

Activitat pràctica 3 (AC3)

Comunicacions Digitals

Curs 2022-2023

Joan Claudi Socoró

Índex

1. Introducció	3
2. Funcions per la modulació i el canal	4
3. Codis convolucional	5
4. Entrellaçat	6
5. Eina bertool	7
6. Requisits	8
7. Entregable	10

1. Introducció

Aquest tercer exercici té com a objectiu que experimenteu **esquemes de codificació de canal avançats**, i que els poseu en pràctica en **canals variants en el temps amb esvaïments (o fading) no selectius en freqüència**, alhora que verifiquem la seva capacitat de correcció respecte d'esquemes més elementals.

Tornem a usar en aquesta activitat l'entorn de simulació basat en al eina bertool que vàreu estar treballant a l'activitat AC1. L'objectiu és que genereu diversos fitxers basats en la plantilla bertooltemplate.m amb els que realitzareu diverses simulacions de Montecarlo. Partint d'una de les modulacions estudiades en l'AC1, comparareu la corba de BER obtinguda amb i sense codificació de canal en un canal variant en el temps. Quant a esquemes de codificació de canal, cal que al menys compareu:

- Ús d'un únic codi, de tipus convolucional
- Ús d'un únic codi convolucional amb entrellaçat
- Ús de dos codis concatenats en sèrie amb dos entrellaçats (un convolucional i un de bloc)

En la figura següent es mostren els diferents esquemes de transmissió a simular amb la eina bertool.m.

