

**ANÁLISIS COMPARATIVO ENTRE UN SISTEMA
SISO Y UN SISTEMA MIMO**

ANEXOS

**CARLOS FRANCISCO PLAZA VÁSQUEZ
EDWIN LEONARDO REY MUÑOZ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELECOMUNICACIONES
GRUPO I+D EN NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES
POPAYÁN
2008**

**ANÁLISIS COMPARATIVO ENTRE UN SISTEMA SISO Y UN SISTEMA
MIMO**



ANEXOS

**CARLOS FRANCISCO PLAZA VÁSQUEZ
EDWIN LEONARDO REY MUÑOZ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELECOMUNICACIONES
GRUPO I+D EN NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES
POPAYÁN
2008**

TABLA DE CONTENIDO

| | |
|--|---|
| ANEXO A. ESQUEMAS DE RECEPCIÓN PARA SISTEMAS CON MULTIPLEXACIÓN ESPACIAL | 1 |
| A.1 RECEPTORES LINEALES | 1 |
| A.1.1 Receptor Zero Forcing (ZF) | 1 |
| A.1.2 Receptor de Error Cuadrático Medio Mínimo (Minimum Mean-Square Error Receiver, MMSE) | 1 |
| A.2 RECEPTORES NO LINEALES..... | 2 |
| A.2.1 Receptor de Máxima Verosimilitud (Maximum Likelihood, ML) | 2 |
| A.2.2 Receptor V-BLAST | 2 |
| ANEXO B. MÉTODOS DE COMBINACIÓN EN EL RECEPTOR | 4 |
| B.1 COMBINACIÓN POR SELECCIÓN (SELECTION COMBINING, SC) | 4 |
| B.2 REALIMENTACIÓN (FEEDBACK) | 5 |
| B.3 COMBINACIÓN POR IGUAL GANANCIA (EQUAL GAIN COMBINING, EGC)..... | 5 |
| B.4 COMBINACIÓN POR MÁXIMA RELACION (MAXIMUM RATIO COMBINING, MRC)..... | 6 |
| ANEXO C. DOCUMENTACIÓN DE LA SIMULACIÓN | 8 |

INDICE DE FIGURAS

| | |
|--|---|
| Figura B-1. Esquema de Combinación de Diversidad : Combinación por Selección SC..... | 5 |
| Figura B-2. Esquema de Combinación de Diversidad: Combinación por Igual Ganancia EGC..... | 6 |
| Figura A-3. Esquema de Combinación de Diversidad: Combinación por Máxima Relación MRC..... | 7 |

ANEXO A. EQUEMAS DE RECEPCIÓN PARA SISTEMAS CON MULTIPLEXACIÓN ESPACIAL

Las arquitecturas en recepción compiten de acuerdo al desempeño en el procesamiento de la señal, es así como los receptores para sistemas con multiplexación espacial pueden ser divididos en tres clases: El receptor de máxima verosimilitud (*The Maximum Likelihood Receiver*, ML), el receptor lineal (*Zero Forcing*, ZF), el receptor de error cuadrático medio (*Minimum Mean Square Error*, MMSE) y el receptor V-BLAST [1][2][3].

A.1 RECEPTORES LINEALES

A.1.1 Receptor Zero Forcing (ZF)

El receptor lineal simple consiste en una matriz que se encarga de realizar el filtrado a los datos transmitidos, la cual básicamente invierte la matriz de transferencia del canal, asumiendo que la matriz de canal H es cuadrada e invertible. Bajo esta suposición, una estimación del vector de símbolos de datos transmitidos s se obtiene como:

$$s = H^{-1}r \quad (A.1)$$

donde s es el vector de símbolos transmitidos, H^{-1} es la matriz inversa del canal H y r los datos transmitidos.

Es así como el receptor lineal ZF separa perfectamente las señales convirtiendo el canal en sub-canales paralelos, logrando la eliminación de la interferencia Co-canal s_j ($j=0,1,\dots,Mt-1$). Sin embargo, el receptor ZF amplifica la potencia de ruido, y cuando el canal presenta un desvanecimiento profundo, es decir unas malas condiciones del canal y un valor de relación señal a ruido pequeño, se produce un pobre desempeño del receptor [1][2][3].

