Facultad de Ingeniería

Carrera de Ingeniería Electrónica\

CURSO

CODIFICACION DE FUENTE Y CANAL

PROYECTO

DISEÑO DE UN RADIOENLACE WI-FI CON CODIFICADOR DE BLOQUE BHC PARA TRASMITIR HERRAMIENTAS DE DIAGNOSTICO DE RAYOS X

INTEGRANTES Y CÓDIGO

Pazos Gaspar, Alessandro Francisco u201726358

PROFESOR

Dr. Valdéz Velásquez López, Carlos Rafael

SECCIÓN

EL71

FECHA

7 de octubre 2023

1. **Explicación teórica**
   En el departamento de Huánuco en la provincia de Yanas, se presentan problemas para la conectividad Wifi para la transmisión de diagnóstico médico de rayos X. Estos datos contienen mucha información la cual se desea proteger ya que en la provincia no se cuenta con especialistas para analizar estos datos, con lo cual se envían por medio de un radioenlace Wi-Fi hacia la ciudad de Huánuco ya que es la ciudad más cercana con varios especialistas capacitados para estos diagnósticos, así como también una mejor conectividad.

2. **Diseño de la solución**
   2.1. Codificador BHC (31,11), t = 2
   Nuestro primer codificador tiene un mensaje que consta de 11 bits y se agregaron 20 bits de redundancia, este sistema detecta y corrige errores simples y dobles, en base a estos valores calculamos la probabilidad de error de bits decodificados.

$$n = 31\,, k = 11$$
$$R_c = \frac{k}{n} \approx 31.43\%$$

Sabes que la probabilidad de error de bit codificador tiene la siguiente expresión

$$P_b \approx \binom{n-1}{t} \alpha^t$$

En donde α es la probabilidad de error por bit a la salida del demodulador y lo calculamos con las siguientes ecuaciones

$$\alpha = Q\left(\sqrt{2\gamma_c}\right), \quad \gamma_c = R_c\gamma_b, \quad R_c = \frac{k}{n}$$

Para un Pb = 10^{-6} se calcula γ_b

$$\sqrt[t]{\frac{10^{-6}}{\binom{n-1}{t}}} = \alpha$$

$$\alpha = \sqrt[2]{\frac{10^{-6}}{\binom{31-1}{2}}} \approx 4.7946 \times 10^{-5}$$

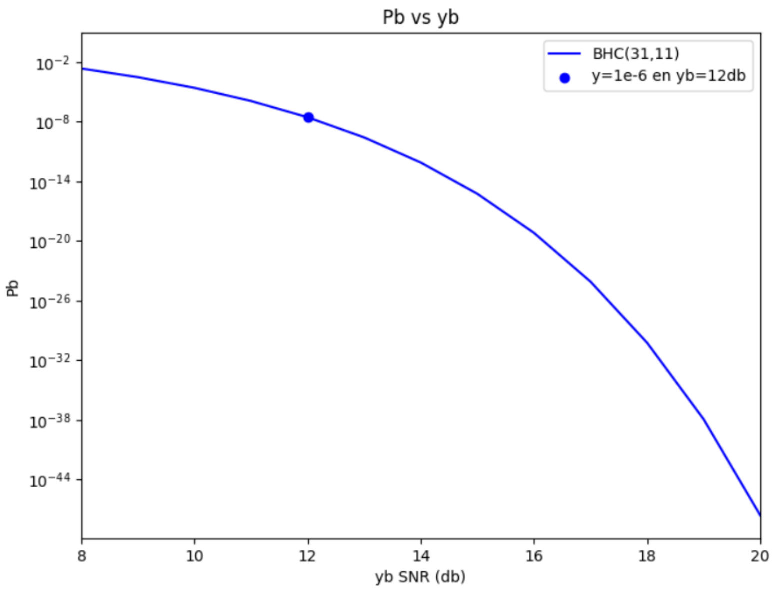
$$\alpha = Q\left(\sqrt{2R_c\gamma_b}\right) = erfc\left(\frac{\sqrt{2R_c\gamma_b}}{\sqrt{2}}\right) \times \frac{1}{2}$$

$$\alpha = \ erfc\left(\sqrt{R_c\gamma_b}\right) \times \frac{1}{2}$$

Con esta expresión se procede a calcular la inversa de la función para poder asi hallar el valor de γ_b

$$erfc^{-1}(2\alpha) = \ \sqrt{R_c\gamma_b}$$
$$\gamma_b = \frac{erfc^{-1}(2\alpha)^2}{R_c} \approx 13.31\ db$$

Con estos valores procedemos a graficar la curva teórica P_b vs y_b de forma aproximada para este primer codificador



En la siguiente gráfica observamos que nuestro modulador posee un $P_b = 10^{-6}$ cuando el sistema tiene un SNR igual a 12 db