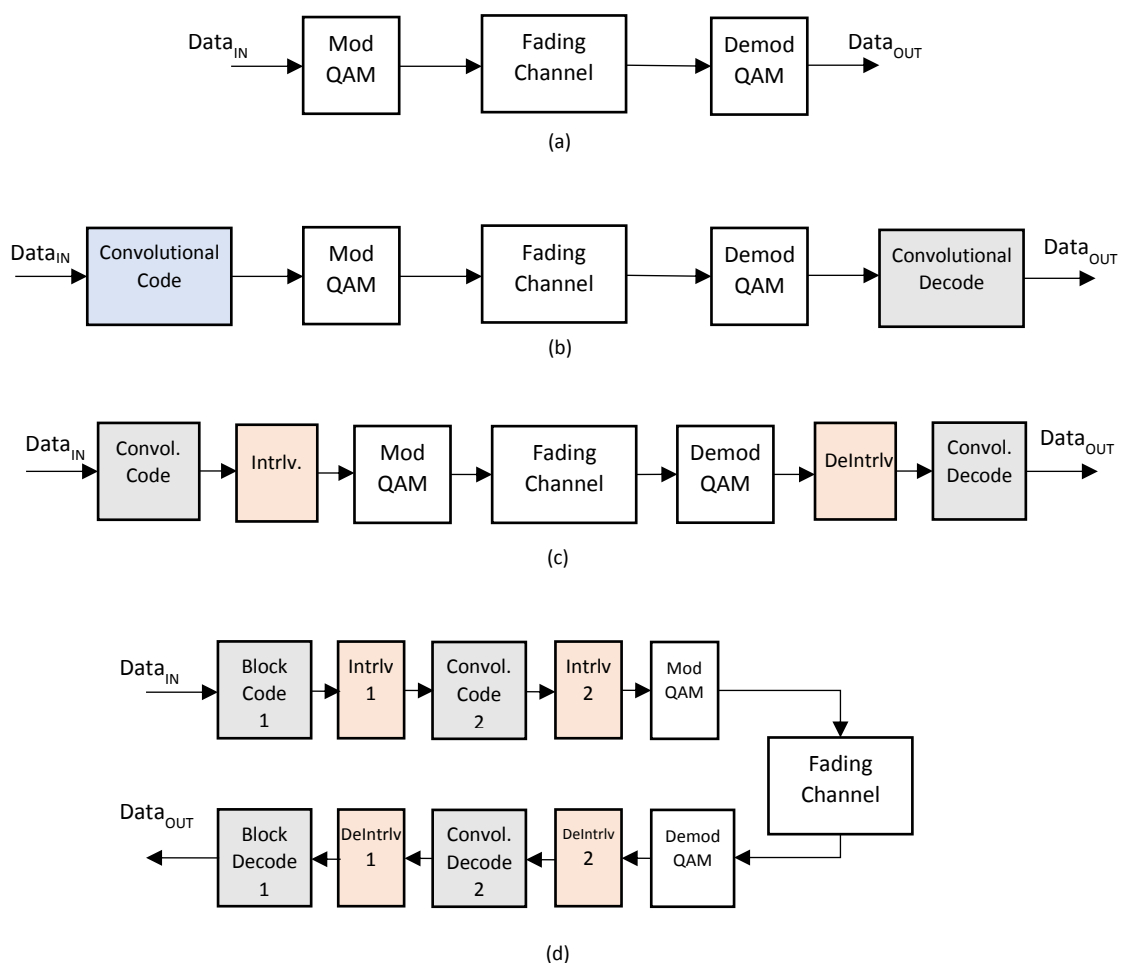


Figura 1. Esquemes de transmissió a simular amb la eina bertool.m

Us recomanem que iniciu la programació dels diferents esquemes de forma incremental, partint de la simulació que ja teníeu per la AC1 (esquema a), escollint la modulació QAM que vàreu avaluar amb les millors prestacions (corba amb menor BER), a on li afegiu primer el canal amb *fading*. Si heu fet correctament la simulació hauríeu de comprovar que enlloc d'obtenir una corba de BER amb forma exponencial (que obteníeu amb un canal AWGN) aquesta té una forma més aviat lineal (vegeu Figura 2 més endavant), comportament més típic dels canals amb esvaïments. Després seguiu amb l'esquema b), introduint el codificador i descodificador convolucional, i si heu fet bé tots els processos la corba de BER hauria de tenir un comportament una mica millor que a l'esquema a. Seguidament, afegiu el sistema d'entrellaçat després del codificador en el sistema emissor i de desentrellaçat abans del descodificador en el receptor (esquema c) i finalment afegiu dues etapes més en cada extrem del sistema, un nou codificador i descodificador així com un nou interleaver i deinterleaver (esquema d). En aquest darrer cas, es demana que useu algun dels esquemes de codificació de bloc que vàreu avaluar a l'AC2 o bé alguna variant d'aquests.

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

- k : nº de bits d'entrada del codificador
- n : nº de bits de sortida del codificador
- K : nº d'etapes de k bits del registre del codificador convolucional (paràmetre de memòria o també anomenat *ConstraintLength*).

2. Funcions per la modulació i el canal

Es tracta de que simuleu sempre una de les modulacions explorades en l'exercici AC1 (la que obtenia una corba de BER més bona – valors més baixos), amb canal no selectiu en freqüència amb *fading* i comparant les corbes de probabilitat d'error obtingudes sense codificació de canal i també usant diversos esquemes de codificació de canal.

Simulareu una transmissió a una velocitat que estigui definida entre 50 Kbits/s i 200 Kbits/s, i el canal no selectiu en freqüència amb *fading* tindrà un ample de banda Doppler (o velocitat de canvi del canal) de 50 Hz. La relació entre la velocitat de transmissió i la velocitat de canvi del canal és rellevant, ja que si aquests dos paràmetres són prou semblants o comparables aleshores estarem davant d'un canal ràpid, que és molt més difícil de tractar, assolint BER molt més altes. En canvi, amb els valors que us proposem en la simulació, estarem davant d'un canal prou lent (fixeu-vos que la velocitat del canal és força menor a la velocitat de la informació), amb el que podem arribar a compensar millor els seus efectes. A part, també us proposem que **en la simulació del canal suposeu que coneixem els coeficients d'esvaïment**, aspecte que a la pràctica implica l'ús de trames reservades per estimar el canal per així poder-lo compensar. Per tant, es tracta d'una simplificació que ens permet veure quins serien els resultats de la transmissió suposant un coneixement perfecte del canal, el que podem descriure com una cota superior de la fiabilitat en condicions ideals.

