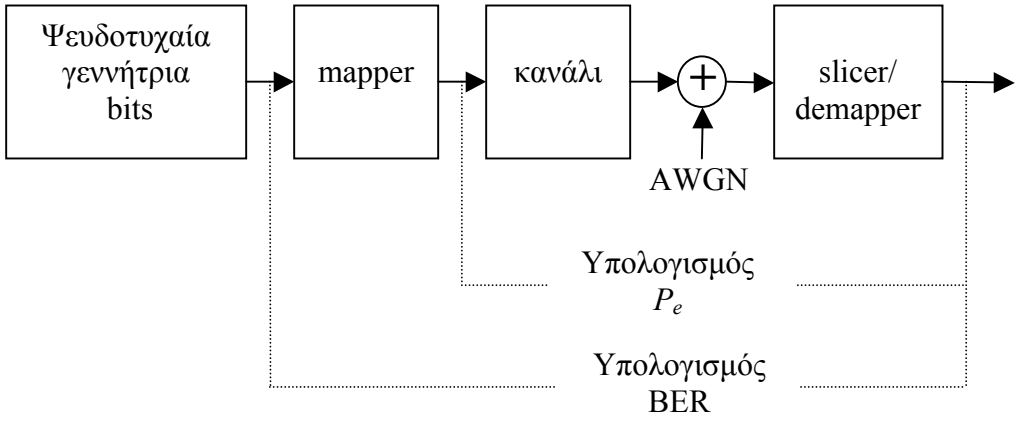
EE725 – Ειδικά Θέματα Ψηφιακών Επικοινωνιών
Δημήτρης Τουμπακάρης
20/04/2010

Τελική εργασία μαθήματος – Μέρος 1ο

Στο πρώτο μέρος της εργασίας θα υλοποιηθεί ένα απλό σύστημα διαμόρφωσης/αποδιαμόρφωσης και μετάδοσης μέσα από κανάλι AWGN και θα γίνουν γραφικές παραστάσεις του BER και της πιθανότητας σφάλματος συμβόλου, P_e, ως συνάρτηση του SNR. Το διάγραμμα του συστήματος φαίνεται στο παρακάτω σχήμα.



Ο θόρυβος θα θεωρηθεί AWGN. Επομένως, θα θεωρήσουμε κανάλι με κέρδος 1 και άπειρο εύρος ζώνης. Συνεπώς, $\mathbf{y}=\mathbf{x}+\mathbf{n}$, όπου $\mathbf{x}$, $\mathbf{y}$ και $\mathbf{n}$ είναι διανύσματα N διαστάσεων (ανάλογα με τον αριθμό συναρτήσεων βάσης που χρησιμοποιούμε). Θα θεωρήσουμε, επίσης, ότι ο θόρυβος σε κάθε διάσταση έχει την ίδια μέση ισχύ.

1α) Υλοποιήστε ένα mapper για διαμόρφωση PAM και έναν για διαμόρφωση QAM. Ο mapper θα δέχεται ως είσοδο τον αριθμό b των bits ανά σύμβολο του αστερισμού και μια ακολουθία b bits και θα παράγει το αντίστοιχο σύμβολο του αστερισμού (2 τιμές ανά σύμβολο για την QAM). Η αντιστοιχία ακολουθιών bits – σημάτων μπορεί να γίνει όπως εσείς θέλετε, αλλά θα πρέπει να διευκρινίσετε σαφώς ποια είναι. Εάν χρησιμοποιήσετε C++ μπρείτε να δημιουργήσετε το αντικείμενο της κλάσης mapper χρησιμοποιώντας το b και να μην το χρησιμοποιείτε στις διαδοχικές κλήσεις του αντικειμένου. Στο τέλος του κειμένου επισυνάπτεται ο QAM mapper του VDSL2 τον οποίο μπορείτε να υλοποιήσετε, εάν αυτό σας διευκολύνει. Αγνοήστε την περίπτωση κώδικα Trellis. Με βάση τον QAM mapper σκεφτείτε πώς μπορείτε να υλοποιήσετε τον PAM mapper. Μπορείτε, ωστόσο, να διαλέξετε mapper/demapper οποιουδήποτε άλλου standard ή και δικής σας επιλογής, αρκεί να διευκρινίσετε. Για το mapper να **μη** χρησιμοποιηθούν πίνακες, αλλά πράξεις. Μπορείτε να χρησιμοποιήσετε κάποιους βοηθητικούς πίνακες, αλλά <u>σε καμία περίπτωση</u> ολόκληρο τον πίνακα της απεικόνισης από την

