

Activitat pràctica 2 (AC2)

Comunicacions Digitals

Curs 2024-2025

Joan Claudi Socoró, Ferran Castro

Índex

1.	Introducció	3
2.	Funcions de la llibreria <i>Communications Toolbox</i> a utilitzar	3
3.	Codis a simular.....	4
4.	Requisits	4
5.	Document a lliurar	6
6.	Data màxima de lliurament	6

1. Introducció

En aquest segon exercici simularem codis lineals de blocs i compararem la seva probabilitat d'error. Concretament, simularem...

- Un codi lineal general
- Un codi de Hamming

L'objectiu és comparar la probabilitat d'error dels dos tipus de codificadors diferents (un codificador de cada família). Visualitzarem les dos corbes de BER en una sola gràfica i donarem conclusions finals, fent exercici de reflexió i comentant els resultats.

Per fer la comparativa simularem la transmissió de $\geq 4 \times 10^6$ bits en un canal BSC (*Binary Symmetric Channel*) amb una certa probabilitat de transició p , i obtindrem la probabilitat d'error després de decodificar cadascun dels dos codis. Utilitzarem 9 valors de p , donats per l'expressió següent;

$$p(i) = 10^{\left(\frac{i-16}{8}\right)} ; i = 0, 1 \dots 8$$

Aquests valors de p es troben entre 0.01 i 0.1.

Per fer la simulació utilitzarem les llibreries del Communications Toolbox de Matlab per simular els processos de codificació, transmissió pel canal i decodificació, amb una filosofia *hard-decision decoding*.

Donat que es tracta d'una simulació de Montecarlo, aquest cop caldrà que programeu una funció de tipus script de Matlab (sense capçalera *function* a l'inici) que implementi les simulacions dels dos codis, tal i com se us indica més endavant en aquest document.

Alguns aspectes de notació que convé recordar, i que usarem al llarg d'aquest document, són:

- k : nº de bits d'entrada del codificador (paraula missatge d'entrada)
- n : nº de bits de sortida del codificador (paraula codificada de sortida)

2. Funcions de la llibreria *Communications Toolbox* a utilitzar

A continuació us donem algunes indicacions sobre funcions que podeu usar per tal de realitzar les vostres simulacions:

- **randi.m**: per generar matrius de missatges o de dades a transmetre
- **gen2par.m**: per poder calcular la matriu de verificació de la paritat (**H**) a partir de la matriu generadora del codi (**G**) i viceversa, en els casos que això sigui necessari
- **syndtable.m**: per calcular la taula de patrons d'error associats als $2^{(n-k)}$ síndromes del codi, ordenats per ordre decimal de la paraula síndrome corresponent. Aquesta funció us permet estudiar el tipus de codi que esteu analitzant

- **gfweight.m**: per calcular la distància mínima d'un codi lineal de bloc
- **encode.m**: per codificar un conjunt de dades. Permet especificar el tipus de codi: lineal general, cíclic o Hamming. Nosaltres farem servir lineal general i Hamming.
- **decode.m**: per descodificar un conjunt de dades. Nota: Si se li passa la taula de patrons d'error que retorna la funció syndtable.m, la funció accelera el procés de descodificació.
- **bsc.m**: per simular la transmissió d'un conjunt de dades per un canal BSC que té una certa probabilitat de transició p .
- **biterr.m**: per calcular el nombre de bits erronis entre dues seqüències de dades binàries

Podeu explorar el “help” de cada funció per solucionar la tasca encomanada.

Observeu que a les funcions **encode** y **decode** podeu passar-li matrius binàries de tantes files com paraules y tantes columnes com bits té cada paraula a codificar o descodificar. Això us permetrà ser més eficients al realitzar les vostres simulacions.

3. Codis a simular

3.1 Codi lineal general

En aquest cas se us demana que simuleu un codi lineal amb matriu generadora G ;

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

3.2 Codi de Hamming

En aquest cas se us demana que simuleu un codi de Hamming d'ordre $m = 3$. Consulteu els apunts pdf capítol 6 pàg. 58 per deduir les matrius generadora i de verificació de paritat.