Recordeu que per realitzar aquest exercici heu de partir del codi fet en els dos exercicis previs, pel que se suposa que ja teniu els coneixements per simular un canal amb soroll blanc additiu gaussià (AWGN), així com per simular un codi lineal de bloc. Per tant, quant a codi emfatitzarem només aquelles funcions que han d'incorporar-se per poder resoldre el nou exercici, entenent que les prèvies ja les hem comentat i discutit en les activitats anteriors. Respecte el canal amb *fading*, aquest també afegeix soroll igualment que l'AWGN, pel que també haureu de calcular la potència de soroll tal i com fèieu a la AC1, i seguint la mateixa metodologia.

A continuació us donem algunes indicacions sobre funcions que podeu incorporar per tal de realitzar les vostres simulacions amb la modulació QAM que hagueu escollit de la AC1:

- **randi.m**: Funció per generar seqüències de bits d'informació. Igual que a AC2 ara heu de generar seqüències de bits (0 i 1) que codificareu, afegint bits de redundància, per després modular-les amb la modulació QAM seleccionada de la AC1. Per tant, enlloc de generar seqüències de símbols d'una modulació M-ària, ara partim de bits que es codificaran, i que després es maparan en funció de la constel·lació de punt de la vostra modulació QAM.
- **bin2dec2.m**: Funció que retorna la seqüència d'índex de símbols de la modulació definida associada a la seqüència binària d'entrada. Requereix que s'entri la matriu `constel_bits` que vàreu definir a la AC1, que indica el mapat entre símbols i de la constel·lació associada a la modulació QAM que useu.
- **[detSym_idx,detBits,nerrors] = demodqam2(rxSig, constel_symb, constel_bits, txSymb)**: Nova versió de la funció `demodqam.m` de la primera AC1 que retorna també el vector de bits detectats (variable `detBits`) pel demodulador, en funció de la constel·lació de la modulació usada (variable d'entrada `constel_bits`).
- **[rxSig,wf] = nsfading(txSig, fs, dopplerFreq, noisePower)**: Funció per a simular un canal no selectiu en freqüència, essent `txSig` el senyal transmès banda base (complex) modulats en components IQ, `fs` la freqüència de mostreig (que coincidirà amb la freqüència de símbol, o nº de símbols per segon), `dopplerFreq` és la freqüència Doppler del canal (típicament poseu 50 Hz) i `noisePower` la potència de soroll additiu blanc i Gaussià que s'afegirà al senyal amb esvaïment. En el cas que useu codificació de canal, recordeu que la freqüència de símbol (`fs`) dependrà de la velocitat de bit després d'afegir la redundància del codi de canal així com del número de bits per símbol de la modulació QAM emprada. La funció retorna tant el senyal rebut (`rxSig`, de la mateixa longitud que `txSig`) com els coeficients de *fading* del canal (`wf`). Aquest vector de coeficients de fàding és un vector de números complexos, de forma que el mòdul o valor absolut repercuteix en l'amplitud dels símbols rebuts i l'argument en la fase dels símbols rebuts. Per tal de compensar el canal en el receptor, serà necessari aplicar alguna tècnica d'equalització, que per a un canal amb esvaïments no selectius en freqüència pot ser tant senzill com una divisió punt a punt:

$$eqSig = rxSig./wf$$

Aquesta divisió intenta compensar el fet que el canal no selectiu en freqüència es pot veure com un producte del senyal transmès pels coeficients de *fading* del canal. Al dividir el senyal rebut pels coeficients del canal, aquesta compensació produirà que el soroll quedi accentuat en els períodes que hi hagi esvaïments de senyal (*fading* amb amplituds petites), el que provocarà més errors de desmodulació durant aquets períodes (ràfegues d'errors).

Recordeu d'usar l'ajuda de Matlab per a aprofundir en l'ús de totes les funcions que incorporeu al vostre codi.