ακολουθία bits στο διαμορφωμένο σήμα. Δηλαδή, θα ληφθεί υπόψη όχι μόνο η σωστή υλοποίηση του mapper, αλλά και η εξοικονόμηση μνήμης και λειτουργιών ανάγνωσης από τη μνήμη.
Η υλοποίηση μπορεί να γίνει σε C, C++, Matlab ή σε οποιαδήποτε άλλη γλώσσα κατόπιν συνεννόησης με το διδάσκοντα. Η υλοποίηση του QAM mapper για περιττό (odd) αριθμό bits (CR-QAM) είναι προαιρετική. Ωστόσο πρέπει να υποστηρίζεται οποιαδήποτε άρτια (even) τιμή *b*. Ο PAM mapper πρέπει να υποστηρίζει όλες τις τιμές του *b*.

Σημείωση: Θα πρέπει να υλοποιήσετε εσείς τον PAM mapper. Δηλαδή, δεν επιτρέπεται να χρησιμοποιήσετε π.χ. απευθείας την qammod της Matlab ή modules της Simulink. Επιτρέπεται να προσαρμόσετε ανοικτό κώδικα π.χ. της Matlab αρκεί
I. Ο κώδικας που θα δημιουργήσετε να κάνει *ακριβώς* αυτά που ζητούνται.
II. Να μην αντιγράψετε κώδικα από άλλη εργασία εντός ή εκτός του Πανεπιστημίου.

1β) Υλοποιήστε τους αντίστοιχους slicers/demappers του δέκτη. Ο slicer/demapper θα δέχεται τον αριθμό *b* των bits και το ληφθέν σύμβολο **y** (δύο τιμές για QAM) και θα παρέχει στην έξοδο τα αποκωδικοποιημένα bits καθώς και το εκτιμηθέν σύμβολο <**x**>. Για την υλοποίηση ισχύουν τα ίδια σχετικά με τη χρήση πινάκων όπως και στο 1α. Εάν προτιμάτε μπορείτε να υλοποιήσετε το slicer (απόφαση για το <**x**>) και το demapper (από το <**x**> στα bits) ξεχωριστά. Και στην περίπτωση του slicer η υλοποίηση θα πρέπει να είναι αποδοτική. **Δε** θα πρέπει να χρησιμοποιήσετε πίνακες για να συγκρίνετε το ληφθέν σήμα με τα σήματα του αστερισμού. Υλοποίηση με χρήση πινάκων **δε** θα γίνει δεκτή. Θα πρέπει να βρείτε τρόπο να συγκρίνετε το **y** μόνο με όσα **x** απαιτούνται για να αποφασίσετε ποια είναι η τιμή του <**x**>. Αφού βρείτε το <**x**>, το demapping στα bits θα πρέπει να γίνει με χρήση πράξεων και όχι με πίνακα. Προσοχή: Για όσους υλοποιήσουν το slicer/demapper και για τις προαιρετικές τιμές μονού *b*: η βέλτιστη υλοποίηση του slicer για αστερισμούς σταυρού (CR-QAM) δεν είναι τετριμμένη.

Σημείωση: Ισχύουν και εδώ τα περί έτοιμων συναρτήσεων, όπως και στο προηγούμενο ερώτημα 1α).