4. Requisits

En aquesta activitat en grup es demana que programeu una única funció de tipus script amb nom `simula_AC2_grupX.m` (essent X el vostre número de grup), que es pugui cridar per a realitzar la simulació dels dos codis en un canal BSC, i que com a resultat generi una gràfica final del tipus de la de la figura 1. No es tracta d'aconseguir exactament la mateixa gràfica sinó de generar-ne una altra amb codis diferents dels simulats en l'exemple, i on cada codi tingui un comportament diferent.

Com veieu, la figura ha de mostrar el resultat de la simulació indicant els 9 punts per a cada codi, a on l'eix X mostra la probabilitat d'error del canal (probabilitat de transició del model BSC) i l'eix Y mostra la probabilitat d'error després de descodificar. La gràfica s'ha de fer en eixos logarítmics (vegeu funció `loglog.m`).

Els dos tipus de codis simulats han de presentar comportaments clarament diferenciats, tal i com veieu a l'exemple de la figura, essent en aquest cas el codi lineal general el de millor

comportament, mentre que el Hamming és el que té pitjors prestacions. Observem com, en la figura 1, l'ordre de comportament va acord amb la quantitat de redundància del codi, és a dir, quant més redundant és el codi, millors presentacions en capacitat de correcció (menor BER després de descodificar les dades rebudes pel canal).

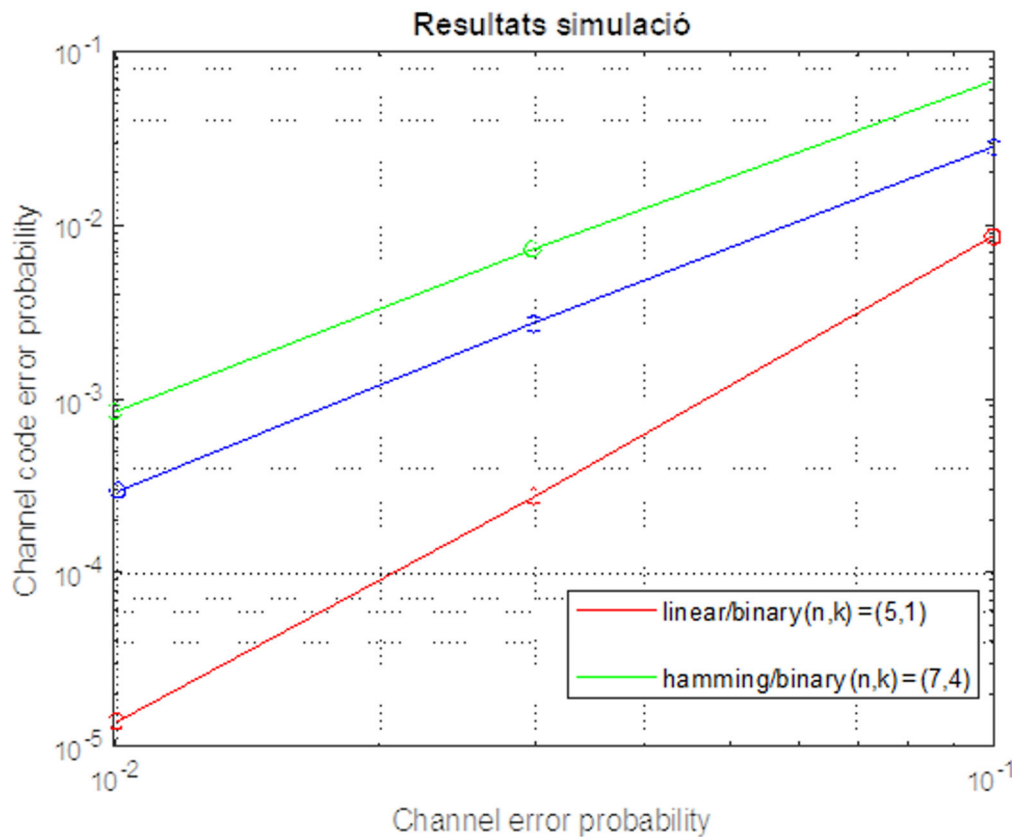


Figura 1. Exemple de resultat generat per la vostra funció. Es mostra només a títol d'exemple.

