de comunicaciones IS-95 con tecnología CDMA desde el transmisor al receptor, con la cual logremos caracterizar el canal de radio digital para obtener el equivalente modelado de Markov a partir de las series de error que obtendremos de las simulaciones del canal completo IS-95.

5.2.1. Modelo del interfaz de radio CDMA en IS-95.

En el sistema CDMA de IS-95 el enlace ascendente y descendente difieren ya que son implementados con distinta modulación. Por otro lado el enlace ascendente utiliza detección no coherente mientras que el descendente utiliza detección coherente.

Se ha implementado en este estudio el enlace descendente modelado como un sistema de espectro ensanchado QPSK con receptor (móvil) coherente. El enlace ascendente presenta en un sistema CDMA el problema derivado del control de potencia sintetizado en el fenó1meno near-far y que provoca que el nivel de ruido que se encuentra el móvil no sea constante si no funciona de forma correcta el algoritmo de control de potencia o en su defecto existe un fuerte demanda de tráfico en el sistema. Este problema no existe el enlace descendente ya que el nodo de red transmite en el enlace descendente siempre con la misma potencia y por lo tanto el nivel de interferencia debido a otras comunicaciones depende del número de usuarios que tenga el sistema el cual puede ser controlado con mecanismos de control de congestión.

Para estudiar el canal se ha utilizado una entrada binaria pseudoaleatoria que hace las veces de voz codificada. Esta entrada binaria será comparada con la salida en el receptor determinando el patrón de error que introduce el sistema y que por lo tanto la secuencia de errores que lo caracteriza.

La señal binaria caracterizada en IS-95 por una tasa binaria de 19.2 kbps es ensanchada sobre una señal compleja de 1.2288 Mbps. Este ensanchamiento se produce con la matriz de Walsh de dimensión 64, mediante la cual se consiguen 64 códigos de 64 bits de longitud cada uno. Estos codigos se implementan en la función walsh.m.

```
function [W]=walsh(dim)

% Funcion que genera la matriz de Walsh con dimension
% dim que sera un numero potencia de dos.
%
%    function [W]=walsh(dim)
%
% W: Matriz de Walsh resultante.
```

```
% dim: dimension de la matriz de Walsh

m=log2(dim);
W=0;
for(i=1:m)
   W=[W W;W ~W];
end
      return;
```

Con estos códigos se consigue ensanchar de forma individualizada la señal binaria hasta obtener un señal binaria de tasa 1.2288 Mbps. A este ensanchamiento le llamamos ensanchamiento ortogonal. Por ultimo y antes de modular la señal en QPSK se multiplica la señal por dos secuencias pseudoaleatorias a 1.2288 Mbps generadas con registros de desplazamientos de 6 etapas y que tienes la forma IPN=[1 6] y QPN=[1,2,5,6]. IPN multiplica a la señal binaria en la rama de fase mientras que QPN la multiplica en la rama de cuadratura como se explica en el capitulo 3.3. A este proceso se le llama ensanchamiento en cuadratura. La implementación de los códigos de ensanchamiento la realiza la función pnsec.m.

```
function [pn]=pnsec(n,vec,inicio)

% Funcion que genera una secuencia pseudoaleatoria periodica a partir de
% una estructura en forma de registro de desplazamiento definida por la
% longitud del registro.
%
%             [pn]=pnsec(n,vec,inicio)
%
% pn: Secuencia pseudoaleatoria de longitud 2^n.
% n: Longitud del registro de desplazamiento.
% vec: Configuracion del registro de desplazamiento. Es un vector de
%     longitud menor que n.
% inicio: Estado inicial del registro de desplazamiento. Si no se
%       especifica nada sera [1 0 0 0 .. 0 0 1]

reg=zeros(1,n);
if (nargin==3)
   reg=inicio;
else
   reg(n)=1;
   reg(1)=1;
end
tam=2^n;
out=zeros(1,tam);
for(i=1:tam)
   out(i)=reg(n);
   for(k=n:-1:2)
      reg(k)=reg(k-1);
   end
   aux=0;
   for(j=1:length(vec))
      aux=xor(aux,reg(vec(j)));
   end
   reg(1)=aux;
end
pn=out;
```

```
      return;
```