A.1.2 Receptor de Error Cuadrático Medio Mínimo (Minimum Mean-Square Error Receiver, MMSE)

El receptor MMSE presenta una alternativa para solucionar la interferencia Co-canal y la amplificación de ruido presente en el receptor ZF, dado que minimiza el total de errores generados por el ruido y la interferencia mutua entre señales Co-canal. Un estimado del vector de símbolos transmitidos s , se obtiene de acuerdo con:

$$s = \frac{\rho}{N_t} H^H \left(\sigma_n^2 I_{N_r} + \frac{\rho}{N_t} H H^H \right)^{-1} r \quad (A.2)$$

donde se asume que $E\{ss^H\} = \frac{\rho}{N_t} I_{N_t}$ y $E\{nn^H\} = \sigma^2 I_{N_r}$. El receptor MMSE es menos sensible al ruido con lo cual se reduce la calidad en la separación de la señal. Es decir no se elimina totalmente la interferencia Co-canal. En un caso de una SNR alta se tiene que $\sigma_n^2 = 0$, y el receptor MMSE se convierte en un receptor ZF [1][2][3].

A.2 RECEPTORES NO LINEALES

A.2.1 Receptor de Máxima Verosimilitud (Maximum Likelihood, ML)

El receptor ML, presenta un buen desempeño en términos de su tasa de error, teniendo en cuenta que éste, maneja alta complejidad computacional debido a las estimaciones que tiene que realizar para detectar el símbolo transmitido, dicha complejidad aumenta de manera exponencial con el número de antenas transmisoras N_t y el orden de modulación M_c . El receptor ML es un método que compara la señal recibida con todo el vector posible de señal transmitida que se ve afectado por el canal y estima el vector de símbolo transmitido s de acuerdo al principio de máxima verosimilitud, mostrado como:

$$s' = \arg_s \min \|r - Hs\|^2 \quad (\text{A.3})$$

Donde s' es el vector de símbolo estimado. El receptor ML busca a través de todo el vector recibido el que tiene más probabilidad de haber sido transmitido, sin embargo, el principal inconveniente de esta regla de decisión se basa en su número de métricas a evaluar ya que se calculan como $M_c^{N_t}$ donde N_t es el número de antenas transmisoras y M_c es el orden de modulación, donde se observa que para valores pequeños de N_t y M_c el costo computacional es asumible, pero para valores altos de N_t y M_c se aumenta la complejidad de éste método, considerándose muy difícil la implementación en la práctica. Por ejemplo en caso de tener 4 antenas transmisoras y una modulación 16 QAM, requiere de un total de $16^4 = 65.536$ comparaciones por cada vector de símbolos transmitido. Es así como se muestra la complejidad del receptor ML y puede no ser aplicable [1][2][3].

A.2.2 Receptor V-BLAST

El receptor V-BLAST surge como una alternativa para los receptores lineales ZF, MMSE, que sufren de alta sensibilidad a la interferencia Co-canal, sin embargo este receptor generalmente se combina con un algoritmo de cancelación de interferencia sucesiva (SIC) (que consiste en eliminar de la señal recibida las señales que ya han sido detectadas).

Este esquema de detección no lineal, V-BLAST, en lugar de solo unir las señales transmitidas, primero detecta la señal transmitida más fuerte,

cancela el efecto de ésta señal sobre las demás señales recibidas y luego procede a detectar la señal más fuerte del resto de las señales transmitidas, haciendo este proceso de manera sucesiva con todas las señales.

El orden de detección óptimo tiene una estrategia de anulación y cancelación de la señal más fuerte a la señal más débil. A continuación se describe brevemente los principales pasos del algoritmo V-BLAST asumiendo que el canal es conocido en el receptor.

Anulación (Nulling): Realiza la estimación de la señal más fuerte transmitida por la anulación de todas las señales transmitidas más débiles. (Usando el criterio de ZF o MMSE).

Slicing: La señal estimada es detectada para obtener los bits de información. Detecta el valor de la señal más fuerte transmitida por la partición del valor de la constelación en la señal más cercana

Cancelación (Cancellation): Cancela el efecto de las señales más fuertes transmitidas detectadas por el vector de señal recibida, además reduce la complejidad de detección para las señales que permanentemente se están transmitiendo [1][2][3][9].

ANEXO B. MÉTODOS DE COMBINACIÓN EN EL RECEPTOR

Las técnicas utilizadas en recepción para recibir varias replicas de la misma señal, requiere alguna clase de combinación con el fin de obtener una sola representación de la señal deseada. Hay diferentes métodos de combinación dependiendo del nivel del conocimiento del canal en el receptor. Entre los más comunes se tienen: la combinación por selección (*Selection Combining, SC*), combinación por realimentación (*Feedback*), combinación por igual ganancia (*Equal gain combining, EGC*) y la combinación por máxima relación (*Maximal ratio combining, MRC*) [8][9].

