

Pràctica 3: Aplicació de la Codificació de Canal a la Televisió Digital

Primera part: Codificació Externa

Introducció

En aquesta pràctica veurem quins tipus de codificació de canal s'apliquen als sistemes de televisió digital, i per què.

Com ja sabem, la codificació de canal es una adaptació que realitzem sobre el missatge a enviar per tal de millorar la transmissió d'aquest i minimitzar els errors que el canal de comunicació pugui introduir.

Per tant, la codificació de canal se situa entre la codificació de font i la modulació, com podem veure a la figura 1.

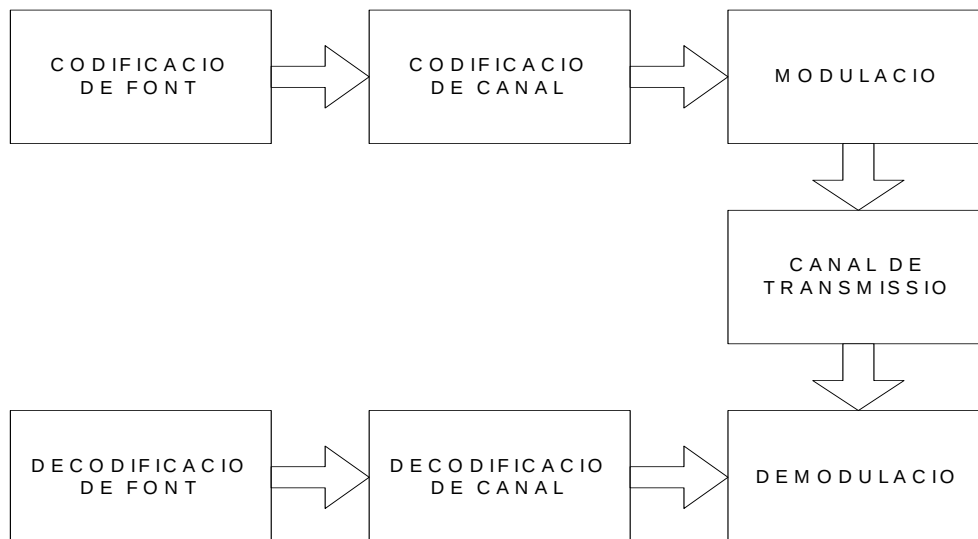


Figura 1 - Diagrama de blocs d'un sistema de comunicacions genèric

En el cas de la televisió digital, la codificació de font es realitza seguint l'estàndard MPEG, que dóna com a resultat el que es diu un *Transport Stream*, que és un seguit de bytes organitzats mitjançant packets de tamany 188. Qui estigui interessat a l'organització particular de les dades dins d'aquests paquets, se li remet al document ETR154 de l'ETSI, on s'explica com utilitzar l'estàndard MPEG per transmissions de televisió digital.

La següent part, la codificació de canal, és la que ens interessa. A la figura 2 veiem el seu diagrama de blocs.

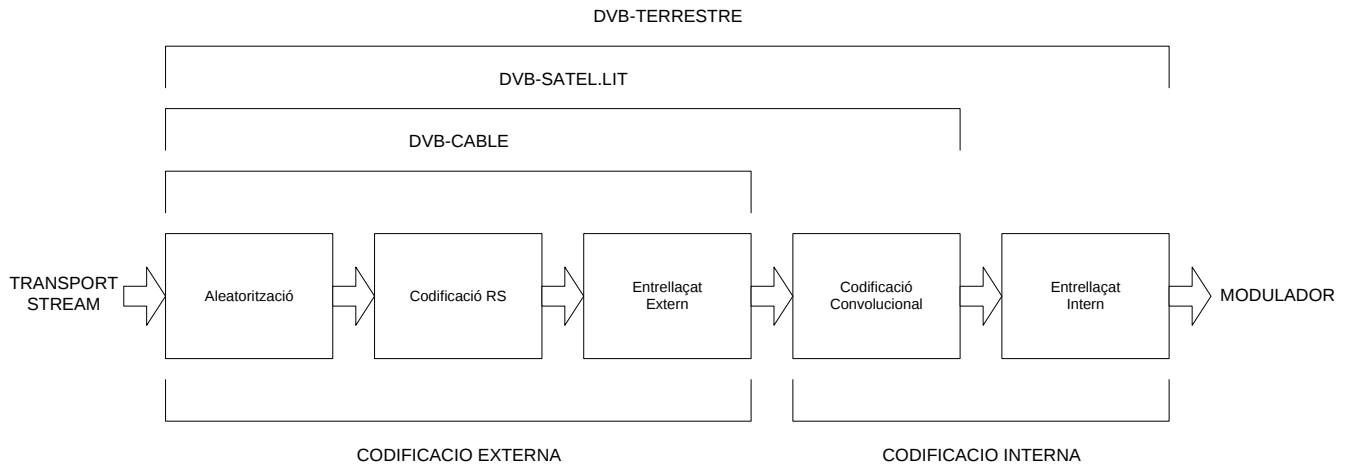


Figura 2 - Codificació de canal en un sistema de televisió digital

Com podem veure, la codificació a realitzar depèn de quin és el canal de comunicació que volem utilitzar. Això és totalment lògic, donat que la codificació de canal, com hem dit abans, no deixa de ser una adaptació del missatge a la transmissió sobre el canal. Com més problemàtic és el canal, més complexa serà la codificació, i més robusta serà la modulació. Per transmissió sobre cable, la modulació serà 64-QAM; per satèl·lit, es fa servir la modulació QPSK; i per transmissió terrestre utilitzarem una modulació d'espectre dispers anomenada COFDM.