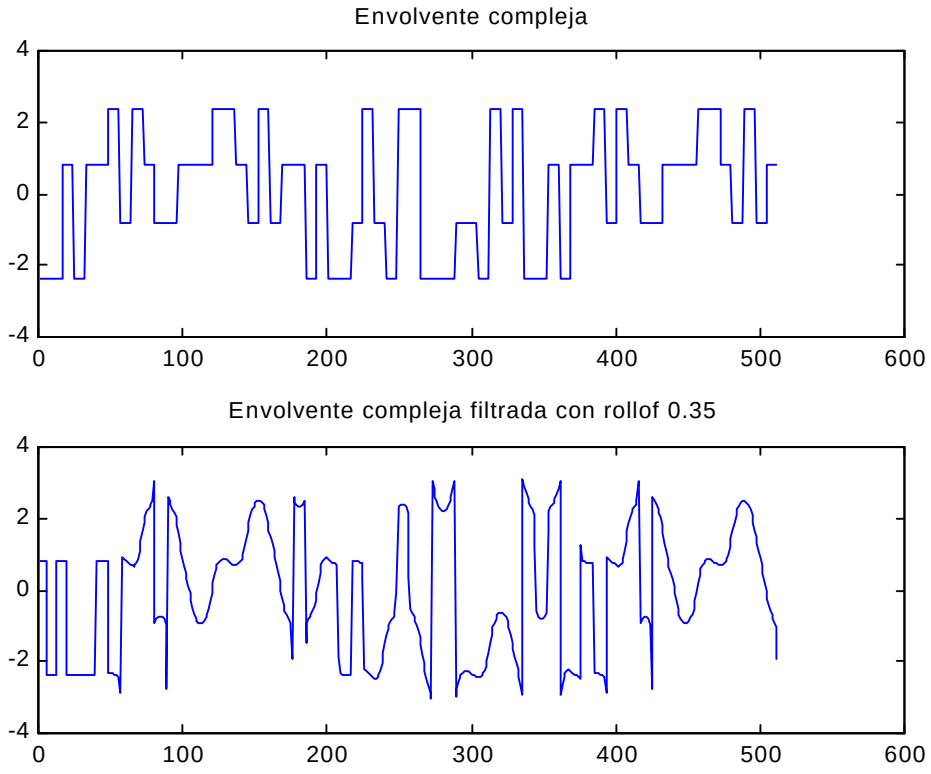
Una vez obtenida la secuencia binaria ensanchada y en cuadratura realizamos hemos de tratar la señal para realizar una simulación del sistema en banda base. Para ello sobre cada una de las ramas sobre la que ya tengamos la señal binaria ensanchada filtraremos con un filtro coseno alzado con 35% de roll-off. La señal de esta forma tratada en un sistema real sería elevada en frecuencia multiplicando la rama en fase por un coseno y la rama en cuadratura por el seno a la banda de 1.8 GHz que es donde esta localizado el sistema. Sin embargo, y ya que la simulación la vamos a realizar en banda base nuestra señal resultante será la suma compleja de la componente filtrada en fase y cuadratura. Todo este proceso es realizado por la función qpsk21sg.m.

```
function [y,bits]=qpsk21sg(R,L,M,sem)

% QPSK1 Funcion que reproduce un transmisor QPSK, para un sistema CDMA
% IS-95 con codigo Walsh y codigo corto(generado con registro de 6 bits).
%
%       y=qpsk21sg(R,L,M,sem)
%
% y: salida del transmisor.
% bits: Secuencia de bits generados.
% R: tasa de bits.
% L: longitud de la señal a generar.
% M: numero de muestras tomadas en cada bit.
% sem: semilla deseada para la simulacion.

% Generacion de los datos binarios.

rand('seed',sem);       % Se establece el valor de la semilla a sem.
bits=2*round(rand(1,L))-1;
n=6;
p1=[1,6];
p2=[1,2,5,6,];
pn1=pnsec(n,p1);
pn2=pnsec(n,p2);
W=walsh(64);
wpn=W(2,:);
ind=find(pn1==0);       % Convertimos las secuencias aleatorias en
                        % bipolares.
pn1(ind)=-1;
ind=find(pn2==0);
pn2(ind)=-1;
ind=find(wpn==0);
wpn(ind)=-1;
biw=zeros(0,0);
for(k=1:L)
   biw=[biw ((pn1+j*pn2).*wpn)*bits(k)];
end
ypre=zeros(0,0);
ypre=zeros(1,length(biw)*M);
for(k=1:length(biw))
   ypre((k-1)*M+1:k*M)=biw(k);
end
```

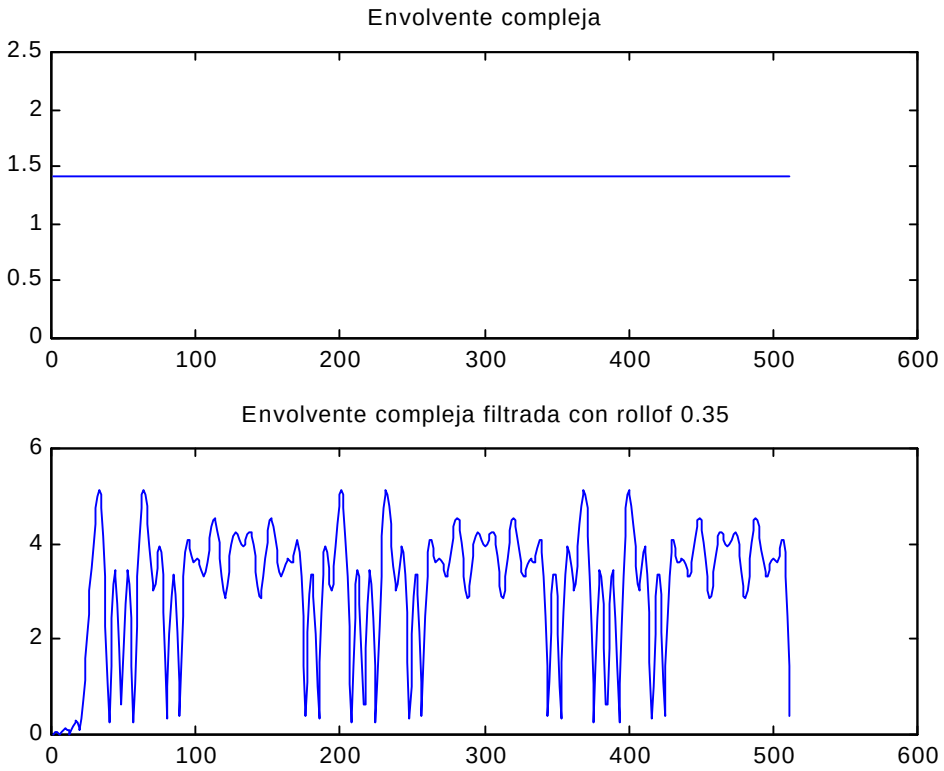
```
% Forma del filtro coseno alzado de longitud. El pulso coseno alzado va a
% durar dos
% periodos de bit.

% El factor de roll-off es del 35%.

clear biw;clear bic;clear pn2;clear pn1;
[num,den]=rcosine(R,M*R,'sqrt',0.35);
y=filter(num,den,ypre);

return
```

Estas señales, ya analogicas, serán simuladas utilizando 8 muestras por chip por lo que la tasa de muestreo binario llega a 9.8304 Msps.