B.1 COMBINACIÓN POR SELECCIÓN (SELECTION COMBINING, SC)

En este método de combinación, el receptor monitorea el nivel de señal recibida en las diferentes ramas que se generan por el uso de M antenas, usando una compuerta lógica (conmutación lógica, swith lógico). Si la señal presenta un nivel por debajo de un umbral predefinido, el conmutador cambia de trayectoria para seleccionar otra antena y realizar el mismo análisis a la señal, evitando así, que las señales con bajo nivel (nivel inferior al umbral predefinido) puedan sumarse de forma destructiva.

Una de las ventajas que presenta esta combinación, es su relativa facilidad de implementación debido a que analiza una a una la señal recibida por cada antena. Sin embargo, entre los inconvenientes que afronta es que el conmutador actúa cuando la señal ya ha sufrido algún tipo de desvanecimiento, otro inconveniente se encuentra en la posibilidad que la señal de la otra antena seleccionada, tenga un nivel más bajo que el umbral definido, por esto, en canales con muy poca SNR no se pueden realizar análisis con precisión ya que esto contribuye con el ruido en recepción.

La forma como opera esta combinación por selección se describe a continuación, la rama actual permanece seleccionada hasta que la energía recibida sea inferior a cierto umbral. Usualmente el indicador de intensidad de la señal recibida (*Received Signal Strength Indicador, RSSI*) es el encargado de analizar estas fallas. Luego se selecciona la siguiente rama dependiendo del umbral. En la Figura B-1 se muestra un esquema de combinación por selección [4][5][6][7].

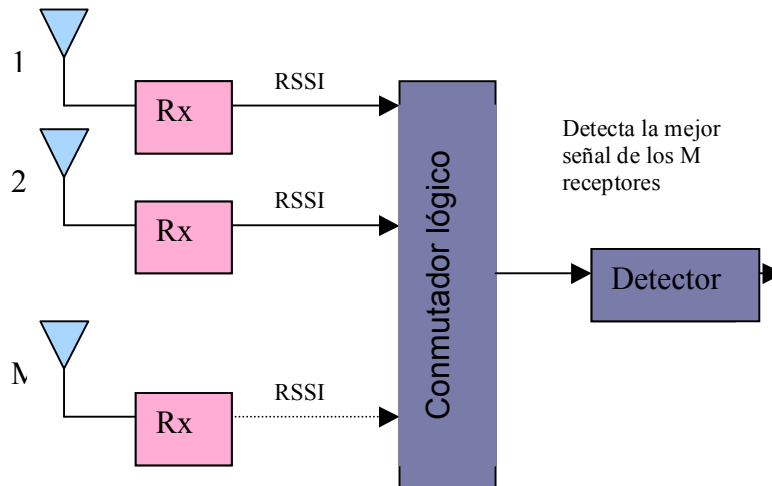


Figura B-1. Esquema de Combinación de Diversidad: Combinación por Selección SC

En resumen se tiene

- El sistema es costoso porque es necesario analizar todas las ramas.
- No es ideal el uso de RSSI como indicador de calidad porque se ve afectado por la interferencia. [5][6][7].
- En SC no se requiere de ninguna información del efecto introducido en la señal al atravesar el canal [4].

B.2 REALIMENTACIÓN (FEEDBACK)

Este tipo de diversidad conocida como feedback o realimentación, está basado en el análisis de M señales en una secuencia fija hasta que se encuentra una que sobrepase el umbral definido, volviéndose la señal escogida hasta que disminuye por debajo del umbral establecido y el proceso de análisis comienza nuevamente. El desempeño en la tasa de error de esta técnica es inferior a los otros métodos razón por la cual su utilización no es muy común [8][6].

B.3 COMBINACIÓN POR IGUAL GANANCIA (EQUAL GAIN COMBINING, EGC)

En este método cada señal de las M ramas de diversidad, tiene un receptor correspondiente que le introduce un compensador de fase $\phi_1, \phi_2, \dots, \phi_M$, buscando con esto que las señales tengan la misma fase y puedan ser sumadas de manera constructiva. Se le conoce como combinador de igual ganancia porque las amplitudes de todas las ramas tienen el mismo peso o

valor. En la Figura B-2 se observa un esquema de este tipo de combinación.

La combinación por igual ganancia presenta un mejor desempeño que la combinación por selección porque todas las ramas son usadas simultáneamente, a diferencia de SC que la complejidad del receptor es dependiente del número de caminos que van a ser resueltos [6][7][9].