Donat que l'implementació de un modulador COFDM és bastant complexa, i que l'entrellaçat intern és molt similar a l'extern, en aquesta pràctica ens limitarem a la transmissió per satèl·lit, que ja ens donarà una visió prou completa d'aquest tipus de sistemes.

En aquesta primera part veurem la codificació externa, o sigui l'aleatorització, la codificació de Reed-Solomon i l'entrellaçat extern, i a la segona part veurem la codificació interna, que com que ens centrarem a la transmissió per satèl·lit es reduirà a la codificació convolucional, i la seva interacció amb la codificació externa.

Es recomana molt i molt llegir el capítol 6 i l'apèndix 1 del llibre "Televisión Digital" de Hervé Benoit (Ed. Paraninfo 1998) abans de començar la pràctica.

El sistema sense codificació de canal

Anem a construir amb Simulink el sistema complet, incloent el transmissor, el canal i el receptor.

Començarem amb el sistema bàsic, amb un codificador de font i un modulador, a la banda de l'emissor, i un demodulador a la banda del receptor. Per simplificar el tema, modelarem el canal com un canal amb soroll blanc gaussià (amb transmissió satèl·lit, evidentment no n'hi ha ni multi-camí ni eco).

Arrenquem el MATLAB, i entrem al Simulink, teclejant "*simulink*" o amb el botó corresponent.

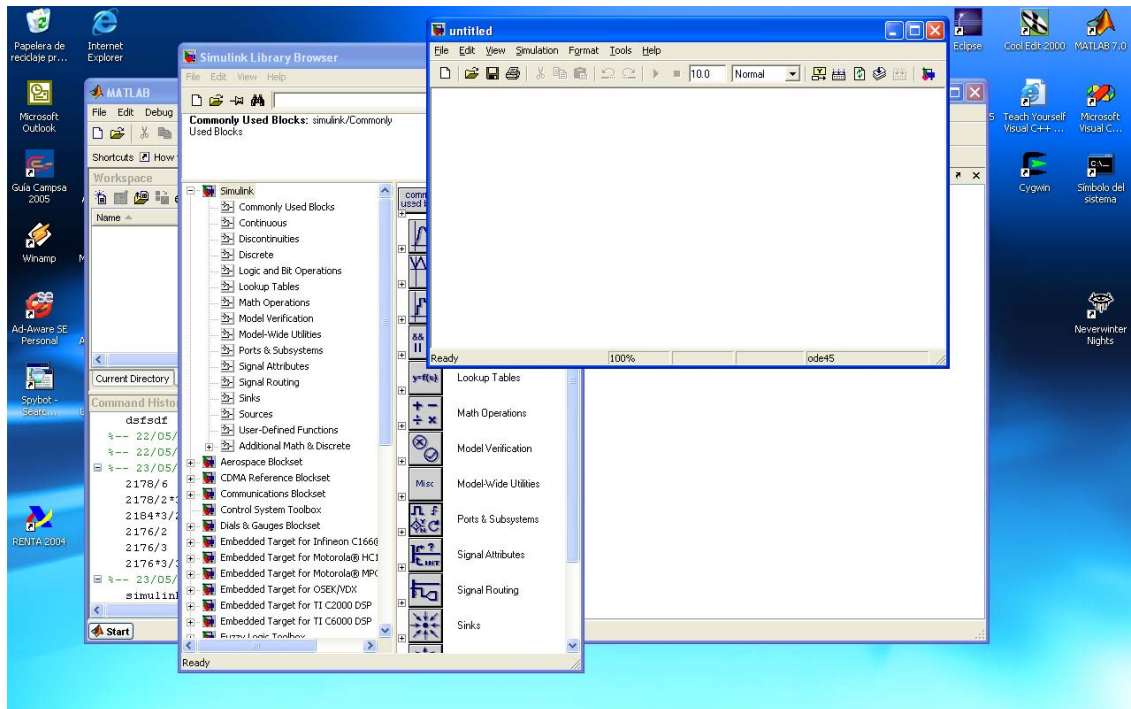


Figura 3 - Finestra d'un nou model de Simulink

Tindrem a una nova finestra el “*Simulink Library Browser*”. Li diem que volem un model nou, a “*File -> New -> Model*”, CTRL+N o el botó corresponent, i tindrem una nova finestra, buida i sense títol, com a la figura 3.

Primer de tot crearem un generador de trames, que ens donarà les trames que transmetrem. Com ja hem dit abans, les trames que ens arriben de la codificació de font són trames de 188 bytes. Per simplicitat, farem servir un generador aleatori de nombres sencers, que ens farà a la vegada de generador de font i d'aleatoritzador, que és la primera etapa de la codificació externa..