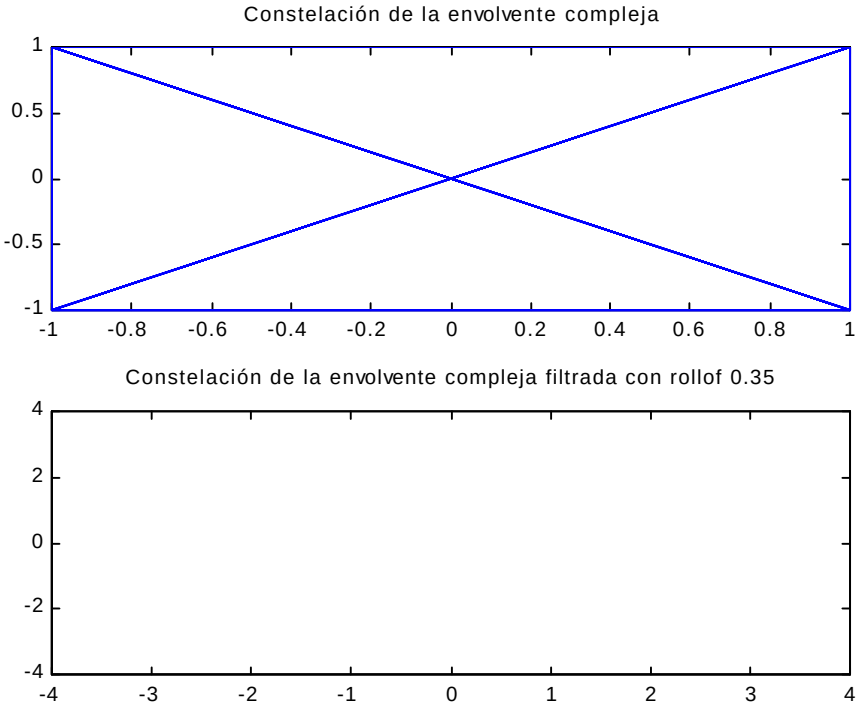
En las siguientes gráficas vemos la fase de la secuencia binaria una vez ensanchada en cuadratura. En las abcisas están numeradas las muestras. Cada 8 muestras representan un chip y cada 64 chips representan un bit de información. De esta forma, sobre 512 muestras tenemos representado un bit de información. En la gráfica siguiente tenemos la fase filtrada con el filtro coseno alzado. Los saltos son producidos por que la fase esta representada entre $-p$ y p .



Es en el ángulo (fase) donde una señal QPSK envía la señal de información
El modulo antes y después de ser filtrada la señal en banda base es:

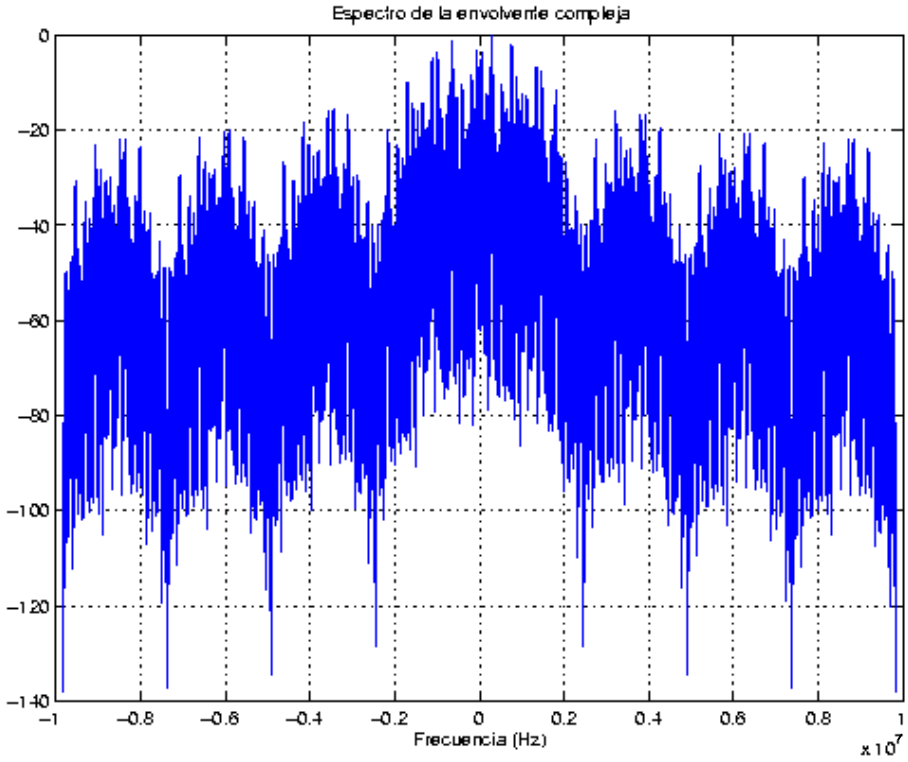


En las graficas hay representadas 512 muestras, las cuales y teniendo en cuenta la tasa de muestreo de 9.8302 Msps equivalen a 64 chips de la secuencia ensanchada de voz codificada a 19.2 kbps. Ya que cada bit de información es ensanchado sobre 64 chips en las graficas esta representado un bit de información codificada.

