

Ψηφιακές Τηλεπικοινωνίες
Ακαδημαϊκό Έτος 2016/2017

Γεράσιμος Ρόμπολας AM: 5636

3 Ιανουαρίου 2017

Περιεχόμενα

Περιεχόμενα	1
1 Εισαγωγή	2
2 Περιβάλλον Εκτέλεσης	2
3 Συμπίεση Διακριτής Πηγής με Χρήση της Κωδικοποίησης DPCM	3
3.1 Υλοποίηση συστήματος DPCM	3
3.2 Ανάλυση Αρχικού Σήματος και Σφάλματος Πρόβλεψης	4
3.3 Ανάλυση Απόδοσης του Φίλτρου Πρόβλεψης	8
3.4 Ανάλυση Αρχικού και Ανακατασκευασμένου Σήματος	11
4 Προσομοίωση Ομόδυνου Ζωνοπερατού Συστήματος M-PSK	15
4.1 Περιγραφή	15
4.1.1 Mapper	16
4.1.2 Modulator	16
4.1.3 AWGN Κανάλι	17
4.1.4 Demodulator	17
4.1.5 Detector	17
4.1.6 Demapper	18
4.1.7 Script	18
4.2 Μετρήσεις BER	19
Αναφορές	22

1 Εισαγωγή

Στα πλαίσια της συγκεκριμένης εργασίας πραγματοποιείται η υλοποίηση και ανάλυση δύο βασικών συστημάτων των ψηφιακών τηλεπικοινωνιών. Συγκεκριμένα, τα συστήματα που περιγράφονται παρακάτω είναι τα εξής:

- Σύστημα Συμπίεσης Διακριτής Πηγής με Χρήση της Κωδικοποίησης DPCM
- Σύστημα Προσομοίωσης Ομόδυνου Ζωνοπερατού Συστήματος M-PSK

Στη συνέχεια παρουσιάζονται αναλυτικά οι αλγόριθμοι υλοποίησης των παραπάνω συστημάτων, καθώς και οι αντίστοιχες μετρήσεις/μετρικές για την αξιολόγηση της απόδοσής τους, υπό συγκεκριμένες συνθήκες εκτέλεσης.

2 Περιβάλλον Εκτέλεσης

Τα χαρακτηριστικά του περιβάλλοντος που χρησιμοποιήθηκε για τον υλοποίηση και την αξιολόγηση των διαφόρων συστημάτων περιγράφονται παρακάτω.

Λειτουργικό Σύστημα	Windows 10 64-bit
Γλώσσα Προγραμματισμού	Matlab R2015a

3 Συμπίεση Διακριτής Πηγής με Χρήση της Κωδικοποίησης DPCM

3.1 Υλοποίηση συστήματος DPCM

Για την κατασκευή του συστήματος κωδικοποίησης DPCM υλοποιήθηκε αρχικά ένας ομοιόμορφος κβαντιστής N δυαδικών ψηφίων, δηλαδή 2^N επιπέδων. Κατά την υλοποίησή του τα όρια κάθε επιπέδου θεωρήθηκαν ως εξής¹:

$$(+\infty, +a)$$

$$[+a, 0)$$

$$[0, -a)$$

$$[-a, -\infty)$$

Στη συνέχεια, για την υλοποίηση του συστήματος DPCM απαιτήθηκε η υλοποίηση του υποσυστήματος φίλτρου πρόβλεψης. Η υλοποίηση του φίλτρου πρόβλεψης βασίζεται στο γραμμικό συνδυασμό των p προηγούμενων τιμών, οι οποίες βρίσκονται απηθηκευμένες στη μνήμη. Ωστόσο κατά την αρχικοποίηση του υποσυστήματος δεν διατίθενται τιμές αποθηκευμένες στην μνήμη. Επομένως για να αποκτήσει κάποιες αρχικές τιμές το υποσύστημα, ώστε να μπορεί να λειτουργήσει, θεωρήθηκε ότι οι p πρώτες τιμές μεταδίδονται αυτούσιες, δίχως σφάλματα.

¹Αναπάρσταση κβαντιστή 4 επιπέδων.