- Todas las ramas presentan la misma fase y posteriormente son sumadas.
- En EGC se necesita el conocimiento del cambio de fase introducido por el desvanecimiento en el canal, es así como solo el sistema que use detección coherente puede utilizarlo; sin embargo, ésta es una seria limitación porque en general es complicado seguir la trayectoria de las fases de la señal recibida [4].

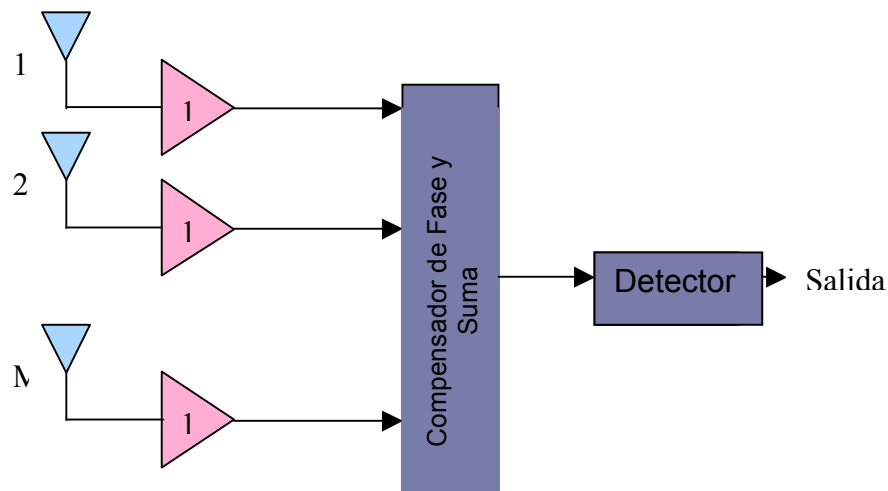


Figura B-2. Esquema de Combinación de Diversidad: Combinación por Igual Ganancia EGC

B.4 COMBINACIÓN POR MAXIMA RELACION (MAXIMUM RATIO COMBINING, MRC)

Método similar al EGC porque se aplica la misma compensación de fase $\phi_1, \phi_2, \dots, \phi_M$, haciendo que las señales tengan la misma fase para luego ser sumadas de manera constructiva. Este método difiere de EGC debido a que cada rama es pesada por un factor k_1, k_2, \dots, k_M , el cual depende de la SNR de cada señal, es decir el efecto de este factor de ajuste le da un mayor peso a la señal más fuerte.

La dificultad en la implementación de MRC es asumir los factores de peso y los cambios de fase como conocidos con exactitud. Es una tarea compleja

llevar a cabo un receptor digital, considerando que los factores de peso y cambios de fase son en tiempo real. La complejidad del receptor es dependiente del número de caminos a resolver, en la Figura B-3, se muestra el esquema de combinación de diversidad MRC [5][6][7].

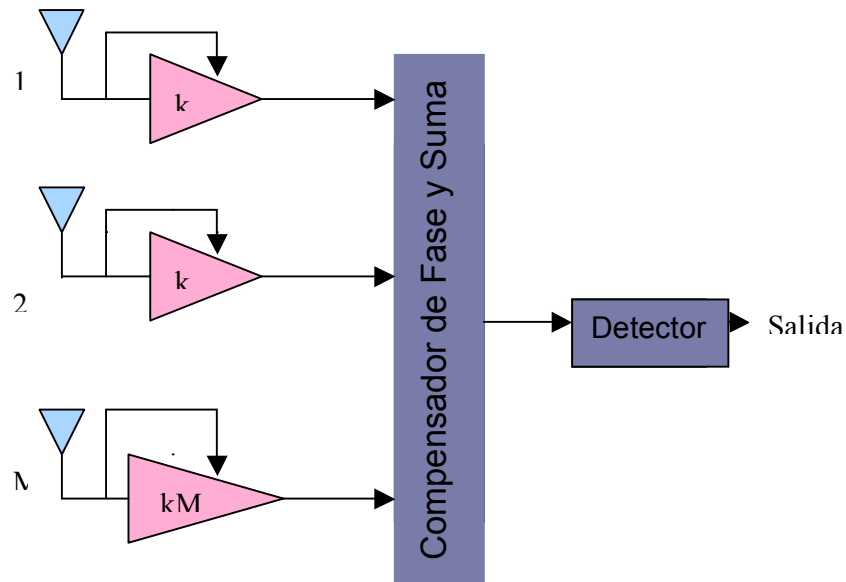


Figura A-3. Esquema de Combinación de Diversidad: Combinación por Máxima Relación MRC

En resumen

- A cada rama se le asigna un peso antes de realizar la suma en proporción a su propia relación señal a ruido (SNR).
- Tiene un mejor desempeño que EGC, pero requiere de cierta complejidad para estimar la SNR.
- Este método necesita el conocimiento de amplitudes y fases de todos los canales involucrados, y requiere de un análisis y rastreo de todos los componentes [9].

