

**Instituto Politécnico Nacional**

*Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas*

Teoría de la información

**Practica 3**

Codificación de canal

**Profesor**

Jorge Rojas Beltrán

**Grupo**

2TM4

**Alumno**

Alvarado Balbuena Jorge Anselmo

2019/04/09

**Ejercicio 1**

**Código**

k=4; % (7,4)

for i=1:2^k

for j=k:-1:1

if rem(i-1,2^(-j+k+1))>=2^(-j+k)

u(i,j)=1;

else

u(i,j)=0;

end

end

end

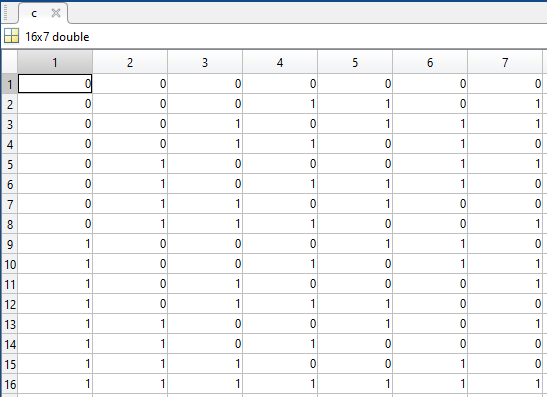
% [I P]

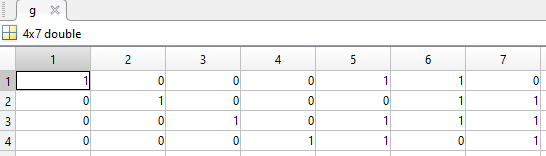
g=[1 0 0 0 1 1 0; 0 1 0 0 0 1 1; 0 0 1 0 1 1 1; 0 0 0 1 1 0 1];

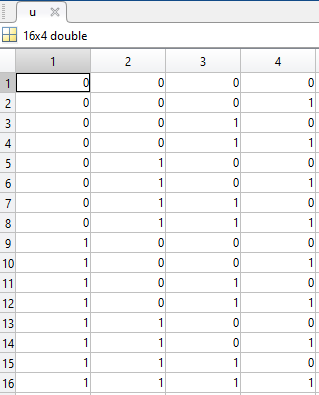
c=rem(u\*g,2); % Funcion de residuo de division

d\_min=min(sum((c(2:2^k,:))')); % Distancia minima de Hamming

**Resultados**







**Ejercicio 2**

**Código**

I = eye(11);

paridad=[1 0 0 1;

1 1 1 1;

0 0 1 1;

1 0 1 1;

0 1 0 1;

1 1 0 0;

0 1 1 0;

1 1 1 0;

0 1 1 1;

1 1 0 1;

1 0 1 0];

g=[I paridad];

k=11;

for i=1:2^k

for j=k:-1:1

if rem(i-1,2^(-j+k+1))>=2^(-j+k)

u(i,j)=1;

else

u(i,j)=0;

end

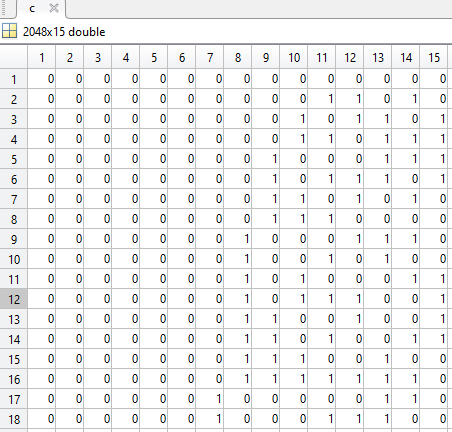
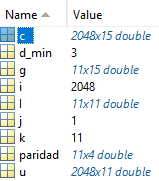
end