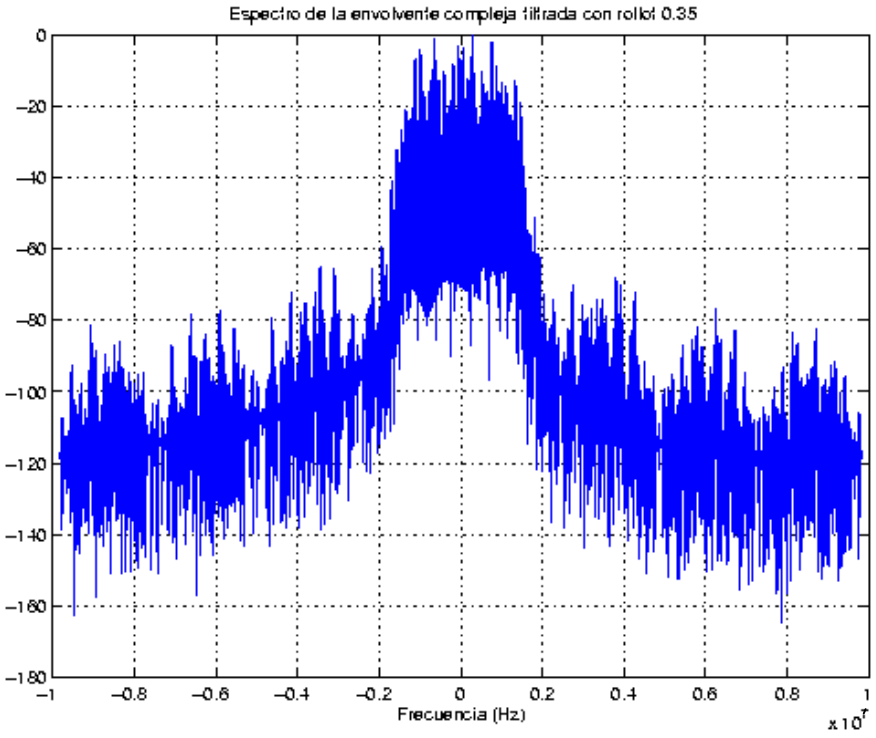


En estas últimas graficas están representadas las constelaciones antes y despues del filtrado. En la gráfica primera las transiciones son rectas que unen cada uno de los puntos que forman la constelación, mientras que una vez filtrado, estas transiciones difieren en función de los estados por los que haya pasado antes la señal. Observamos como los caminos son simétricos con respecto al centro.

El espectro de estas dos señales antes y después del filtro conformador se representa en las figuras siguientes en unidades logarítmicas:



El espectro de la señal conformada es:



Una vez procesada la señal (codificación, ensanchado, filtrado) la señal se encuentra en disposición de ser modificada por el modulador para una vez elevada en frecuencia ser capaz de poder propagarse por el medio apropiado, en este caso el interfaz aire. Sin embargo y apoyándonos en los modelos equivalentes paso bajo podemos evitar esta etapa la cual conllevaría una elevadísima tasa de muestreo que haría inviable este tipo de simulación.

El canal en el cual se va a propagar la señal esta caracterizado, como ya hemos expuesto de forma teórica en otros capítulos y experimentalmente, por desvanecimientos que pueden dar la sensación de aleatoriedad y que vienen determinados por el efecto multitrayecto que definen a este tipo de entornos. Para su simulación se proponen una serie de modelos sugeridos por los organismos reguladores de los sistemas de telecomunicaciones inalámbricos.

Estos modelos proponen sistemas por etapas que como ya se ha visto nos posibilitan simular los fenómenos que caracterizan estos sistemas. En nuestra simulación vamos a utilizar un modelo con tres etapas definidas en la siguiente tabla:

Entorno	t_1(ns)	t_2(ns)	t_3(ns)	Doppler	Doppler BW
Peatón	0	1500	14500	Plano	12 Hz
Vehículo	0	1500	15500	Jakes	180 Hz
Potencia de los rayos	0 dB	-3 dB	-6 dB		

Este modelo de tres etapas se implementa en la función canal.m.

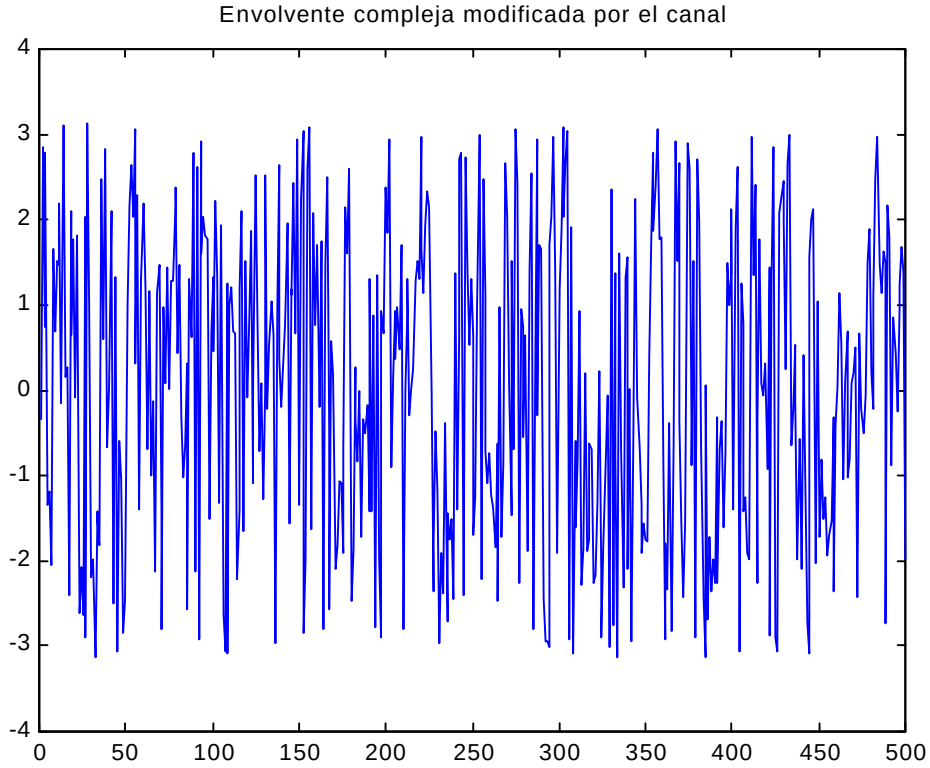
```
function h=canal(fm,L,sem,tipo)

% Funcion en la que se implementa el modelo de taps
%
%              h=canal(fm,L,sem,tipo)
%
% fm: maxima desviacion de la frecuencia Doppler.
% L: longitud de la secuencia de entrada.
% sem: semilla deseada para la simulacion.


N=1024;          % Numero de muestras que van a tomarse en la generacion
                 % del espectro.
T=L/1228.8e3;
delta=1/T;

rt=[0 1500 15000]*1e-9;                % Retraso de cada tap.PCS


arp=[0 -3 -6];               % Pot. medias relativas de cada tap. PCS
[S,si]=doppler(1,fm,N,delta);

lt=length(rt);               % numero de taps distintos de cero del
                             % modelo.
h=zeros(lt,L+1);
if(tipo==1)
   a=zeros(lt,2);
   for (i=1:lt)
```