3. Codis convolucional

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

- **poly2trellis.m**: per generar la estructura del diagrama de Trellis bàsic que servirà tant per a codificar com per a decodificar el codi convolucional. Li passem el paràmetre K així com la matriu de connexions dels bits de sortida del codi (*CodeGenerator*), de dimensions $k \times n$. Mireu l'exemple del codi amb velocitat $k/n = 2/3$ i $K = [5 \ 4]$ que teniu a l'ajuda de la funció, a on, com podeu veure, en el cas que $k = 2$, el registre d'entrada es desdobra en dos registres de $K - 1$ bits, de forma que cada fila de *CodeGenerator* indica les connexions en octal de cadascun dels dos registres cap a cadascuna de les n sortides. Un altre exemple podria ser el codi que hem estudiat a les transparències del tema 3, el codi amb $k/n = 1/2$ i $K = 3$. En aquest cas definiríem el codi amb la crida `trellis = poly2trellis(3,[7 5])`. A les transparències de classe disposeu d'una taula de codis convolucionals òptims amb diferents valors de k/n així com K . Useu algun dels codis que disposeu en aquesta taula.
- **commcnv_plotnextstates.m**: Per visualitzar les transicions del diagrama de Trellis del codi, a on li passem el camp `nextStates` de l'estructura que retorna la funció `poly2trellis.m`. Aquesta funció us pot servir per entendre millor el tipus de codi.
- **convenc.m**: Funció per codificar una trama de bits missatge amb un codi convolucional definit a partir del resultat de la funció `poly2trellis.m`. La trama d'entrada ha de tenir una longitud múltiple del nombre de bits per trama del missatge (k), i en les vostres simulacions heu de fer coincidir amb la longitud de bloc associada a una iteració dins del bucle que emreu per estimar un determinat valor de BER per a una determinada E_b/N_0 . En el cas que useu l'esquema de codificació concatenada (esquema d de la Figura 1) la longitud de la trama tindria en compte l'increment de bits provocat pel primer codi (codi de bloc lineal).
- **vitdec.m**: Funció per decodificar una trama de bits d'entrada amb un codi convolucional usant l'algorisme de Viterbi. Se li especifica també la estructura que retorna la funció `poly2trellis.m` així com el retard de decodificació (paràmetre `tbdepth`, típicament $5K$ per a un codi amb velocitat $1/2$). Es recomana que useu les opcions 'term' i 'hard'. En la opció 'term' el decodificador pressuposa que el codificador s'inicia i finalitza en l'estat 0, pel que alhora de codificar haureu d'afegir una cua de $(K-1)*k$ zeros a la trama d'informació d'entrada del codificador convolucional. A part, en aquest mode 'term' el decodificador no provoca cap retard de decodificació, pel que no heu de tenir en compte el valor del paràmetre `tbdepth`.
- **iscatastrophic.m**: Funció que ens permet saber si un codi pot arribar a presentar el comportament catastròfic, pel que, en cas afirmatiu, caldria descartar-lo com a bon candidat pel nostre sistema de transmissió.
- **distspec.m**: Funció que permet calcular la distància lliure (o mínima) d'un codi convolucional.

Per saber-ne més, us emplacem a que exploreu l'ajuda de cada funció amb l'objectiu de solucionar la tasca encomanada.

4. Entrellaçat

De les diverses funcions disponibles a la llibreria Communications Toolbox sobre entrellaçat us indiquem les següents, tot i que deixem la opció que mireu d'usar d'altres com per exemple, d'entrellaçat convolucional, si ho considereu oportú:

- **randperm.m**: aquesta funció permet generar una vector de permutacions aleatòries d'una certa longitud L . Per tant, us recomanem el seu ús per tal de generar el mapat

d'un sistema d'entrellaçat de bloc de tipus aleatori, definida la dimensió L del bloc a aleatoritzar.

- **intrlv.m**: per aplicar l'entrellaçat de bloc a una trama d'informació d'entrada segons el vector de mapat entrat també com argument d'entrada. Es recomana que useu el vector generat amb la funció `radnperm.m`.
- **deintrlv.m**: per aplicar el desentrellaçat de bloc a una trama d'informació segons el vector de mapat entrat també com argument d'entrada. Aquest vector de mapat ha de ser acord a l'usat en el procés d'entrellaçat (crida a la funció `intrlv.m`).

Una consideració interessant quant al sistema d'entrellaçat és quina pot ser la profunditat d'entrellaçat adequada (paràmetre L). Donada la naturalesa de les vostres simulacions, a on per tal d'estimar una certa probabilitat d'error realitzeu un bucle a on en cada iteració simuleu un cert nombre de bits (paràmetre `nBitsBloc` definit a la tasca AC1, que ara podeu tornar a definir), es recomana que useu aquest mateix paràmetre com a referència per als sistemes d'entrellaçat que simuleu. Per exemple, en la simulació amb codi convolucional i entrellaçat (esquema c de la Figura 1), una vegada el bloc de `nBitsBloc` ha estat codificat pel codi convolucional a la sortida tindrem $(n\text{BitsBloc} + (K-1)*k)*(n/k)$ bits, essent (n,k,K) els paràmetres del codi convolucional emprat.