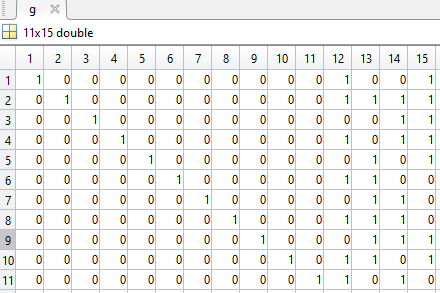
end

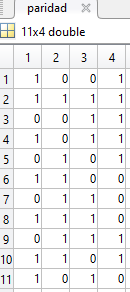
c=rem(u\*g,2);

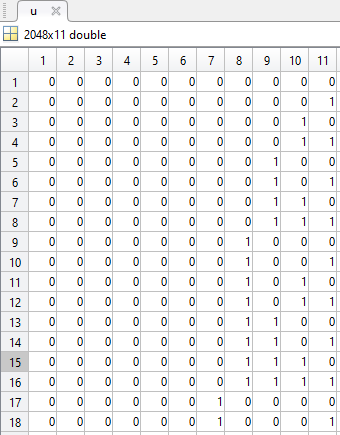
d\_min=min(sum((c(2:2^k,:))'));

**Variables y resultados**









**Ejercicio 3**

**Código**

% Matriz generadora ejercicio 1

G=[1 0 0 0 1 1 0;

0 1 0 0 0 1 1;

0 0 1 0 1 1 1;

0 0 0 1 1 0 1];

msg=[1 1 0 1]; % Fila 14 de u del ejercicio 1

% Funcion de codificacion

% Parametros: msg,n,k,metodod de codificacion, matriz generadora

code=encode(msg,7,4,'linear',G); % Fila 14 de c

% Funcion de decodificacion

dec=decode(code,7,4,'linear',G); % Msg inicial

I = eye(11); % Identidad

paridad=[1 0 0 1; % Paridad

1 1 1 1;

0 0 1 1;

1 0 1 1;

0 1 0 1;

1 1 0 0;

0 1 1 0;

1 1 1 0;

0 1 1 1;

1 1 0 1;

1 0 1 0];

g=[I paridad];

msg1=[0,0,0,0,0,0,0,1,0,0,0]; % Fila 9 u

% Fila 9 de c del ejercicio 2

code1=encode(msg1,15,11,'linear',g);

dec1=decode(code1, 15, 11, 'linear', g);

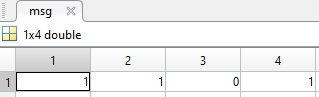
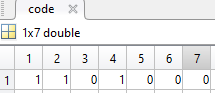
msg2=[0,0,0,0,0,0,0,1,0,0,1]; % Fila 10 u

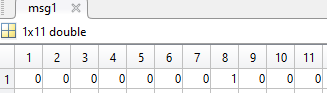
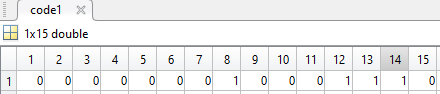
% Fila 10 de c del ejercicio 2

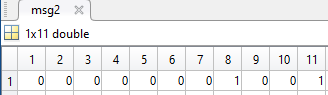
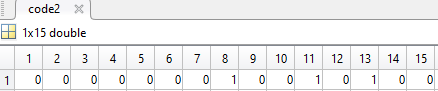
code2=encode(msg2,15,11,'linear',g);

dec2=decode(code2, 15, 11, 'linear', g);