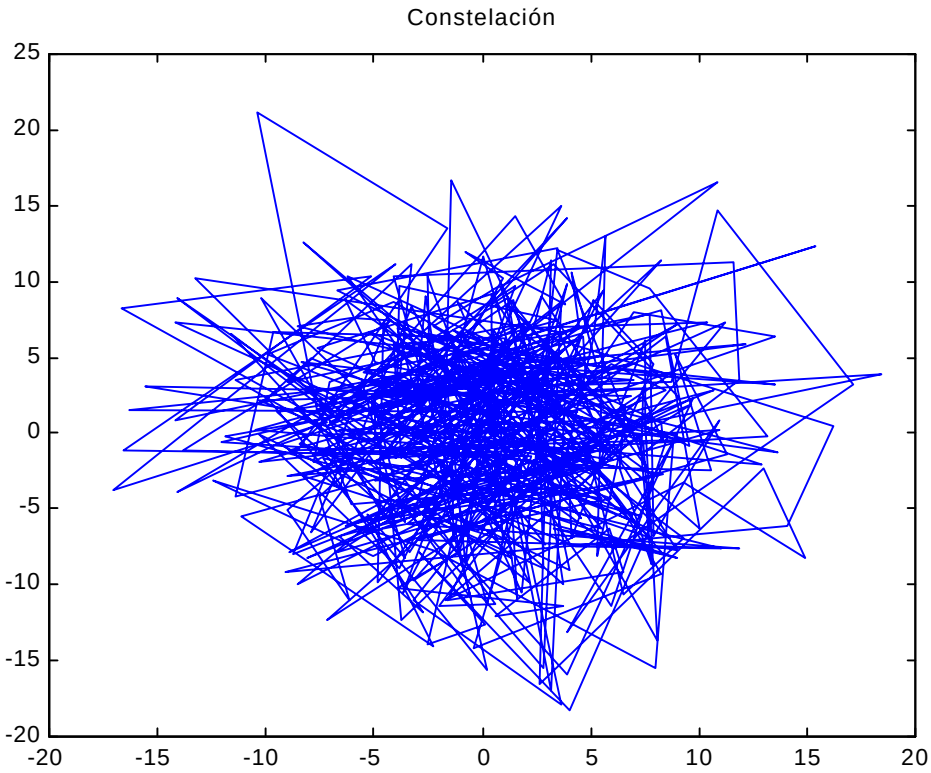
```
      a(i,:)=raych(10^(arp(i)/10),si,1,sem+2*i);
      h(i,:)=a(i,1);
   end
else
   for (i=1:lt)
       h(i,:)=raych(10^(arp(i)/10),si,L,sem+2*i);
    end
end
```

En estos modelos se tiene en cuenta el efecto Doppler que introduce el terminal portátil en su movimiento relativo con respecto a la estación base. Esto esta implementado mediante un filtro con una determinada forma y que esta caracterizado por un ancho proporcional a la velocidad que posee la estación móvil. Esto junto a la atenuación define de forma completa el modelo de etapas con retraso que vamos a utilizar. Existen y se pueden encontrar modelos como ya he comentado en las recomendaciones de cada sistema. Al ser la tasa de muestreo 9.8304 Msps, el tiempo entre dos muestras de nuestra señal es de 100 ns no existe ningún problema ya que el intervalo de muestreo es menor que los retrasos entre etapas, rt=[0 1500 15000]*1e-9 ns, y como consecuencia no es necesario establecer un muestreo más fino para que se pueda retrasar en cada etapa un numero fijo de muestras.