3.2 Ανάλυση Αρχικού Σήματος και Σφάλματος Πρόβλεψης

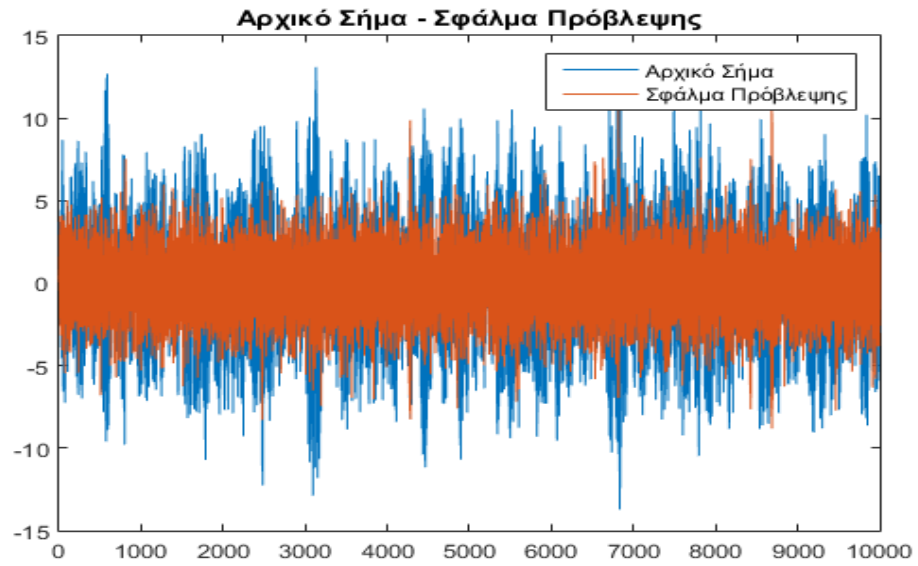
Για την αξιολόγηση της απόδοσης του προβλέπτη του συστήματος DPCM, λήφθηκαν οι αντίστοιχες μετρήσεις του σφάλματος πρόβλεψης για $p = 4, 10$ και $N = 1, 2, 3 \text{ bits}$. Συγκεκριμένα παρουσιάζονται παρακάτω οι μετρήσεις του σφάλματος πρόβλεψης που λήφθηκαν σε σχέση με το αρχικό σήμα.

Από τις συγκεκριμένες γραφικές παραστάσεις είναι εμφανές ότι το σφάλμα πρόβλεψης επηρεάζεται μόνο από τον αριθμό bits κβάντισης, ανεξάρτητα του πλήθους p των προηγούμενων τιμών, στον οποίον στηρίζεται ο προβλέπτης. Συγκεκριμένα παρατηρείται αναλογική βελτίωση του σφάλματος κβάντισης όσο αυξάνεται ο αριθμός bits κβάντισης των δειγμάτων. Το γεγονός αυτό οφείλεται στη μεγαλύτερη ακρίβεια που μας προσφέρουν τα επιπλέον bits κβάντισης, με αποτέλεσμα ο προβλέπτης να έχει την δυνατότητα να προσφέρει μεγαλύτερη αξιοπιστία των προβλεπόμενων τιμών.

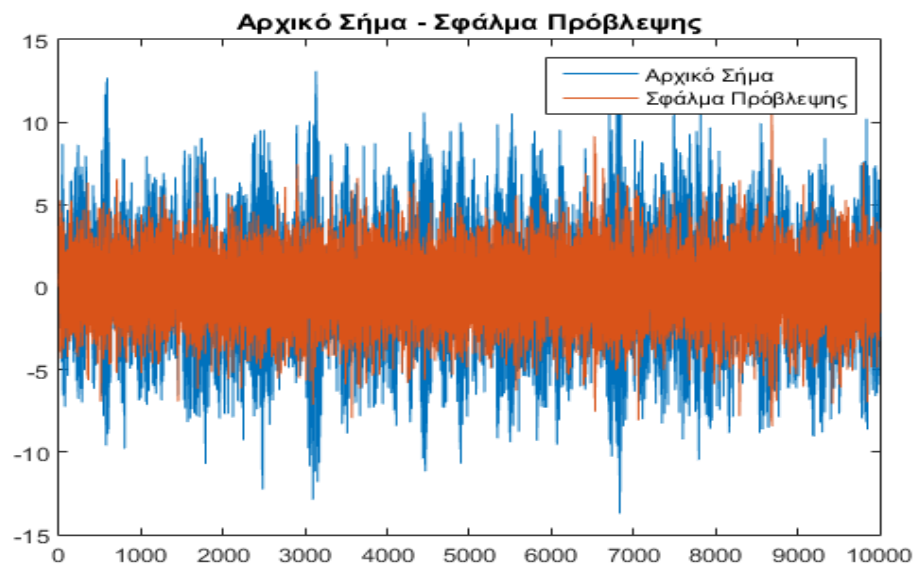
Ωστόσο το ίδιο δεν συμβαίνει καθώς αυξάνεται ο αριθμός p των προηγούμενων τιμών. Συγκεκριμένα είναι εμφανές ότι ο αριθμός p επηρεάζει ελάχιστα την απόδοση του προβλέπτη, υπο τη συνθήκη πάντα ότι $p \geq 4^2$. Στη συνέχεια διερευνάται αναλυτικότερα η επίδραση των p προηγούμενων τιμών στη διαδικασία πρόβλεψης.