**Resultados**

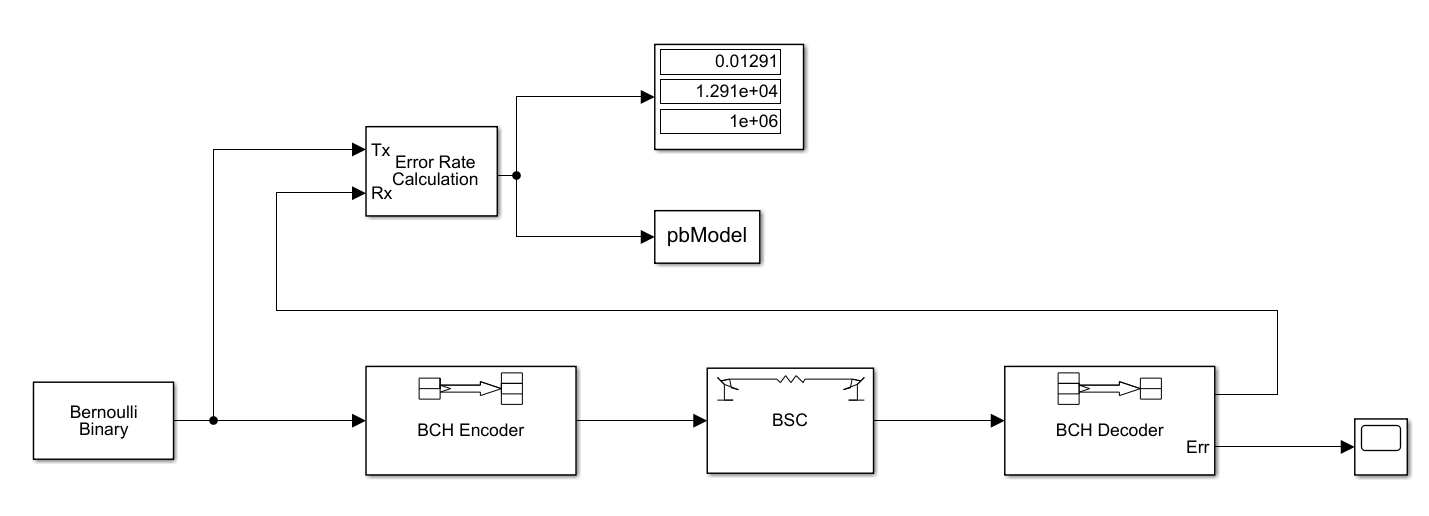






**Ejercicio 4**

**Modelo simulink**



**Parámetros de configuración**



**Código**

p=0.07864:-0.0015:0.0059-0.0015;

pb157=zeros(1,length(p));

pb74=zeros(1,length(p));

pb1511=zeros(1,length(p));

j=1;

for i=p

% BCH(15,7)

pbsc157=i;

simOut157 = sim('codec157.slx');

pb157(j)=pbModel157(1);

% BCH(7,4)

pbsc74=i;

simOut74 = sim('codec74.slx');

pb74(j)=pbModel74(1);

% BCH(15,11)

pbsc1511=i;

simOut1511 = sim('codec1511.slx');

pb1511(j)=pbModel1511(1);

j=j+1;

end

figure(4);

plot(p,pb157,'-.',p,pb74,'--',p,pb1511,'LineWidth',1.5);

title('$BCH(15,11), \ BCH(7,4), \ BCH(15,11)$','Interpreter','latex');

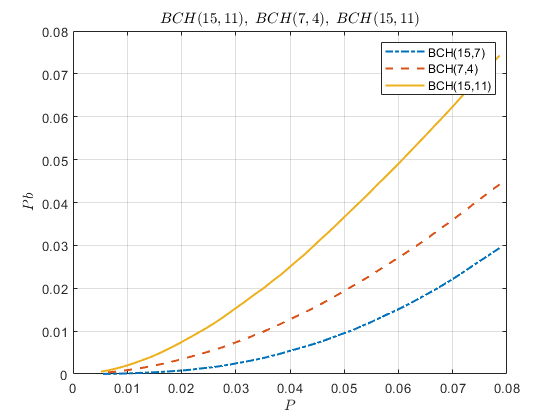
xlabel('$P$','Interpreter','latex');

ylabel('$Pb$','Interpreter','latex');

legend('BCH(15,7)','BCH(7,4)','BCH(15,11)');

grid on;

**Resultados**

****

**Conclusiones**