Busqueu al *Communications Blockset* el bloc *Random Integer Generator*, i poseu-lo al nostre model. Configureu-lo com a trama (*frame*) de 188 bytes, i calculeu el temps de mostreig (*Sample time*) suposant que el senyal que ens arriba té un *bitrate* de 24Mbits/s. Enrecordeu-vos en de posar un *M-ary number* adient.

A continuació, al mateix blockset, busqueu el bloc *QPSK Modulator Baseband* i poseu-lo també al nostre model. Als paràmetres del bloc, poseu *Input type* a *bit*.

Evidentment, no podem enganxar els nostres dos blocs, donat que el generador dóna trames de sencers, i el modulador vol una trama de bits. Necessitem un altre bloc al mig que ens faci la conversió. Busqueu el bloc *Integer to Bit Converter*, i fiquem-lo entre els dos. Poseu 8 bits per paraula. Enganxeu els tres blocs ara.

Anem a mirar si té coherència això. Activeu “*Format -> Port/Signal Displays -> Signal Dimensions*”. Ara hauríem de tenir el mateix que la figura 4. Per cert, ara és un bon moment per gravar el nostre model. Enrecordeu-vos ens d'anar gravant de quan en quan.

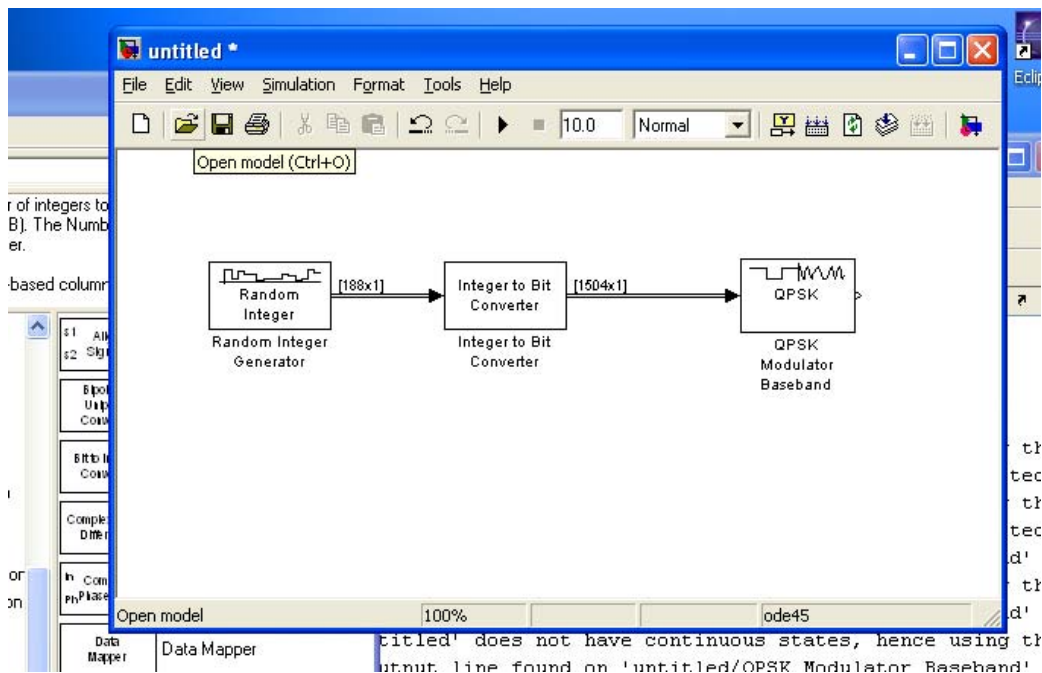


Figura 4 - Mostrem les dimensions dels senyals

Ara tenim les dimensions dels senyals que es mouen entre blocs. Veiem que entre el Generador i el Conversor va una trama de 188 bytes (doble línia vol dir trama), mentre que entre el Conversor i el Modulador va una trama de 1504 bits. Com que $1504=188 \cdot 8$, tot quadra.

(Nota: Per actualitzar les dimensions dels senyals cada cop que canviem el model, podem fer “Edit -> Update Diagram o CTRL+D”).

Molt bé. Hem fet el transmissor. Ara farem el receptor. Enganxarem els blocs de *QPSK Demodulator Baseband*, *Bit to Integer Converter* i finalment un *Terminator*. Aquest últim es troba al Blockset estàndard de Simulink, i serveix bàsicament per acabar el camí del senyal, donat que realment al final no farem res amb les dades, només volem veure el comportament del sistema. Per que quedi maco, girarem els blocs picant a sobre del bloc amb el botó dret i fent “Format -> Flip Block”. I enrecordeu-vos de configurar adientment els paràmetres dels blocs.

Ara ficarem el canal, que lligarà l'emissor amb el receptor. Abans de transmetre, hem de desfer les trames en mostres, i al receptor haurem de tornar a crear les trames. Això ho farem amb els blocs *Unbuffer* i *Buffer* respectivament. Aquests blocs es troben a la *Signal Processing Toolbox*. El bloc d'Unbuffer no necessita cap paràmetre, però el bloc Buffer necessita saber el nombre de mostres. En aquest cas tenim que havíem entrat al modulador QPSK amb una trama de 1504 bits, i el modulador genera 1 mostra cada 2 bits, treu una trama de $1504/2=752$ mostres, per tant volem que el demodulador rebi una trama del mateix nombre de mostres. O sigui que ficarem com a tamany del buffer de recepció 752 mostres.