A continuació es detallen aspectes concrets a tenir en compte a l'hora de programar la vostra funció `simula_AC2_grupX.m`:

- Per a cada codi (n, k) i valor de p (probabilitat d'error del canal) es dividiran els 4×10^6 bits d'informació a transmetre en blocs de `nBitsBloc` (p. ex. valor més proper a 100.000 que sigui múltiple de la durada de bloc d'entrada; k pels codis Hamming i lineal general), i es programarà un bucle que per cada bloc acumuli els errors trobats i els sumi al nombre total d'errors. Si anomenem `nBlocs` com el número total de blocs simulats, aleshores, el número total de bits simulat serà finalment `nBitsBloc × nBlocs` $\geq 4 \times 10^6$ bits (mai menor a aquest valor). `nBlocs`, per tant, serà un número igual o bé molt proper a 40
- Per a cada tipus de codi, la funció realitzarà la simulació de Montecarlo per a cada probabilitat de transició del canal p (0.1 ... 0.01), i n'obtindrà la probabilitat d'error després de descodificar dividint el número de bits erronis obtinguts entre el número de bits simulats total ($\geq 4 \times 10^6$)

- Al simular blocs de molts bits en cada iteració del bucle, teniu en compte que per generar la informació dels missatges és convenient usar el format de matriu (crida a `randi.m`) a on cada bloc tingui `nParaules` files per `k` columnes (essent el número de bits de cada bloc simulat $nParaules \times k$). Les subsegüents funcions (`encode.m`, `bsc.m`, `decode.m`) accepten aquest format de dades (p. ex. `encode.m` retornarà una matriu de $nParaules \times n$ bits, etc.)
- Per poder estudiar bé el tipus de codi que esteu simulant, us recomanem que useu la crida a la funció `encode.m` amb l'argument 'linear/binary' en els casos Hamming i lineal general. Per tant, per aquests dos casos és recomanable que abans hagueu generat la matriu generadora del codi (funcions `hammgen` pels Hamming), el que us permetrà estudiar també les capacitats de correcció dels codis (funcions `gfweight.m` i `syndtable.m`)
- Pels codis Hamming i lineal general, a l'hora de descodificar amb la funció `decode.m` escolliu l'opció de passar-li a la funció la taula de patrons d'error generada amb la crida a la funció `syndtable.m`. D'aquesta forma, no caldrà que la funció generi aquesta taula cada cop que es faci una iteració del bucle per un cert bloc de dades a simular, i així la vostra simulació anirà més ràpida

5. Document a lliurar

La documentació a lliurar d'aquesta activitat serà un fitxer anomenat *Entregable_AC2_grupX.zip* (essent X el vostre número de grup) i que contindrà els següents ítems;

- Funció de tipus script de Matlab amb nom `simula_AC2_grupX.m`, que al executar-la faci totes les simulacions i dibuixi finalment la gràfica de resultats (exemple figura 1)
- Document `Discussio_AC2_grupX.pdf`, que inclogui 4 pàgines:
 - Portada amb el nom dels integrants del grup que han realitzat el treball
 - Càlcul teòric, per cada un dels dos codis simulats, de les matrius de generació i comprovació de paritat, distància mínima del codi i capacitats de detecció i correcció i taula de síndromes
 - Gràfica de resultats. Indicar en la llegenda de la gràfica el tipus de codi així com les dimensions (n, k) de cada codi.
 - Discussió dels resultats, a on es faci un raonament dels resultats obtinguts, fent el màxim de referència als càlculs teòrics abans esmenats

6. Data màxima de lliurament

El lliurament d'aquesta activitat es realitzarà amb data límit **diumenge dia 1 de Juny a les 23:55h**. Qualsevol pràctica lliurada més tard d'aquesta data podrà optar com a màxim a una nota de 7.