L = 100; % Número de símbolos en la simulación

M = 8; % Constelacion PAM de M simbolos

k = log2(M); % Número de bits por símbolo

Eb = 7; % Valor de Eb en unidades naturales (calculado de forma teórica en el Hito3.1.)

Eb\_dB = 10\*log10(Eb); % Valor de Eb en dB

Eb\_No\_dB = 0:1:17; % Valores de Eb/No en dB

for count = 1:length(Eb\_No\_dB)

indices\_simbolos\_Tx = randint(L, 1, M); % Generamos L símbolos aleatorios (con valores de 1 a 8)

simbolos\_Tx = pammod(indices\_simbolos\_Tx, M); % Generamos los L símbolos aleatorios (valores [-7, -5, -3, -1, 1, 3, 5, 7])

No\_dB = Eb\_dB - Eb\_No\_dB(count); % Valor de No en dB

No = 10^(No\_dB/10); % Valor de No en unidades naturales

ruido = sqrt(No/2)\* randn(L, 1); % Generamos L muestras de ruido gaussiano de varianza No/2

simbolos\_Rx = simbolos\_Tx + ruido; % Sumamos el ruido aditivo

decision = zeros(L, 1);

for n=1:L % Generamos el decisor. Manera poco bonita, pero eficaz

if simbolos\_Rx(n) < -6

decision(n) = -7;

elseif simbolos\_Rx(n) < -4

decision(n) = -5;

elseif simbolos\_Rx(n) < -2

decision(n) = -3;

elseif simbolos\_Rx(n) < 0

decision(n) = -1;

elseif simbolos\_Rx(n) < 2

decision(n) = 1;

elseif simbolos\_Rx(n) < 4

decision(n) = 3;

elseif simbolos\_Rx(n) < 6

decision(n) = 5;

elseif simbolos\_Rx(n) > 6

decision(n) = 7;

end

end

SER\_experimental(count) = sum((decision ~= simbolos\_Tx))/L; % Calculamos la SER experimental

SER\_teorica(count) = ((2\*(M-1))/M)\*qfunc(sqrt((6\*k\*Eb)/((M.^2-1)\*No))); % Calculamos la SER teórica

end

% Representamos los resultados.

figure;

clf;

semilogy(Eb\_No\_dB,SER\_experimental,'b-', Eb\_No\_dB,SER\_teorica, 'r-');

grid on;

legend('Experimental','Teorica');

xlabel('Eb/No (dB)');

ylabel('SER experimental');

%Cuantos más simbolos haya más se parecerá la SER experimental a la teórica.