1γ) Δημιουργήστε μια ψευδοτυχαία ακολουθία bits με χρήση έτοιμων συναρτήσεων της γλώσσας προγραμματισμού που χρησιμοποιείτε και μεταδώστε τη με διαδοχικές κλήσεις του mapper και slicer/demapper που υλοποιήσατε χωρίς να χρησιμοποιήσετε θόρυβο. Βεβαιωθείτε ότι BER=0 και P_e=0. Δοκιμάστε διαφορετικά μεγέθη αστερισμών.

1δ) Εισάγετε θόρυβο και υπολογίστε τις BER και P_e για διαφορετικές τιμές του SNR και διαφορετικά μεγέθη αστερισμών. Για να μεταβάλετε τον SNR μπορείτε να μεταβάλετε είτε τη μέση ενέργεια του θορύβου ή τη μέση ενέργεια του αστερισμού. Βεβαιωθείτε ότι καθώς αλλάζετε το μέγεθος του αστερισμού δε μεταβάλλετε τη μέση

ενέργειά του. Προσοχή στη σταθερά με την οποία πρέπει να πολλαπλασιάσετε την έξοδο του mapper ώστε να έχετε το σωστό SNR στο δέκτη.

Παραδοτέα

- Παράδοση το αργότερο 2 εβδομάδες μετά την ημερομηνία εξέτασης για την εξεταστική στην οποία θέλετε να βαθμολογηθείτε (Ιούλιος ή Σεπτέμβριος 2009). Ωστόσο, εάν στην εργασία υπάρχουν λάθη ή εάν δεν έχουν γίνει αυτά που ζητούνται, ο διδάσκων δε δεσμεύεται για την έγκαιρη παράδοση της βαθμολογίας.
- Επιτρέπεται (και ενθαρρύνεται) η συνεργασία αλλά όχι η αντιγραφή ή ο καταμερισμός της εργασίας. Ο καθένας πρέπει να γράψει το δικό του κώδικα και να παραδώσει τη δική του αναφορά. Επίσης, ενθαρρύνεται η επικοινωνία με το διδάσκοντα για οποιαδήποτε βοήθεια χρειαστείτε.

Π1. Το πρόγραμμα υλοποίησης του πομποδέκτη του σχήματος σε ηλεκτρονική μορφή με όλες τις συναρτήσεις (mapper, demapper/slicer κλπ). Η υποστήριξη αστερισμών QAM για μονές τιμές b είναι προαιρετική. Για τη δημιουργία θορύβου και ψευδοτυχαίων bits μπορείτε να χρησιμοποιήσετε συναρτήσεις βιβλιοθήκης. Ο χρήστης θα πρέπει να καλεί ένα πρόγραμμα (π.χ. simulator.exe ή simulator.m) το οποίο, με τη σειρά του, θα καλεί τις συναρτήσεις δημιουργίας ψευδοτυχαίων bits, mapping, δημιουργίας και προσθήκης AWGN, slicing/demapping, υπολογισμού BER κλπ. Θα πρέπει να δώσετε αρκετές πληροφορίες (είτε σε συνοδευτικό κείμενο ή με σχόλια στον κώδικα) ώστε κάποιος χρήστης να μπορέσει να τρέξει και να καταλάβει σχετικά γρήγορα τον κώδικά σας. Στις πληροφορίες θα πρέπει να περιλαμβάνονται συνοπτικές οδηγίες χρήσης και περιγραφή του τι κάνει όχι μόνο ο simulator, αλλά και κάθε συνάρτηση ξεχωριστά. Συνιστάται η παράδοση ενός αρχείου zip με όλα τα απαραίτητα αρχεία για να «τρέξει» το πρόγραμμα. Καλό είναι να αποφύγετε χρήση συναρτήσεων οι οποίες χρειάζονται συγκεκριμένους compilers/βιβλιοθήκες για να τρέξουν. Εργασίες χωρίς επαρκή σχόλια στον κώδικα και χωρίς επαρκείς επεξηγήσεις δε θα γίνουν δεκτές.