D'altra banda, en aquest punt també és important que considereu que els resultats de les vostres simulacions poden estar fortament influenciats pel valor de L , atès que l'entrellaçat proporciona diversitat temporal en canals variants en el temps quan la seva profunditat és suficientment gran per a que dins de la seva longitud el canal tingui comportaments prou diversos (és a dir, aquest presenti comportaments bons i dolents prou vegades, o que es produeixin uns quants esvaïments al llarg de tota la trama reordenada). Per tant, quan compareu els sistemes amb i sense entrellaçat (per exemple, esquemes b i c de la Figura 1), en principi l'entrellaçat haurà estat ben dissenyat (valor de L) si en afegir-lo s'aprecia un canvi de pendent important en la corba de BER. És en aquest context que podem dir que **el sistema està proporcionant la suficient diversitat temporal per a que el descodificador en tregui un rendiment visible**.

5. Eina bertool

Ahora de fer les vostres simulacions amb la eina `bertool.m` recordeu que es tracta que useu la simulació de Montecarlo, i que especifiqueu el fitxer de simulació en el camp "Simulation MATLAB file", un dels 4 fitxers a lliurar segons s'indica a l'apartat següent de Requisits.

Una consideració important a tenir en compte és el valor del paràmetre "Number of errors" que heu de posar. Com recordareu, en l'AC1 vàrem indicar que un valor de 100 és adequat per a canals aleatoris, com és el cas d'un canal AWGN. No obstant, quan el canal presenta variacions en el temps, estem davant d'un canal no estacionari. Per aquest motiu, cal tenir present que per a poder obtenir corbes de BER prou fiables en aquest tipus de canals variants en el temps haureu d'incrementar el valor del paràmetre "Number of errors" i, en conseqüència, les vostres simulacions no seran tant ràpides com quan la simulació la fèieu en un canal AWGN. La suavitat obtinguda a les corbes de BER és un bon indicador que hem fixat adequadament aquest paràmetre, mentre que una corba de BER poc suau (amb pics i crestes) ens indica que cal augmentar el seu valor.

Sobre l'altre paràmetre "Number of bits" podeu deixar-lo a un valor de $1e8$.

6. Requisites

En aquesta activitat en grup es demana que entregueu quatre funcions compatibles amb la eina bertool.m, anomenades:

- simula_AC3_a.m (esquema a de la Figura 1, sistema sense codificació de canal)
- simula_AC3_b.m (esquema b de la Figura 1, sistema amb codificació convolucional)
- simula_AC3_c.m (esquema c de la Figura 1, sistema amb codificació convolucional i entrellaçat)
- simula_AC3_d.m (esquema d de la Figura 1, sistema amb codificació concatenada en sèrie que inclou dues etapes d'entrellaçat i dos codis, un codi de bloc i el convolucional anterior)

A títol d'exemple, i per a que disposeu d'una referència, continuació es mostren les corbes de probabilitat d'error obtingudes en 4 simulacions amb modulació 4-QAM usant els 4 esquemes de la figura 1. En aquest cas s'ha simulat una modulació a 100 Kbits/s i un canal variant amb una freqüència Doppler de 50 Hz. I la resta de paràmetres són:

- Number of errors = 100000
- Number of bits = 1e8
- nBitsBloc = 10000
- Codi convolucional (2,1,3) amb polinomis generadors [5 7] i $d_f = 5$
- Codi de bloc cíclic (15,5) amb $d_{min} = 3$
- Entrellaçats de bloc de tipus aleatori amb profunditat d'entrellaçat (L) corresponent a la durada del bloc de simulació en cada punt de l'esquema (30000 per l'entrellaçat després del codi cíclic, i de 60004 bits per l'entrellaçat just després el codi convolucional).