Si ara enganxem la sortida del bloc Unbuffer amb l'entrada del bloc Buffer, i actualitzem els tamanyes dels senyals, veiem que tot quadra (si no ens hem equivocat), com a la figura 5. Efectivament veiem que no hi cap nombre entre aquests dos blocs, perquè és un camí directe, com volíem.

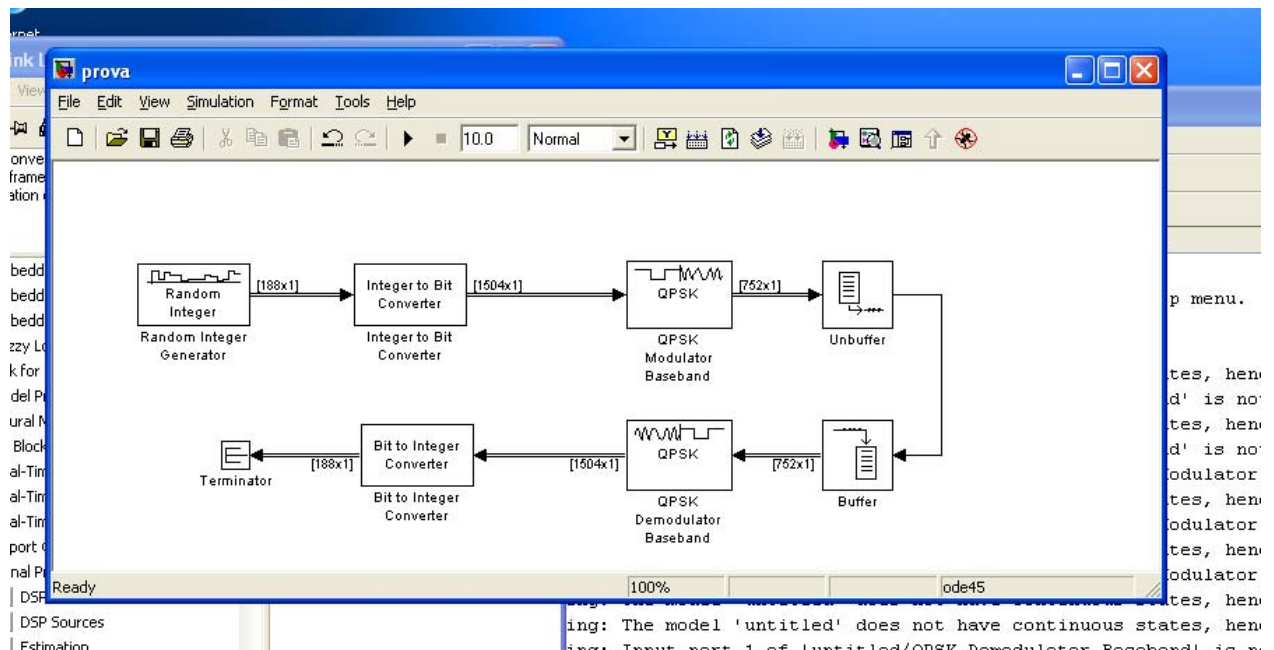


Figura 5 - Tancant el circuit

Fins i tot ara podem executar el model amb el botó de *Play*, tot i que no veurem res si ho fem.

Anem ara a ficar el model del nostre canal. Serà el bloc *AWGN Channel*, i el configurarem en mode *Signal to noise ratio (E_s/N_0)*, un E_s/N_0 de 20 i un període de símbol adient que calculareu tenint en compte que un símbol correspon a 2 bits.

I ara ha arribat el moment de poder mesurar el comportament del sistema. Ficarem el bloc *Error Rate Calculation* (a la *Communications Blockset*), i compararem per una banda la trama de bits que entra al modulator QPSK de l'emissor amb la trama de bits que surt del demodulator QPSK al receptor. Per poder veure la sortida, hem de dir primer que la tregui per un port, posant l'opció de *Output data* a *Port*. Després ficarem la nova sortida del bloc com a entrada d'un bloc *Display* (a la blockset estàndard) que ficarem al model. Redimensioneu el bloc per tal de poder veure les tres dades que dona el càlcul de BER: BER, nombre de bits erronis rebuts, i nombre total de bits rebuts. Si executeu ara el model, veureu com el display se'n va actualitzant, contínuament (figura 6).

Mirem ara el BER que ens surt. És molt alt. Ens diu que prop d'un 50% dels bits que arriben són erronis. Això no té gaire sentit, perquè una relació senyal soroll de 20 dB amb modulació QPSK és molt generosa.

Això ho podem comprovar visualment. Afegim el bloc *Discrete-Time Scatter Plot Scope* a la sortida del transmissor per mirar la constel·lació que transmetem, i a l'entrada del receptor per mirar la constel·lació que rebem. Podem veure a la figura 7 que hi ha prou marge com per pensar que el BER hauria de ser 0 o molt proper a 0.