2.2. Codificador BHC (63,39), t = 4

De la misma forma calculamos para este codificador, el cual presenta una mayor distancia Hamming lo cual nuestro sistema puede detectar y corregir desde errores simples a errores cuádruples.

$$n = 63\,, k = 39$$

Calculamos la eficiencia del codificador

$$R_c = \frac{k}{n} \approx 62\%$$

Para un Pb = 10^{-6} se calcula γ_b

$$\sqrt[t]{\frac{10^{-6}}{\binom{n-1}{t}}} = \alpha$$

$$\alpha = \sqrt[4]{\frac{10^{-6}}{\binom{63-1}{4}}} \approx 1.1571 \times 10^{-3}$$

$$\alpha = Q\left(\sqrt{2R_c\gamma_b}\right) = erfc\left(\frac{\sqrt{2R_c\gamma_b}}{\sqrt{2}}\right) \times \frac{1}{2}$$

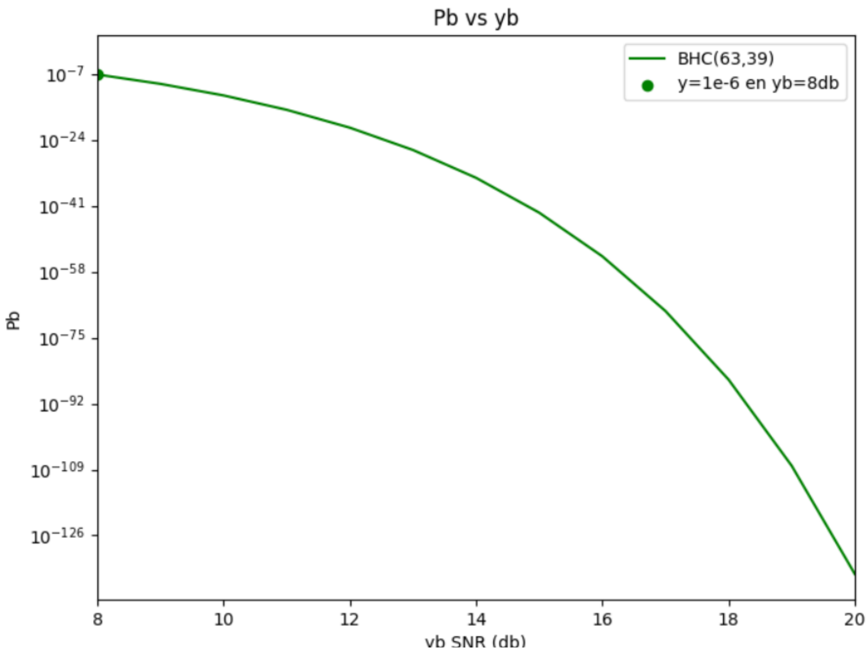
$$\alpha = \ erfc\left(\sqrt{R_c\gamma_b}\right) \times \frac{1}{2}$$

Con esta expresión se procede a calcular la inversa de la función para poder asi hallar el valor de γ_b

$$erfc^{-1}(2\alpha) = \sqrt{R_c\gamma_b}$$

$$\gamma_b = \frac{erfc^{-1}(2\alpha)^2}{R_c} \approx 8.75\ db$$

Procedemos a realizar la gráfica aproximada del codificador en donde yb varia de 8 a 20 db.



2.3. Comparación de los 2 sistemas y un sistema sin codificar

De la misma forma se calcula γ_b para un Pb = 10^{-6} del sistema sin codificar

$$P_b = \alpha = 10^{-6}$$

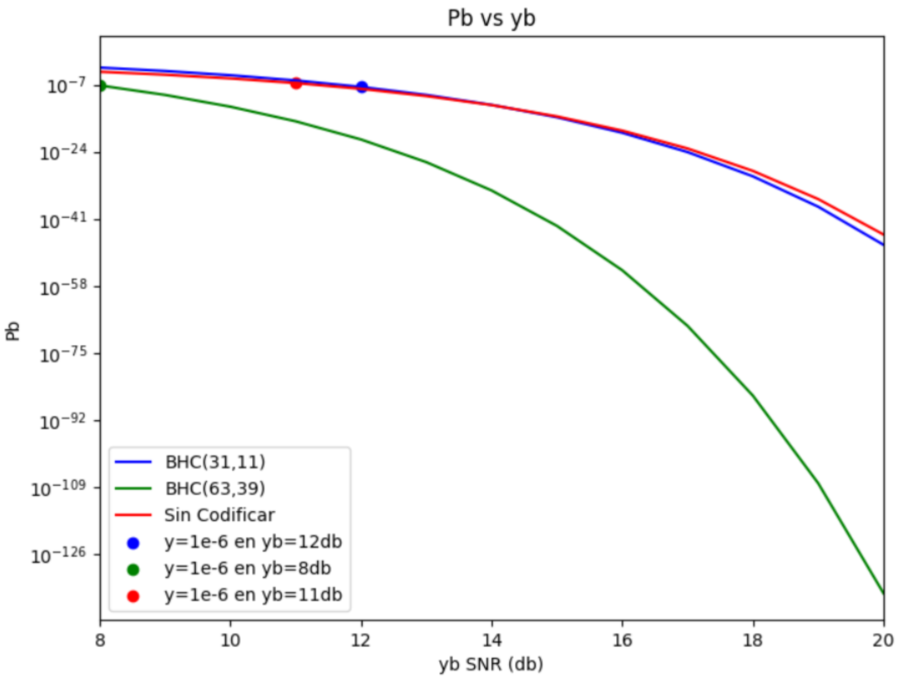
$$\alpha = Q\left(\sqrt{2\gamma_{ub}}\right) = erfc\left(\frac{\sqrt{2\gamma_{ub}}}{\sqrt{2}}\right) \times \frac{1}{2}$$