Π2. Δημιουργήστε γραφικές παραστάσεις (plots) του BER ως συνάρτηση του SNR και του P_e ως συνάρτηση του SNR για τιμές του BER και P_e από 1 έως 10^{-7}. Εάν σας βολεύει μπορείτε να υπερθέσετε πολλές καμπύλες στην ίδια γραφική παράσταση. Βεβαιωθείτε ότι χρησιμοποιήσατε ικανό αριθμό bits για τις προσομοιώσεις. Δε χρειάζεται να κάνετε προσομοιώσεις για πάρα πολλά σημεία της γραφικής παράστασης, αρκεί να χρησιμοποιήσετε τόσα ώστε να έχετε μια καλή προσέγγιση. Οι γραφικές παραστάσεις να είναι για BPSK, 4-PAM, 8-PAM, QPSK, 16-QAM και 64-QAM. Πριν παραδώσετε τις γραφικές παραστάσεις (εκτυπωμένες ή σε αρχείο) συγκρίνετε με τη βιβλιογραφία για να βεβαιωθείτε ότι τα αποτελέσματά σας είναι σωστά. Εργασίες με

γραφικές παραστάσεις που δε συμφωνούν με τη βιβλιογραφία δε θα γίνουν δεκτές, εκτός αν υπάρχει λόγος για την ασυμφωνία.

Π3. Συκρίνετε τα αποτελέσματα των προσομοιώσεων για την P_e με το Union Bound, το NNUB και το προσεγγιστικό NNUB για κάθε περίπτωση του Π2. Η σύγκριση μπορεί να γίνει με γραφικές παραστάσεις. Σχολιάστε.

Π4. Δημιουργήστε μια γραφική παράσταση της οποίας ο οριζόντιος άξονας να είναι το SNR και ο κάθετος να είναι ο αριθμός bits ανά διάσταση της διαμόρφωσης η οποία επιτυγχάνει $P_e = 10^{-6}$ για το SNR του οριζόντιου άξονα. Δηλαδή, για κάθε διαμόρφωση βρείτε για ποιο SNR επιτυγχάνεται $P_e = 10^{-6}$ και προσθέστε ένα σημείο στο διάγραμμα (π.χ. για τη BPSK ο οριζόντιος άξονας είναι SNR~13.7 dB και ο κάθετος b = 1 bit/μετάδοση). Στην ίδια γραφική παράσταση σχεδιάστε και τη χωρητικότητα καναλιού ως συνάρτηση του SNR. Σχολιάστε.

Παράρτημα: VDSL2 QAM mapper

10.3.3.2 Constellations

An algorithmic constellation mapper shall be used to construct sub-carrier QAM constellations with a minimum number of bits equal to 1 and a maximum number of bits equal to 15.
The constellation points are denoted (X,Y). X and Y shall lie at the odd integers ±1, ±3, ±5, etc. For convenience of illustration, each constellation point in Figures 10-9 through 10-13 is labeled by an integer whose unsigned binary representation is $(v_{b-1}\ v_{b-2}\ ...\ v_1\ v_0)$.

10.3.3.2.1 Even values of *b*

For even values of b, the integer values X and Y of the constellation point (X,Y) shall be determined from the b bits $(v_{b-1},\ v_{b-2},...,v_1,v_0)$ as follows. X and Y shall be odd integers with two's-compliment binary representations $(v_{b-1}\ v_{b-3}\ ...\ v_1\ 1)$ and $(v_{b-2}\ v_{b-4}\ ...\ v_0\ 1)$, respectively. The MSBs, v_{b-1} and v_{b-2}, shall be the sign bits for X and Y, respectively. Figure 10-9 shows example constellations for $b = 2$ and $b = 4$.