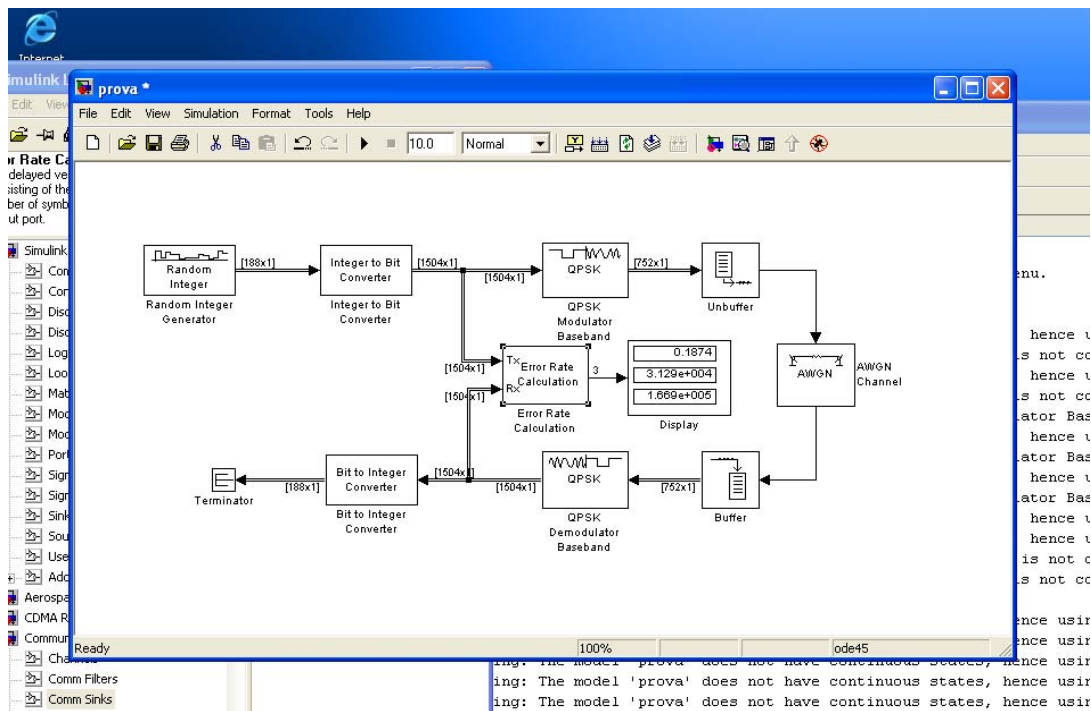


Figura 6 - Calculant el BER

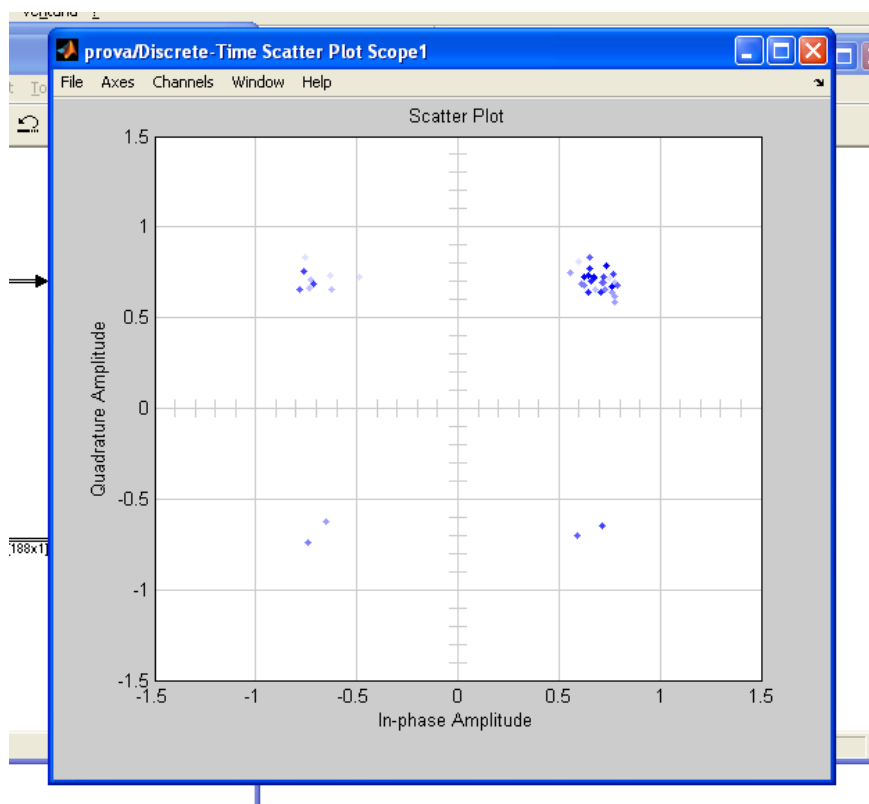


Figura 7 - Constel·lació rebuda amb SNR=20

Quin és el problema? El problema és que el mòdul de càlcul de BER no està comparant els bits que toquen. El que està passant és que el mòdul de Buffer introdueix un delay del senyal equivalent al nombre de mostres que guarda al Buffer. Lògicament, perquè ha d'esperar a que arribin totes les mostres de la trama abans d'enviar la trama sencera. Per tant, el mòdul de càlcul de BER està comparant els bits del senyal emès amb els bits del senyal rebut, però endarrerit un nombre de bits determinat. O sigui que encara que rebéssim exactament els mateixos bits, no coincidarien perquè els bits del receptor estan endarrerits.