ANEXO C. DOCUMENTACIÓN DE LA SIMULACIÓN

En este trabajo se realizaron básicamente tres grupos de simulación conformados en un programa principal que llama a los otros dos grupos donde se puede calcular la capacidad y la BER de los sistemas.

- Un sistema SISO – Con y sin Codificación Convolutiva
- Un sistema MIMO - Con y sin Codificación Convolutiva
- Un sistema principal que integra los dos sistemas MIMO-SISO.

Las simulaciones se desarrollaron con las características establecidas en el capítulo tres

CODIGO MATLAB

Descripción de las funciones utilizadas para la simulación.

Generación de los datos a transmitir.

La función "randint" genera K datos binarios aleatorios.

- En SISO la función es:

| | |
|-------------------------------------|---|
| <pre>K; x = randint(K,1);</pre> | <pre>%Número de datos a transmitir %la función randint, genera K datos aleatorios entre 0 y 1</pre> |
|-------------------------------------|---|

- En MIMO la función es:

| | |
|---------------------------------|--|
| <pre>data = randint(K,1);</pre> | <pre>%la función randint genera K datos aleatorios entre 0 y 1</pre> |
|---------------------------------|--|

Codificación de los datos

- La función "coder" realiza la codificación para SISO, la cual recibe los datos generados "x" y devuelve los datos codificados en "xcod".

```
function xcod= coder(x);           % Función de Codificación
xsincodif = x;                    % Se asignan los datos a
                                   xsincodif
codconvolu = xconvol(xsincodif); % llama a la función codificación
                                   convolucional
xcod = codconvolu;                % Los datos codificados se asignan a
                                   xcod
```

- La función "codmimo", se encarga de realizar la modulación y codificación de los datos para MIMO, la cual recibe los datos binarios aleatorios "data" y los organiza en la matriz de *Alamouti* para el proceso de codificación.

```
function d=codmimo(data) %Función de codificación
                           para MIMO que recibe los datos de data
[c1,c2]= $\begin{pmatrix} c_1 & c_2 \\ -c_2^* & c_1^* \end{pmatrix}$  %matriz de Alamouti.
d = [s1 s2; -conj(s2) conj(s1)]; %Es la matriz de Alamouti que se encarga
                                   del proceso de codificación de los datos, y
                                   los devuelve por la variable d
```

Modulación de los datos (sean codificados o sin codificar)

- La función "modPSK" se encarga de realizar la modulación para SISO utilizando la función "pskmod" de MATLAB, que recibe la información "xcod" codificada o "x" sin codificar, en este caso "xcod", y el orden de modulación "M".

```
xmod=modPSK(xcod,M);    %Función que Modula los datos codificados xcod
                        %con un orden de modulación M, que es igual
                        %a 2 para BPSK y los entrega a la variable
                        %xmod
```

- La función "codmimo" realiza el proceso de modulación y codificación de los datos para MIMO, que recibe los datos "data" y los devuelve en "d", modulados y codificados porque es una función que integra estos dos procesos. A continuación solo se muestra el proceso de modulación.

```
function d=codmimo(data) %    %Función que realiza el proceso de
                             %modulación
symb_one = 1;                %Cuando llega un símbolo 1 le asigna una
                             %valor 1 por ser una modulación BPSK
symb_zero = -1;             %Cuando llega un símbolo 0 le asigna una
                             %valor de -1 por ser una modulación BPSK

bits(1)=data(1,1);
bits(2)=data(1,2);

if bits(1) == 0
    s1 = symb_zero;
else
    s1 = symb_one;
end

if bits(2) == 0
    s2 = symb_zero;
else
    s2 = symb_one;
end
```

Adición de ruido blanco gaussiano a los datos en recepción para SISO y para MIMO

La función "awgn" adiciona ruido blanco dependiendo de la variación de la SNR en dB a los datos que ya han pasado por el canal, es decir a la señal transmitida "senalTx".

```
senal(:, :, band) = awgn(senalTx, snr(g), 'measured'); %para MIMO
rxsenal = awgn(txsenal, snr(oye), 'measured'); %Para SISO
```

En recepción se utilizan funciones que ejecuten el proceso inverso a lo realizado en transmisión para recuperar los datos como la demodulación y decodificación.