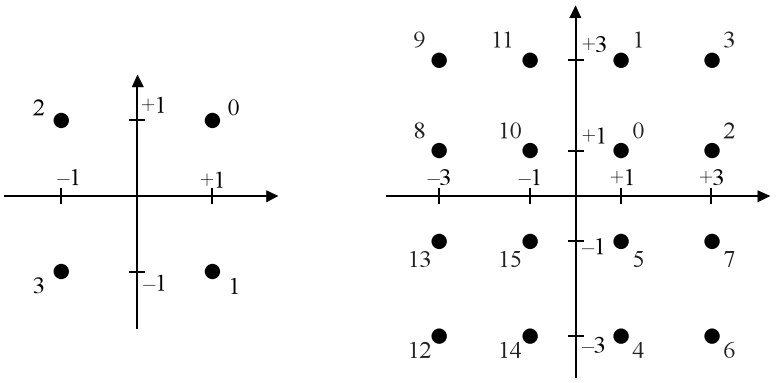


Figure 10-9 – Constellation labels for b = 2 and b = 4

NOTE – The 4-bit constellation shall be obtained from the 2-bit constellation by replacing each label *n* by the 2x2 block of labels:

4n + 1	4n+3
4n	4n+2

The same procedure shall be used to construct the larger even-bit constellations recursively. The constellations obtained for even values of *b* are square in shape.

10.3.3.2.2 Odd values of *b*

10.3.3.2.2.1 *b* = 1

Figure 10-10 shows the constellation for the case *b* = 1.

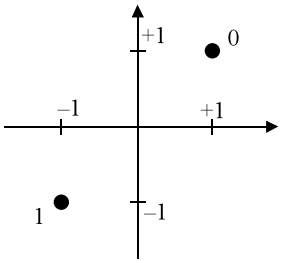


Figure 10-10/G.993.2 – Constellation labels for *b* = 1

When trellis coding is used, the 2-bit words generated by the trellis encoder shall be mapped on two 1-bit sub-carriers using the same labelling for 1-bit and 2-bit constellations as described above. An example for $v_0 = 1$ and $v_1 = 0$ is shown in Figure 10-11, in which the constellation for the 2-bit word is on the right-hand side of the diagram.

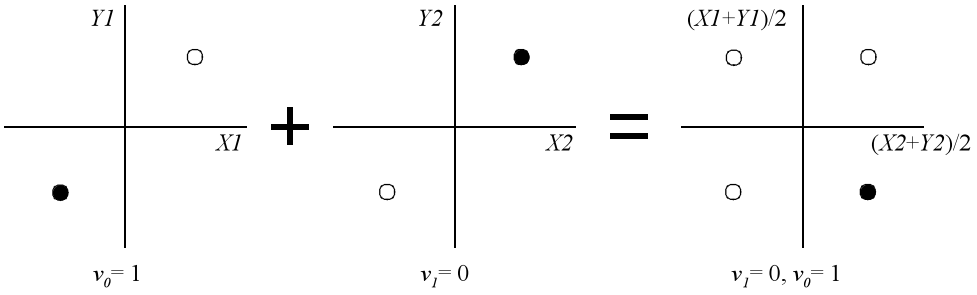


Figure 10-11/G.993.2 – Combination of a pair of 1-bit constellations to build a 2-bit constellation

10.3.3.2.2.2 *b* = 3

Figure 10-12 shows the constellation for the case $b = 3$.