Això es pot arreglar fàcilment ajustant el paràmetre *Receive delay* del bloc de càlcul del BER.

Exercici 1. Calculeu el retard de recepció del nostre sistema. Sabreu que és correcte perquè el BER us donarà exactament zero.

Exercici 2. Trobeu ara fins a quina SNR (aproximadament) aguanta la modulació QPSK. Considereu que el llindar és el que dona un senyal quasi lliure d'errors (quasi-error-free), això és, quan el BER és de $2e-4$. Cal remarcar que perquè el resultat sigui correcte és important que el període de símbol del canal sigui correcte.

Codificació de Reed-Solomon

El primer pas de la codificació de canal (després de l'aleatorització, que l'hem ajuntat amb la codificació de font), és la utilització del codi de Reed-Solomon RS(204,188,T=8). Tal com s'explica al capítol de la lectura recomanada, el codi RS(204,188,T=8) és una versió reduïda del codi RS(255,239,T=8). Es construeix de la següent manera:

- 1) S'afegeixen 51 zeros davant de la trama, creant una nova trama de 239 mostres
- 2) Es codifica aquesta nova trama mitjançant RS(255,239,T=8), afegint 16 nous bytes i creant una nova trama de 255 bytes
- 3) Es treuen finalment els 51 zeros que havíem afegit al començament de la trama, quedant per tant una trama de sortida de 204 mostres.

Construirem ara un mòdul que implementi aquesta codificació especial, i posarem aquest nou mòdul al nostre model.

Posem el bloc *Subsystem* (bloc estàndard) al nostre model, i canviem-li el nom a, per exemple, "Codificador RS(204,188,T=8)". Entrem al nostre nou bloc fent doble-click. Ara tenim una pantalla semblant a la que ens mostra el Simulink quan obrim un model nou. La diferència és que aquest sub-sistema ve amb una entrada i una sortida, i nosaltres el que farem serà transformar l'entrada en la sortida.

Com hem explicat abans, el nostre sub-sistema tindrà tres parts, una part que afegirà zeros, una part que farà el RS(255,239,T=8), i una darrera part que traurà els zeros que hem afegit abans.

La primera part la farem amb el bloc *Insert zero (Communications Blockset)*. Com a paràmetre li passarem un vector columna.

La segona part la farem amb el bloc *Integer-Input RS Encoder (Communications Blockset)*, i el configurem perquè sigui RS(255,239,T=8) (el T=8 és només informatiu, no cal configurar-ho).

La tercera part la farem amb el bloc *Puncture (Communications Blockset)*, que configurarem també perquè ens elimini les 51 primeres mostres, que són els zeros que hem afegit al començament.

Tindrem finalment un sub-sistema com el que veiem a la figura 8. (Si us surt un error al actualitzar els tamanys dels senyals, no us preocupeu, ara ho arreglarem).

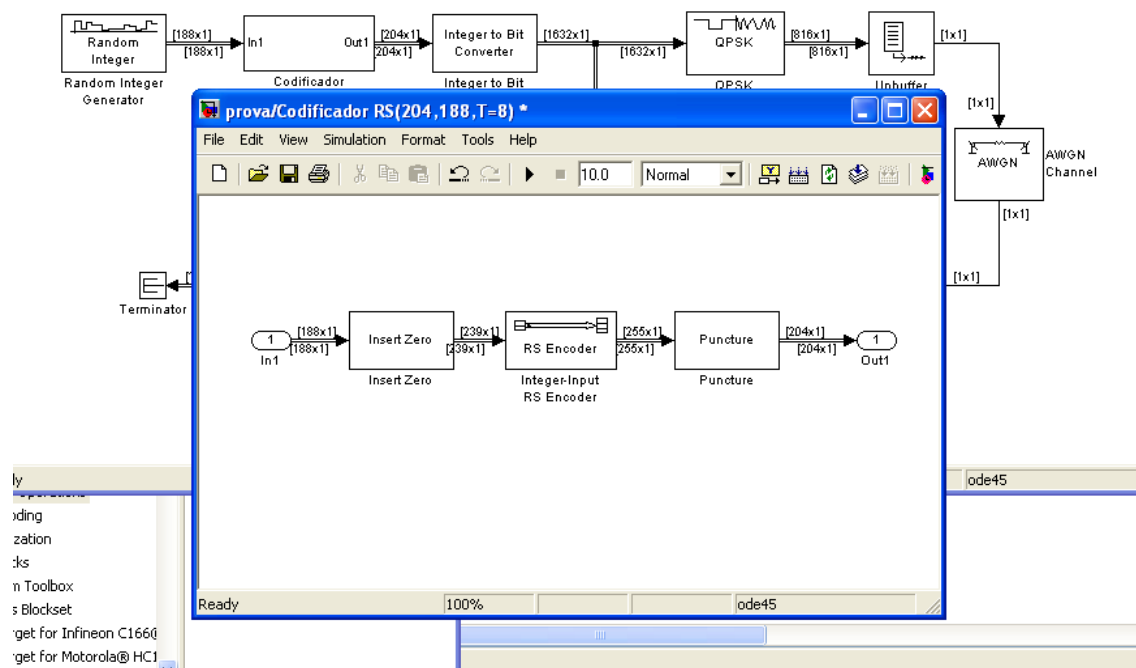


Figura 8 - Sub-sistema codificador de Reed-Solomon

Feu el sistema descodificador equivalent (connecteu un *Terminator* al port d'error del bloc descodificador RS). Si intentem executar ara el model, ens donarà error al bloc de càlcul de BER, i els tamanys d'alguns ports sortiran amb interrogant. Això vol dir que no sap quin és el tamany perquè hi ha alguna inconsistència al model.