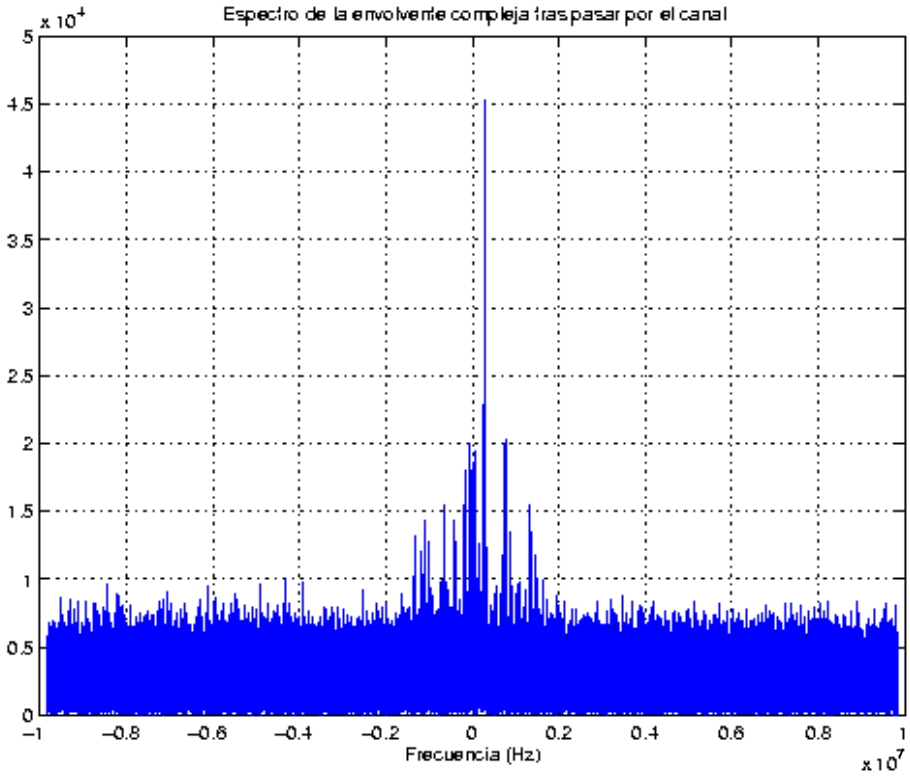
Tras pasar por el modelo discreto de etapas a la señal se le añade ruido gaussiano blanco antes de ser recogida por el receptor. En el caso de las gráficas mostradas la relación senal a ruido es -10 dB.

El aspecto de la señal antes de entrar en el receptor así como la forma de su constelación es:

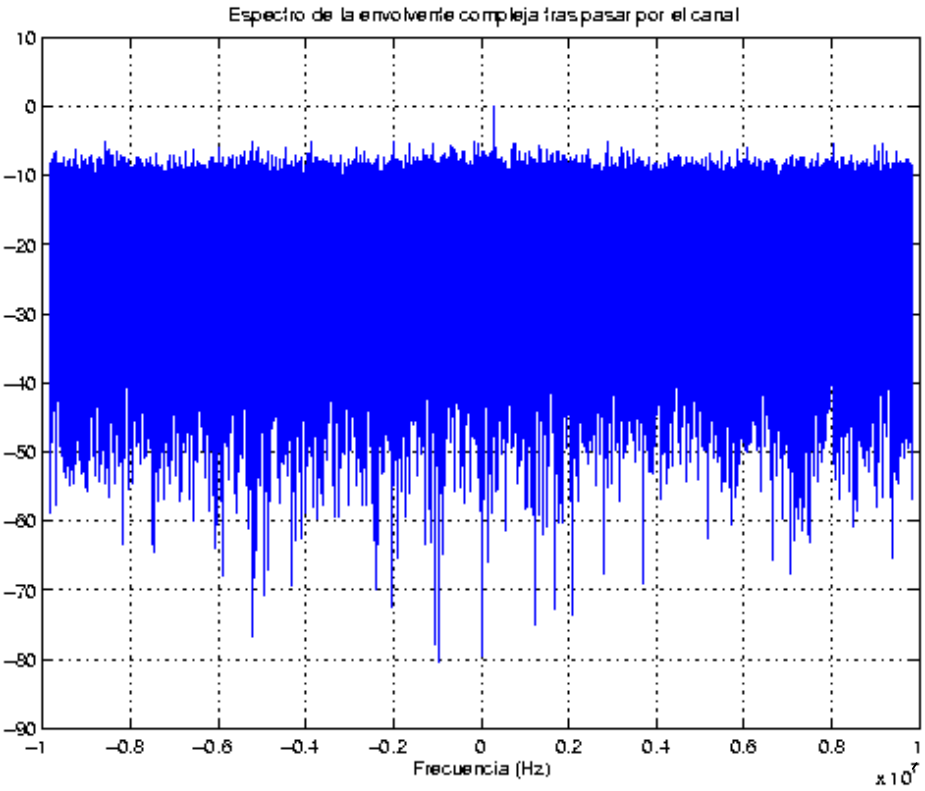




El espectro de la señal con ruido aditivo es:



Si representamos en escalas logarítmicas obtenemos:



El nivel uniforme representado junto a la señal representa la relación de ruido con respecto a la señal que se da en este entorno. Este ruido puede ser debido a señales que con sistema CDMA utilizan la misma frecuencia para propagarse, ya que tras la recepción cualquier señal que no estuviera ensanchada con el código adecuado será vista por el sistema como ruido. Por lo tanto no debemos considerar en nuestro estudio el efecto que han de tener las demás señales ensanchadas con distinto código antes de ser tratadas por el receptor sino considerar la potencia que dañara nuestra señal tras ser modificada esta señal por el receptor y como consecuencia atenuarla en tanto en cuanto la señal correcta tendrá de ganancia. Esta ganancia en un sistema CDMA esta determinada por la relación de que existe entre el ancho de banda de voz y el ancho de banda una vez ensanchado.

En el caso IS-95 donde la señal de información esta codificada a una tasa de 19.2 kbps y donde la tasa de chip es 1.2288 Mbps la ganancia es:

$$G = \frac{1.2288\ Mbps}{19.2\,kbps} = 64 = 18.06\,dB$$

En la ganancia del sistema influye el tipo de receptor que se utilice. En este caso hemos utilizado un receptor RAKE que ayudándose de las componentes multicamino las trata de forma constructiva y teniendo mayor gancia que la tendria el sistema por el ensanchamiento en si mismo. El receptor se implementa en la función decisor21sg.m.