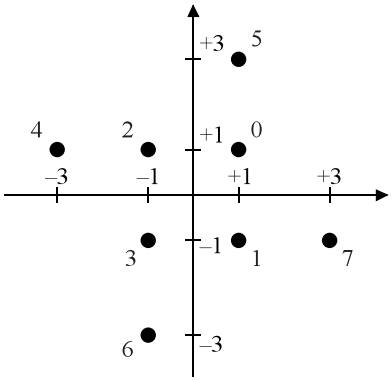


Figure 10-12/G.993.2 – Constellation labels for *b* = 3

10.3.3.2.2.3 *b* > 3

If b is odd and greater than 3, the two MSBs of X and the two MSBs of Y shall be determined by the five MSBs of the b bits ($v_{b-1}\ v_{b-2}\ ...\ v_1\ v_0$). Let $c = (b+1)/2$, then X and Y shall have the twos-compliment binary representations ($X_c\ X_{c-1}\ v_{b-4}\ v_{b-6}...v_3\ v_1\ 1$) and ($Y_c\ Y_{c-1}\ v_{b-5}\ v_{b-7}\ v_{b-9}...v_2\ v_0\ 1$), where X_c and Y_c are the sign bits of X and Y respectively. The relationship between X_c, X_{c-1}, Y_c, Y_{c-1}, and ($v_{b-1}\ v_{b-2}...v_{b-5}$) shall be as shown in Table 10-3.

Table 10-3/G.993.2 – Determining the top two bits of *X* and *Y*

$v_{b-1}\ v_{b-2}...v_{b-5}$	$X_c\ X_{c-1}$	$Y_c\ Y_{c-1}$
0 0 0 0 0	0 0	0 0
0 0 0 0 1	0 0	0 0
0 0 0 1 0	0 0	0 0
0 0 0 1 1	0 0	0 0
0 0 1 0 0	0 0	1 1
0 0 1 0 1	0 0	1 1
0 0 1 1 0	0 0	1 1
0 0 1 1 1	0 0	1 1
0 1 0 0 0	1 1	0 0
0 1 0 0 1	1 1	0 0
0 1 0 1 0	1 1	0 0
0 1 0 1 1	1 1	0 0
0 1 1 0 0	1 1	1 1
0 1 1 0 1	1 1	1 1
0 1 1 1 0	1 1	1 1
0 1 1 1 1	1 1	1 1
1 0 0 0 0	0 1	0 0
1 0 0 0 1	0 1	0 0
1 0 0 1 0	1 0	0 0
1 0 0 1 1	1 0	0 0
1 0 1 0 0	0 0	0 1
1 0 1 0 1	0 0	1 0
1 0 1 1 0	0 0	0 1
1 0 1 1 1	0 0	1 0
1 1 0 0 0	1 1	0 1
1 1 0 0 1	1 1	1 0
1 1 0 1 0	1 1	0 1
1 1 0 1 1	1 1	1 0
1 1 1 0 0	0 1	1 1
1 1 1 0 1	0 1	1 1
1 1 1 1 0	1 0	1 1
1 1 1 1 1	1 0	1 1

Figure 10-13 shows the constellation for the case $b = 5$.

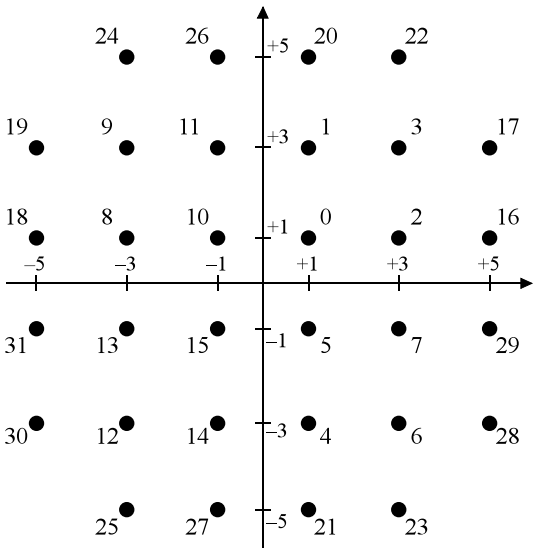


Figure 10-13/G.993.2 – Constellation labels for b = 5

NOTE – The 7-bit constellation shall be obtained from the 5-bit constellation by replacing each label *n* by the 2 x 2 block of labels:

$4n + 1$	$4n+3$
$4n$	$4n+2$

The same procedure shall then be used to construct the larger odd-bit constellations recursively.