La inconsistència la trobem al bloc de *Buffer*. Aquest bloc ens està creant les trames amb les mostres que necessita el demodulador de QPSK. Abans les trames que arribaven al modulador QPSK eren de 1504 bits, que corresponien als 188 bytes de trama. Però ara les trames que li arriben al modulador són de 1632 bits, perquè la codificació RS ha canviat les trames de 188 bytes en trames de 204 bytes. Corregiu el tamany del buffer i executeu el model. Quin BER ens dona ara? Torna a donar un BER massa gros.

Per obtenir el BER correcte hem d'ajustar primer bé el canal. Canviem la SNR a 20 dB i calculem el nou temps de símbol, tenint en compte que ara amb el mateix temps de trama tenim 204 bytes en comptes de 188 bytes.

A continuació ajustem el delay del mòdul per reflectir el nou tamany del buffer. Recordeu que el delay que introdueix el buffer és l'equivalent al nombre de mostres que ha de tenir la trama que crea.

Executeu el programa i comproveu que efectivament el BER torna a ser zero.

Ara afegirem un altre bloc de *Error Rate Calculation* i un altre *Display* per mirar l'efecte de la codificació Reed-Solomon. Compararem les mostres de l'emissor abans de la codificació amb les mostres del receptor després de la descodificació. Heu de posar un conversor de sencer a bit per poder fer el càlcul sobre les trames de sencers.

Ajusteu el delay adientment i comproveu que el BER també us dona 0, com a la figura 9.

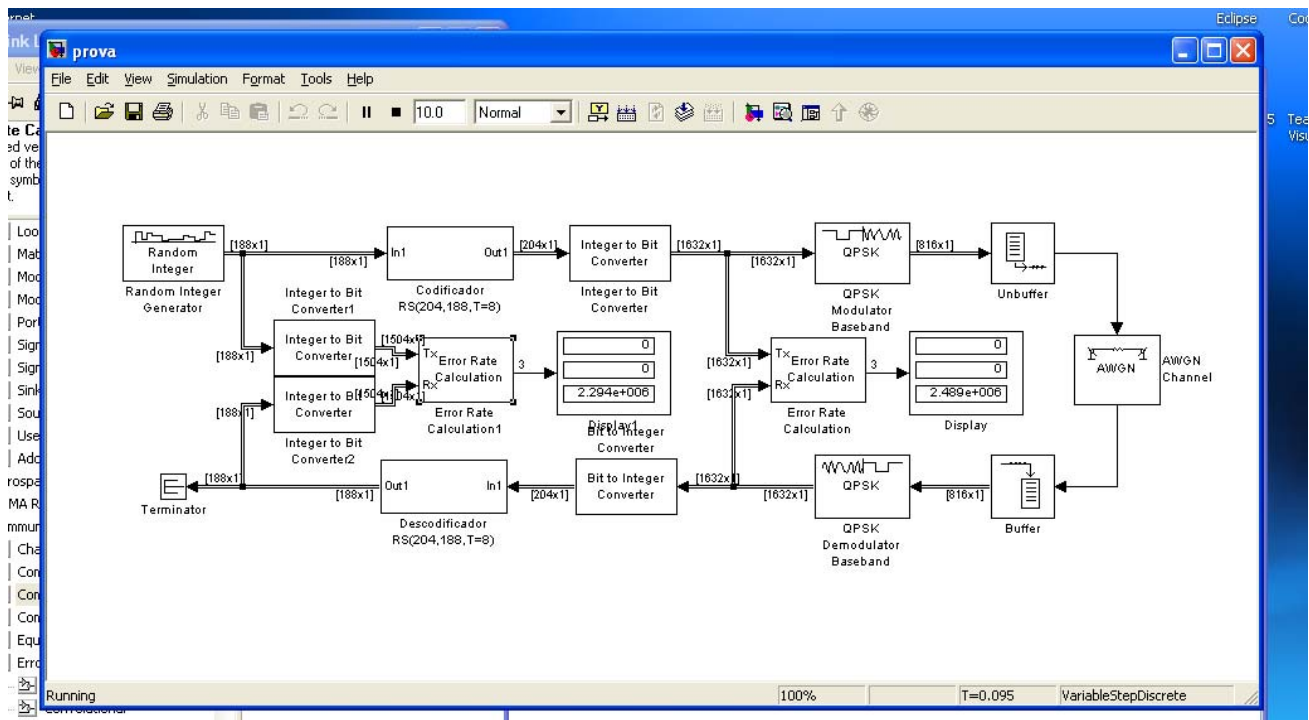


Figura 9 - Sistema complet amb codificació RS

Exercici 3. Poseu el valor de SNR de canal anteriorment trobat per transmissió QEF. Comproveu que encara és QEF abans de la codificació (si no heu calculat bé el període de símbol, el BER per aquest SNR haurà canviat). Quin és el BER real ara, el que tenim després de la codificació? Amb quin nou valor de SNR tenim ara un sistema QEF? Quans dB's hem guanyat?