Para el proceso de demodulación de los datos transmitidos se utilizan las siguientes funciones:

- Para SISO la función "pskdemod" propia de la herramienta, realiza el proceso de demodulación de los datos recibidos "rxsenal" con el orden de modulación establecido por "M".

```
demodpsk = pskdemod(rxsenal, M); %la función pskdemod demodula la señal
recibida "rxsenal" respecto al orden de
modulación M=2, para BPSK
```

- Para MIMO la función "decodmimo" se encarga de hacer la demodulación y la decodificación de los datos, la cual recibe la "senalr" después de pasar por el canal.

```

function dc= decodmimo(H,senalr,t)
symb_zero = -1;
symb_one = 1;
senal=senalr;
He=H;

    for f=1:t;

        y=senal(:,f);

        %Combinador de la Matriz de Alamouti

        s0 = conj(He(1,1))*y(1,1) + He(1,2)*conj(y(1,2)) + conj(He(2,1))*y(2,1) +
        He(2,2)*conj(y(2,2));
        s1 = conj(He(1,2))*y(1,1) - He(1,1)*conj(y(1,2)) + conj(He(2,2))*y(2,1) -
        He(2,1)*conj(y(2,2));

        %demodulador BPSK

        if real (s0) <0
            dec(1,1) =0;
        else
            dec(1,1)=1;
        end% end s0 detector
        if real (s1) < 0;
            dec(1,2) = 0;
        else
            dec(1,2) = 1 ;
        end % end s0 detector
        dc(:,f)=dec();
    end

```

Para el proceso de Decodificación de los datos transmitidos.

- La función "decoder" decodifica los datos en SISO utilizando la función "deconvol" que a su vez ejecuta una función propia de MATLAB para realizar el proceso de decodificación en la señal devolviéndolo en la variable "dxcod".

```

function dxcod= decoder(x);

xcodif = x; %Datos codificados demodulados
deccodconv= deconvol(xcodif);
dxcod = deccodconv; % en dxcod, se almacenan los datos
recuperados

```

- La función "decodmimo" se encarga del proceso de decodificación en MIMO de los datos recibidos "senalr", siendo ésta función la misma que se utiliza en el proceso de demodulación. A continuación se muestra solo

el proceso de decodificación realizado por el combinador usado por el esquema de *Alamouti*.

```
function dc= decodmimo(H,senالر,t) % decodifica_erey(H,senالر,t)
symb_zero = -1;
symb_one = 1;
senالر=senالر;
He=H;

    %Combinador de la Matriz de Alamouti

s0 = conj(He(1,1))*y(1,1) + He(1,2)*conj(y(1,2)) + conj(He(2,1))*y(2,1) +
He(2,2)*conj(y(2,2));
s1 = conj(He(1,2))*y(1,1) - He(1,1)*conj(y(1,2)) + conj(He(2,2))*y(2,1) -
He(2,1)*conj(y(2,2));
```

Las funciones que utilizaron los sistemas en la simulación fueron presentadas anteriormente por separado. A continuación, se observa cómo los dos sistemas se integran un solo archivo de MATLAB para realizar los análisis de los mismos.