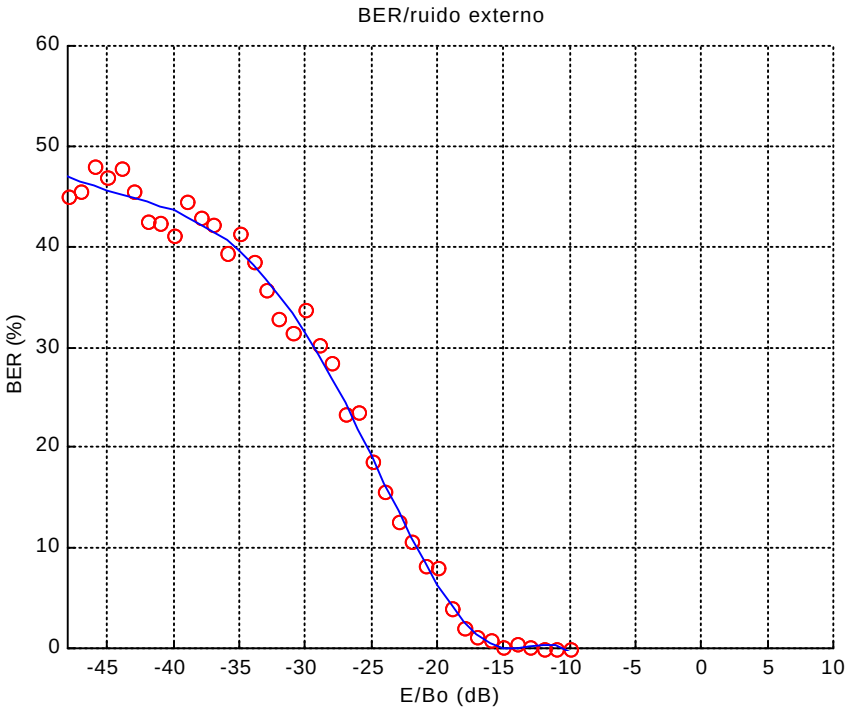
```
function [ber,bits]=decisor21sg(R,M,alfa,y,bi,rt)

% Decisor duro
%
%        function [ber,bits]=decisor21sg(R,M,alfa,y,bi)
%
% ber: Bit error rate.
% M: numero de muestras que tiene cada bit.
% alfa: Factor de rolloff del pulso coseno alzado.
% y: Señal que obtenemos del canal.
% bi: Secuencia de bits original para poder deducir ber.
% rt: Numero de caminos con su retraso necesario para el detector de
% Rake.


[num,den]=rcosine(R,M*R,'sqrt',0.35);
r=filter(num,den,y);
n=6;
p1=[1,6];
p2=[1,2,5,6];
pn1=pnsec(n,p1);
pn2=pnsec(n,p2);
ind=find(pn1==0);       % Convertimos las secuencias aleatorias en
bipolares.
pn1(ind)=-1;
ind=find(pn2==0);
pn2(ind)=-1;
W=walsh(64);
wpn=W(2,:);
ind=find(wpn==0);
wpn(ind)=-1;
biw=zeros(0,0);
```

```
xi=zeros(1,length(pn1)*M);
xq=zeros(1,length(pn2)*M);
for(i=1:length(pn1))
xi(M*(i-1)+1:M*i)=pn1(i)*wpn(i);                    % Se intercalan M-1
ceros entre cada dato.
xq(M*(i-1)+1:M*i)=pn2(i)*wpn(i);                    % Se intercalan M-1
ceros entre cada dato.
end;

ln=length(rt);
phi=2*pi*R*rt/3e8;
del=2*3/R;                 % Retraso que imponen los dos filtros cosenos
alzados.
im=1/(R*M);
sc=zeros(ln,length(r));
sd=zeros(ln,length(r));
tmax=round((del+rt(length(rt)))/im);
suma=zeros(1,length(r)-tmax);
sm=length(suma);
t=round((del+rt)/im);

for(i=1:ln)                 % El filtro coseno alzado tiene retraso 3/R
cada uno.
   sc(i,:)=r*exp(-j*phi(i));
   for(k=1:floor(sm/length(xi)))
      sd(i,(k-1)*length(xi)+1:k*length(xi))=sc(i,(k-
1)*length(xi)+t(i)+1:k*length(xi)+t(i)).*(xi-j*xq);
   end
   suma=suma+sd(i,1:sm);
end

% Un filtro adaptado a un pulso rectangular es un integrador.
lon=floor(sm/length(xi));
bits=zeros(1,lon);
for(j=1:lon)
   aux=cumsum(suma((j-1)*length(xi)+1:length(xi)*j));
   baux=real(aux(length(aux)));
   bits(j)=baux/abs(baux);
end

bic=bi(1:length(bits));

      ber=sum(abs(bic-bits))/(2*length(bits));
```

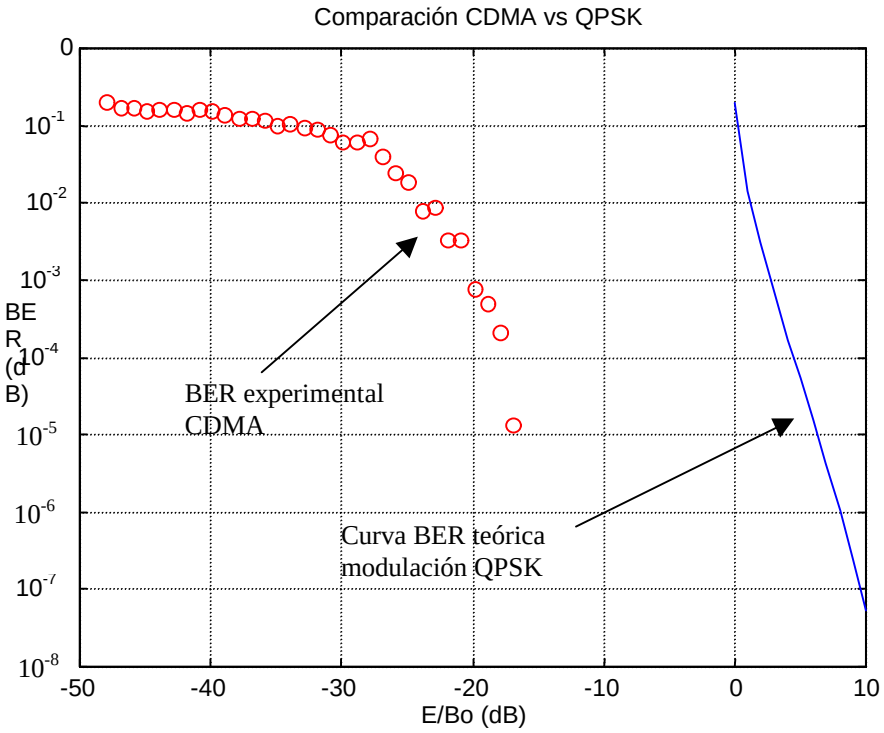
La ganancia por ensanchamiento así como la utilización del receptor RAKE hace que este sistema sea altamente robusto frente a interferencias aun siendo estas potentes y localizadas en el espectro ya que una vez las coge el receptor estas son ensanchadas y convertidas en ruido blanco gaussiano y con la potencia repartida en el espectro que hemos elegido para ensanchar.