Entrellaçat extern

Per acabar la primera part de la pràctica, afegirem el bloc d'entrellaçat intern. Afegiu un bloc *Convolutional Interleaver* a l'emissor, i un bloc *Convolutional Deinterleaver* al receptor. Ajusteu els paràmetres segons la descripció del bloc al capítol de referència.

Afegiu ara els blocs necessaris per calcular el BER amb aquest bloc. Ajusteu el delay del bloc de càlcul de BER tenint en compte que el delay que introdueix el parell de blocs de entrellaçat /desentrellaçat és, segons MATLAB,

Delays of Convolutional Interleavers

After a sequence of symbols passes through a convolutional interleaver and a corresponding convolutional deinterleaver, the restored sequence lags behind the original sequence. The delay, measured in symbols, between the original and restored sequences is

$(\text{Number of shift registers}) * (\text{Maximum delay among all shift registers})$

for the most general multiplexed interleaver. If your model incurs an additional delay between the interleaver output and the deinterleaver input, then the restored sequence lags behind the original sequence by the sum of the additional delay and the amount in the preceding formula

Convolutional Interleaver block. In the special case implemented by the Convolutional Interleaver/Convolutional Deinterleaver pair, note that the number of shift registers is the Rows of shift registers parameter, while the maximum delay among all shift registers is $B * (N-1)$ where B is the Register length step parameter and N is the Rows of shift registers parameter.

Enrecordeu-vos-en de tornar a ajustar el delay del bloc de càlcul de BER per la codificació RS.

Ara ja podeu simular el sistema amb la codificació externa completa (figura 10, a la següent pàgina).

Exercici 4. L'entrellaçat s'afegeix per millorar el comportament del sistema sobre errors de ràfega (varis errors consecutius), ja que aquestes ràfegues queden distribuïdes uniformement sobre la trama actual i la següent trama. No simularem aquest tipus de comportament de canal. Comproveu, però, com l'entrellaçat millora (si es que ho fa) el comportament del sistema en front de soroll blanc gaussià.

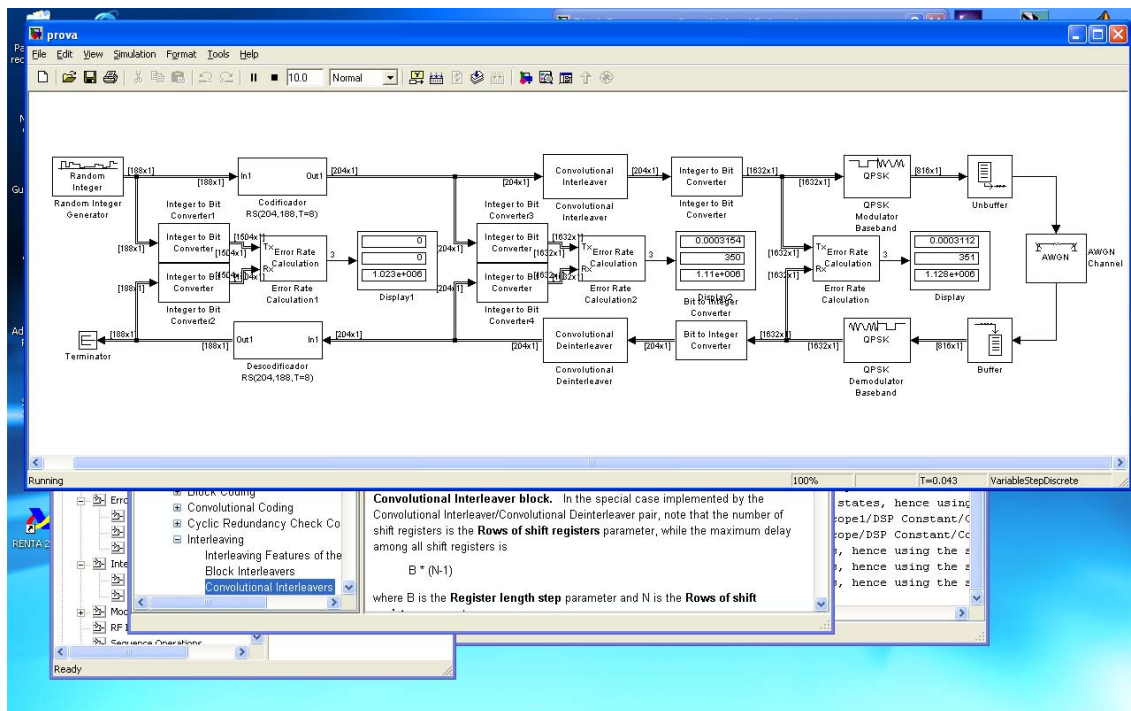


Figura 10 - Sistema amb codificació externa completa