Función principal (Archivo: MIMO_SISO.m)

```
close all
clear all
clc

K = 1e4; %Número de datos a transmitir
SNR_max = 15; %Máxima relación Señal a Ruido
snr=1:2:SNR_max;

seed = [98765 12345];
randn('state', seed(1)); rand('state', seed(2));

x1 = randint(K,1); %Generador de datos binarios
data1=x1;

H = (randn(2, 2) + j*randn(2,2));% Genera el Canal normal (AWGN)
Hn=H;
Hr=H/sqrt(2); %Genera el Canal Rayleigh

nt_V = [1 2];
nr_V = [1 2];
color = ['b';'r';'g';'k';'c'];
notation = ['-^';'->';'<-';'^-';'-s'];
cont=1;
cont1=3;
```



```

for (chan=1:2); % se selecciona el tipo de canal, 1 Rayleigh, 2 AWGN
    for (ant = 1 : 2); % se selecciona el tipo de sistema ant=1 SISO, ant=2 MIMO
        2x2
        if ant==1
            [BER,cap]=SISO(x1,Hn,Hr,chan); % Función sistema SISO
        end
        if ant==2
            [BER,cap]=MIMO(data1,Hn,Hr,chan); %Función sistema MIMO
        end
        end
        if chan==1
            f1=figure(1);
            set(gca,'yscale','log','xlim',[snr(1), snr(end)],'ylim',[1e-5 1]);
            semilogy(snr,BER,notation(cont,:), 'color',color(cont,:));
            hold on
            xlabel('SNR (dB)'); ylabel('BER');
            title('BER con modulación BPSK sobre canal Rayleigh');
            legend_str = []; % generan la leyenda de la grafica de la BER
            for ( i = 1 : length(nt_V))
                legend_str =[ legend_str ;...
                    {'nt = ',num2str(nt_V(i)), ', nr = ',num2str(nr_V(i))}];
            end
            legend(legend_str);
            grid on;

            f2=figure(2);
            plot(snr,cap,notation(cont,:), 'color',color(cont,:));
            hold on
            title('Capacidad con Canal Rayleigh');
            xlabel('SNR in dB');
            ylabel('Capacidad bits/s/Hz');
            legend_str = []; % generan la leyenda de la grafica de la BER
            for ( i = 1 : length(nt_V))
                legend_str =[ legend_str ;...
                    {'nt = ',num2str(nt_V(i)), ', nr = ',num2str(nr_V(i))}];
            end
            legend(legend_str);
            grid on;
            cont=cont+1;
        end
        if chan==2
            f3=figure(3);
            set(gca,'yscale','log','xlim',[snr(1), snr(end)],'ylim',[1e-5 1]);
            semilogy(snr,BER,notation(cont1,:), 'color',color(cont1,:));
            hold on
            xlabel('SNR (dB)'); ylabel('BER');
            title('BER con modulación BPSK sobre canal AWGN');
            legend('BER AWGN');
            legend_str = []; % generan la leyenda de la grafica de la BER
            for ( i = 1 : length(nt_V))
                legend_str =[ legend_str ;...
                    {'nt = ',num2str(nt_V(i)), ', nr = ',num2str(nr_V(i))}];
            end
            legend(legend_str);
            grid on;
        end
    end
end

```

```

f4=figure(4);
plot(snr,cap,notation(cont1,:), 'color',color(cont1,:));
hold on
title('Capacidad con Canal AWGN');
xlabel('SNR in dB');
ylabel('Capacidad bits/s/Hz');
legend_str = []; % generan la leyenda de la grafica de la BER
for(i = 1 : length(nt_V))
legend_str =[ legend_str ;...
{'nt = ',num2str(nt_V(i)), ', nr = ',num2str(nr_V(i))}];
end
legend(legend_str);
grid on;
cont1=cont1+1;
end
end %segundo for
end %primer for

```

Sistema SISO (Archivo: SISO.m)

```

function [BER,cap]=SISO(x1,Hn,Hr,chan)

M = 2;
k = log2(M);
Ntx=1;
Nrx=1;
SNR_max = 15;
snr=1:2:SNR_max;
IM=1;

%xxxxx Recibe los datos binarios generados en el programa principal xxxxxx

x=x1;

%xxxxxxxxxxxxxxxxxxxxxxxx Codificador Convolutacional xxxxxxxxxxxxxxxxxxxxxxxx

xcod=coder(x); % x es la entrada del codificador, función que se apaga o no
               % dependiendo de si el sistema va a utilizar codificación o
               no.

%xxxxxxxxxxxxxxxxxxxxxxxx Modulacion M-PSK xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xmod= pruebamodPSK(xcod,M); %recibe xcod si se usa codificación
                             % recibe x si no usa codificación

entracanal=xmod;             % AL canal se envia la señal
                             codificada y modulada