De esta forma en la siguiente grafica podemos ver el comportamiento del sistema ante el ruido externo sin olvidar que las señales que cohabitan en el sistema pueden ser consideradas ruido tras ser tratadas por el receptor.

En la siguiente gráfica vemos el comportamiento del sistema ante el ruido externo. Corresponde a un modelo de 3 etapas que caracteriza un entorno del tipo peatón:



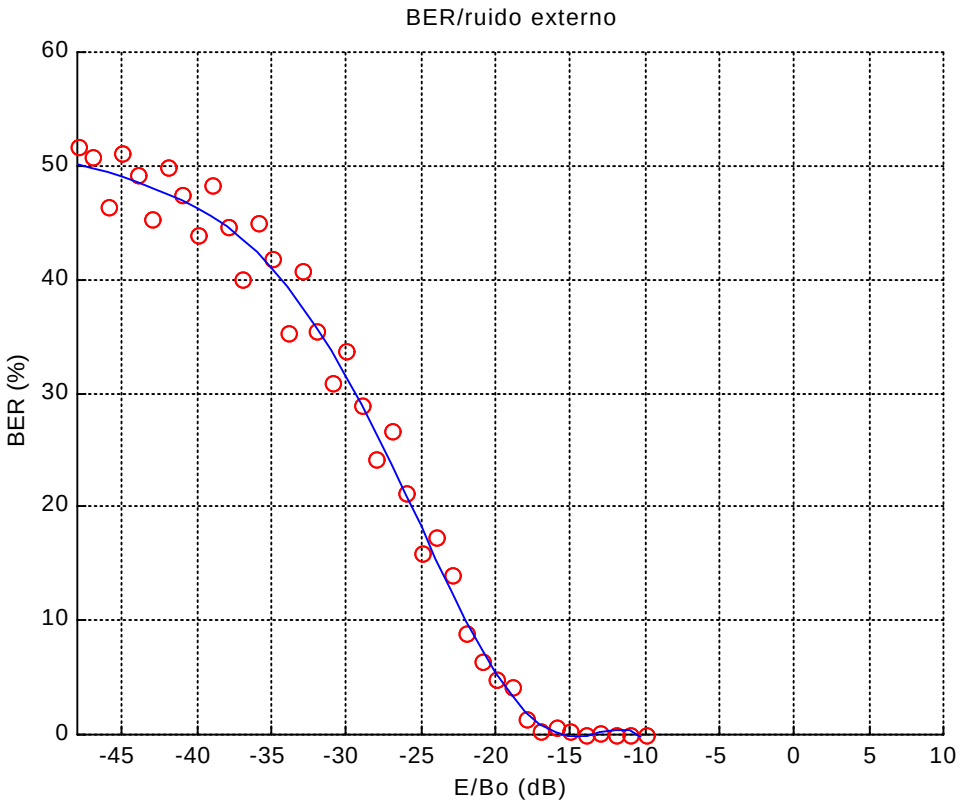
Podemos comparar esta la relación de señal a ruido con la que teóricamente presentaría un sistema con modulación QPSK.



Para el sistema de 2º generación GSM existe un modelo definido en las recomendaciones de ETSI que establece el modelo más perjudicial y que es utilizado para el entrenamiento del ecualizador que forma parte de este sistema. Este modelo esta formado por los siguientes datos:

Numero de rayo	Retraso (ms)	Potencia media (dB)	Doppler
1	0.0	0.0	Jake
2	3.2	0.0	Jake
3	6.4	0.0	Jake
4	9.6	0.0	Jake
5	12.8	0.0	Jake
6	16.0	0.0	Jake

y establece una curva ruido/BER(%) de la forma:



La función que hemos utilizado para comparar la señal binaria origen y la que obtenemos en el receptor es serror.m que nos proporcionará además de la BER encontrada en el sistema la secuencia de error que caracteriza a éste y lo que utilizaremos para echar a andar el algoritmo de Baum-Welch.

```
function [sec]=serror(bit,bite)

% Funcion que comparando las secuencias de bits transmitidas y detectadas
% saca una secuencia en la que cada componente del vector resultado es la
% distancia entre dos errores
% consecutivos.
%
%     [sec]=serror(bit,bite)
%
% sec: vector salida.
% bit: vector de bits correctos.
% bite: vector de bits erroneos.

l1=length(bit);
l2=length(bite);
if(l1>=l2)
   ln=l2;
else
   ln=l1;
end;
aux=zeros(1,ln);
k=0;
j=1;
```

```
for(i=1:ln)
   if(bit(i)==bite(i))
      k=k+1;
   else
      aux(j)=k;
      j=j+1;
      k=0;
   end
end
aux(j)=k;
sec=aux(1:j);
```