$$\alpha = erfc\left(\sqrt{\gamma_{ub}}\right) \times \frac{1}{2}$$

Con esta expresión se procede a calcular la inversa de la función para poder asi hallar el valor de γ_b

$$erfc^{-1}(2\alpha) = \sqrt{\gamma_{ub}}$$

$$\gamma_{ub} = erfc^{-1}(2\alpha)^2 \approx 10.53\ db$$



Al observar la grafica observamos que el sistema BHC (31,11) no es tan eficiente, ya que tiene la misma gráfica que el sistema sin codificar, también podemos obtener la ganancia de cada codificador.

- Ganancia del primer codificador BHC (31,11) por simulación

$$A_{BHC(31,11)} = \gamma_{\sin codificar} - \gamma_{BHC(31,11)}$$
$$A_{BHC(31,11)} = 11\ db - 12\ db = -1\ db$$

- Ganancia del segundo codificador BHC (63,39)

$$A_{BHC(63,39)} = \gamma_{\sin codificar} - \gamma_{BHC(63,39)}$$
$$A_{BHC(63,39)} = 11\ db - 8\ db = 3\ db$$

3. Conclusiones

- En base a la gráfica anterior y a los cálculos previos se logra demostrar que el sistema de codificación con una mayor distancia de Hamming presenta una menor probabilidad de error de bit a la salida del decodificador.
- Se demostró que un codificador BHC con una menor distancia de Hamming llega ser ineficiente, al punto que su gráfica Pb vs γ_b está por encima de un sistema sin codificar.

4. Anexo código en Python

```python
from scipy.special import erfc
from scipy.special import comb
import matplotlib.pyplot as plt
import numpy as np


# Codificador BHC(31,11) t=2
n1 = 31
k1 = 11
t1 = 2
Rc1 = k1/n1
yb_db = np.arange(6,21,1)
yb = 10**(yb_db/10)
yc = Rc1*yb
alpha1 = 1/2*erfc((2*yc)**0.5/2**0.5)
Pb = comb(n1-1,t1)*(alpha1**(t1+1))
indice_yb_db1 = np.where(Pb <= 1e-6)[0][0]

# Codificador BHC(63,39) t=4
n2 = 63
k2 = 39
t2 = 4
Rc2 = k2/n2
yc2 = Rc2*yb
alpha2 = 1/2*erfc((2*yc2)**0.5/2**0.5)
Pb2 = comb(n2-1,t2)*(alpha2**(t2+1))
indice_yb_db2 = np.where(Pb2 <= 1e-6)[0][0]
```

```python
indice_yb_db2 = np.where(Pb2 <= 1e-6)[0][0]

# Sin Codificador
Pub = erfc(yb**0.5)/2
indice_yb_db = np.where(Pub <= 1e-6)[0][0]
plt.figure(figsize=(8, 6))
plt.semilogy(yb_db, Pb, label='BHC(31,11)', color='blue')
plt.scatter(yb_db[indice_yb_db1], Pb[indice_yb_db1], color='blue',
             label=f'y=1e-6 en yb={yb_db[indice_yb_db1]}db')  # Marca e

plt.semilogy(yb_db, Pb2,label='BHC(63,39)', color='green')
plt.scatter(yb_db[indice_yb_db2], Pb2[indice_yb_db2], color='green',
             label=f'y=1e-6 en yb={yb_db[indice_yb_db2]}db')

plt.semilogy(yb_db, Pub,label='Sin Codificar', color='red')
plt.scatter(yb_db[indice_yb_db], Pub[indice_yb_db], color='red',
            label=f'y=1e-6 en yb={yb_db[indice_yb_db]}db')
#plt.xscale("log")
plt.xlim(8,20)
plt.xlabel('yb SNR (db)')
plt.ylabel('Pb')
plt.title('Pb vs yb')
plt.legend()
plt.show()
```