```

```

%xxxxxxxxxxxxxxxxxxxxx Canal Rayleigh o AWGN xxxxxxxxxxxxxxxxxxxxxxxx
if chan==1;
    H=Hr(2,1) % Canal Rayleigh si el canal escogido es 1 (chan=1)
else
    H=Hn(2,1) % Canal AWGN si el canal escogido es 2 (chan=2)
end

    txsenal=H*entracanal; %Señal transmita, cuando se adiciona el efecto del c
                        %canal

    for oye = 1:length(snr) % Ciclo que varia la relación señal a ruido

%xxxxxxxxx Adición de ruido AWGN en recepción xxxxxxxxxxxxxxxxxxxxxxxx

        rxsenal=awgn(txsenal,snr(oye),'measured');

%xxxxxxxxx Demodulación de los datos xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

        demodpsk = pskdemod(rxsenal,M);

%xxxxxxxxx Conversión de símbolo a Bit xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

        % Undo the bit-to-symbol mapping performed earlier.
        z = de2bi(demodpsk,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
        z = reshape(z.',prod(size(z)),1);

%xxxxxxxxxxxxxxxxxxxxx Decodificador Convolutacional xxxxxxxxxxxxxxxxxxxxxxxx

        dxcod=decoder(z); % Decodifica los datos recibidos que se activa o
                        % se desactiva dependiendo del sistema a
                        % estudiar con o sin codificación
        decod=dxcod; % si se le asigna z, no fue codificado
                    % si se le asigna dxcod fue decodificado

%xxxxxxxxxxxxxxxxxxxxx Cálculo de Numero de errores y BER xxxxxxxxxxxxxxxxxxxxxxxx

        bit_error(1,oye) = sum(x~=decod);
        BER=bit_error/length(x);

        Eb_No=snr(oye)-3; EbNo=10^(Eb_No/10); % usado para la capacidad
        cap(1,oye)=log2(det(IM + (EbNo/Ntx)*H*H')); % MIMO channel capacity
end

```



```

%xxxxxxxxxxxxx pasamos al receptor xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

    dc= decodmimo(H,senalr,t); %Decodifican los datos
    w=0;

    for k=1:t;
        rx=dc(:,k);
        w=w+1;
        senalrecibida(1,w)=rx(1,1);
        w=w+1;
        senalrecibida(1,w)=rx(1,2);
    end

%xxxxxxxxxxxxx Decodificador Convolutacional xxxxxxxxxxxxxxxxxxxxxxxx

    %dxcod=decoder(senalrecibida); % Se activa o no dependiendo del
    % sistema en estudio con codificación
    % de canal o sin codificación

%xxxxxxxxxxxxx Calculo del Número de errores y BER xxxxxxxxxxxxxxxx

    bit_error(1,g) = sum(data~=senalrecibida');
    BER =bit_error/length(data);
    Eb_No=snr(g)-3; EbNo=10^(Eb_No/10);
    cap(1,g)=log2(det(IM + (EbNo/Ntx)*H*H')); % capacidad del canal MIMO
end

```

BIBLIOGRAFÍA

- [1] BÖLCSKEI, H. AROGYASWAMI, P. Multiple-Input Multiple-Output (MIMO) Wireless Systems.
- [2] Hoo-Jin Lee, Shailesh Patil, Raghu G. Raj, Fundamental overview and simulation of MIMO systems for Space-Time coding and Spatial Multiplexing. Artículo PDF. Disponible en: http://www.ece.utexas.edu/~wireless/EE381K11_Spring03/projects/4.3.pdf
- [3] GONZALEZ LOPEZ, M. Técnicas Avanzadas de Procesado de Señal para Sistemas Multiple Input Multiple Output (MIMO). Tesis Doctoral, Departamento de Electrónica y Sistemas, Universidad de A Coruña. Febrero de 2004. Disponible en: <http://www.des.udc.es/~miguelgl/docs/tesis.pdf>
- [4] LO, CALEB. Diversity Combining for Mobile Communications. Documento PDF. Disponible en: http://users.ece.utexas.edu/~clo/ee80c_thesis.pdf
- [5] Receiver Techniques for Fading Dispersive channels. Disponible en: [http://www.harathi.net/EE%20499_062_Ch_4.pdf\(2\)](http://www.harathi.net/EE%20499_062_Ch_4.pdf(2))
- [6] FITTON, MIKE. Mobile and Personal Communications. Telecommunications Research Lab, Toshiba Research Europe Limited. Documento PDF. Disponible en: http://users.ictp.it/~radionet/2001_school/lectures/fitton/Mobile_communications.pdf
- [7] GEREIN, NEIL. A Spatial Diversity Scheme For Fixed Point Indoor Wireless Communication. Master of Science, in the Department of Electrical Engineering, University of Saskatchewan. December 2003. Documento PDF. Disponible en: http://library2.usask.ca/theses/available/etd-12232003-182252/unrestricted/Neil_Gerein_thesis.pdf
- [8] Diversity-Combining Techniques. Disponible en: <http://www.vocw.edu.vn/content/m10257/latest/>
- [9] KÜHN VOLKER. Wireless Communications over MIMO Channels: Applications to CDMA and Multiple Antenna Systems. Universidad de Rostock Alemania. 1 ed. Inglaterra. 2006.