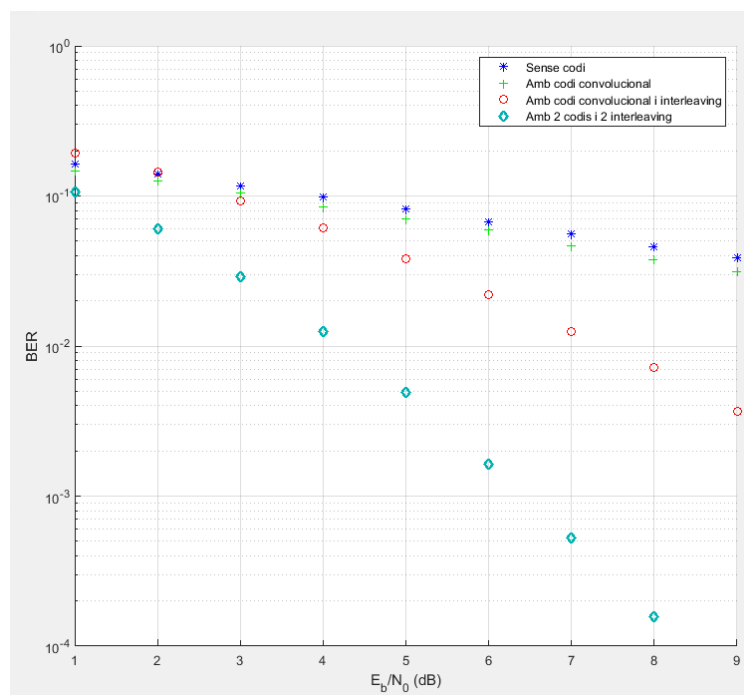


Figura 2. Exemple de resultats dels diferents esquemes de codificació en canal variant no selectiu en freqüència amb esvaïments

Com es pot observar, en un canal amb esvaïments no selectius en freqüència la corba de BER té un comportament més aviat lineal enlloc d'exponencial. Aquest tipus de fiabilitat, deguda a la

presència d'esvaïments del senyal que provoquen ràfegues d'errors en el canal, pot ésser clarament millorada amb codificació de canal. Quan passem del sistema a) al sistema b) introduint un sistema de codificació de canal hi ha un cert guany però no deixa de ser poc significatiu. En el moment que afegim l'entrellaçat, la corba comença a tenir una millora més significativa, fet propiciat per la cooperació entre la diversitat temporal que assoleix l'entrellaçat, i el propi codificador de canal, que millora el seu rendiment pel fet que la distribució dels errors en el canal passa de ser en paquets a ser més aleatòria. Finalment, en afegir una segona etapa d'entrellaçat i codificació millorem molt més encara la corba, arribant a un sistema que pot proporcionar la qualitat de servei adequada en un canal variant en el temps.

7. Entregable

L'entregable d'aquesta activitat estarà format pels següents ítems, tots encapsulats en un únic fitxer amb nom Entregable_AC3_grupX.zip (essent X el vostre número de grup):

- Funcions de tipus script de Matlab amb els noms `simula_AC3_a.m`, `imula_AC3_b.m`, `imula_AC3_c.m`, i `imula_AC3_d.m`, que al executar-les des de la eina `bertool.m` permetin realitzar les respectives simulacions i poder dibuixar finalment la gràfica de resultats (exemple Figura 2).
- Document `Discussio_AC3_grupX.pdf`, que inclogui 3 pàgines:
 - Portada amb el nom dels integrands del grup que han realitzat el treball
 - Gràfica de resultats (semblant a la Figura 2, però amb les vostres simulacions) en un marge de valors de E_bN_0 entre 1 i 9 aproximadament. Indicar en la llegenda de la gràfica el tipus de codi així com les dimensions (n,k) i/o (n,k,K) de cada codi així com les profunditats d'entrellaçat de cada entrellaçat.
 - Discussió dels resultats, a on es faci un raonament dels resultats obtinguts, fent el màxim de referència a aspectes relacionats amb la teoria que heu vist en el tema 3. Per exemple, quina és la capacitat de correcció (t) de cadascun dels dos codis que heu simulat?

El lliurament d'aquesta activitat es realitzarà amb data límit dia 26 de Maig a les 23:55h.