²Χρησιμοποιήθηκε ως είσοδος μια Auto Regressive διαδικασία τάξης 4.

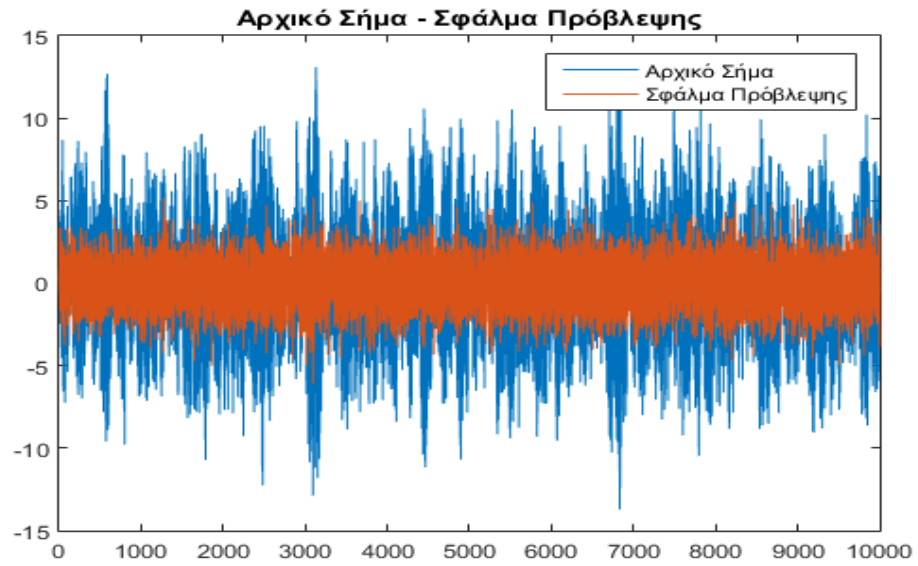
- Για $\text{bits} = 1$ & $p = 4$:



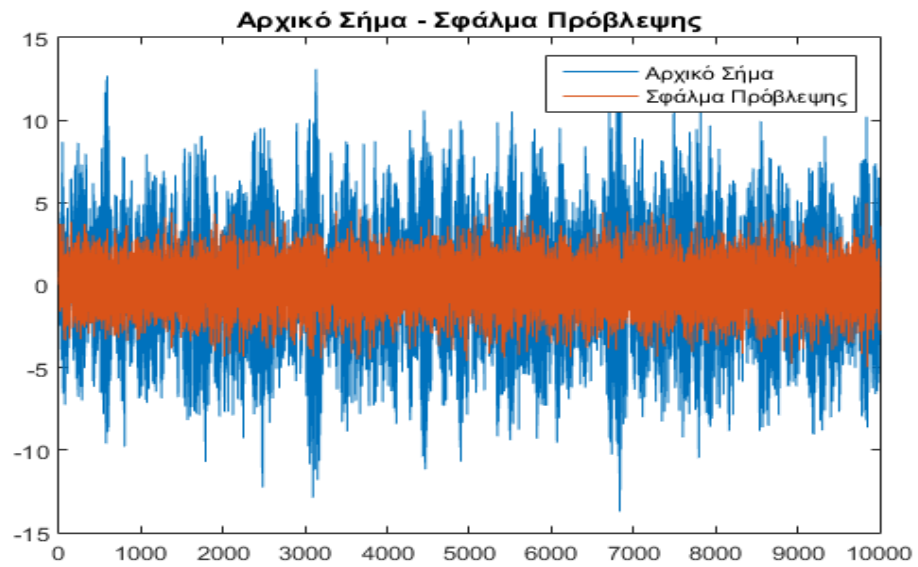
- Για $\text{bits} = 1$ & $p = 10$:



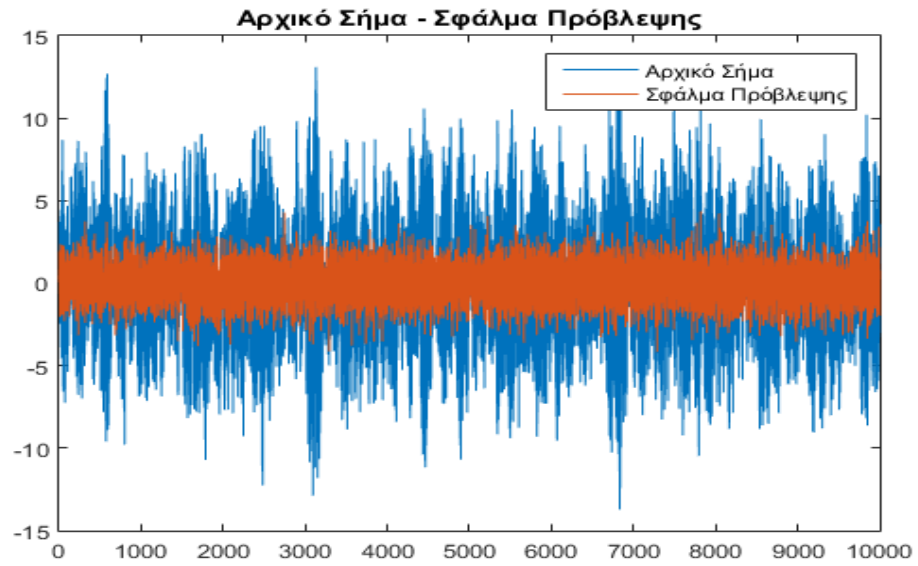
- Για $\text{bits} = 2$ & $p = 4$:



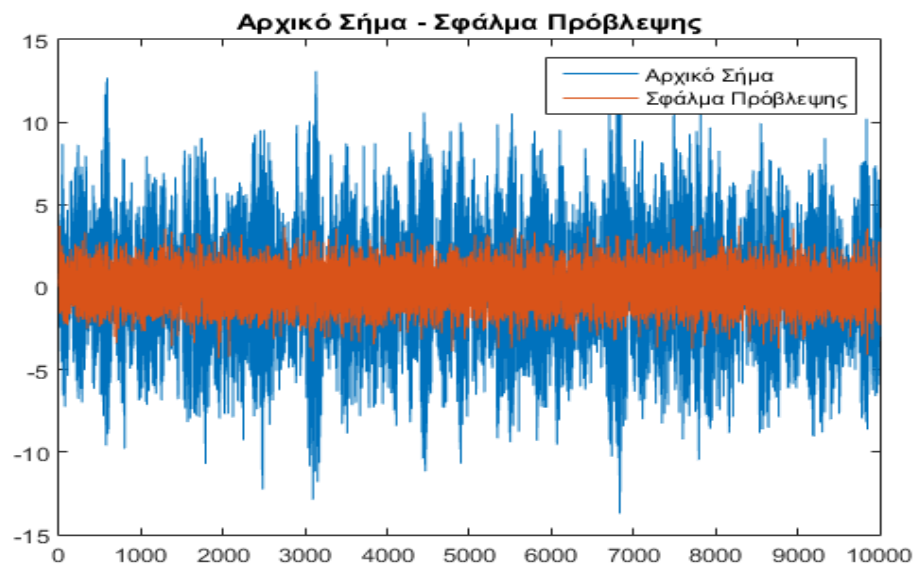
- Για $\text{bits} = 2$ & $p = 10$:



- Για $\text{bits} = 3$ & $p = 4$:



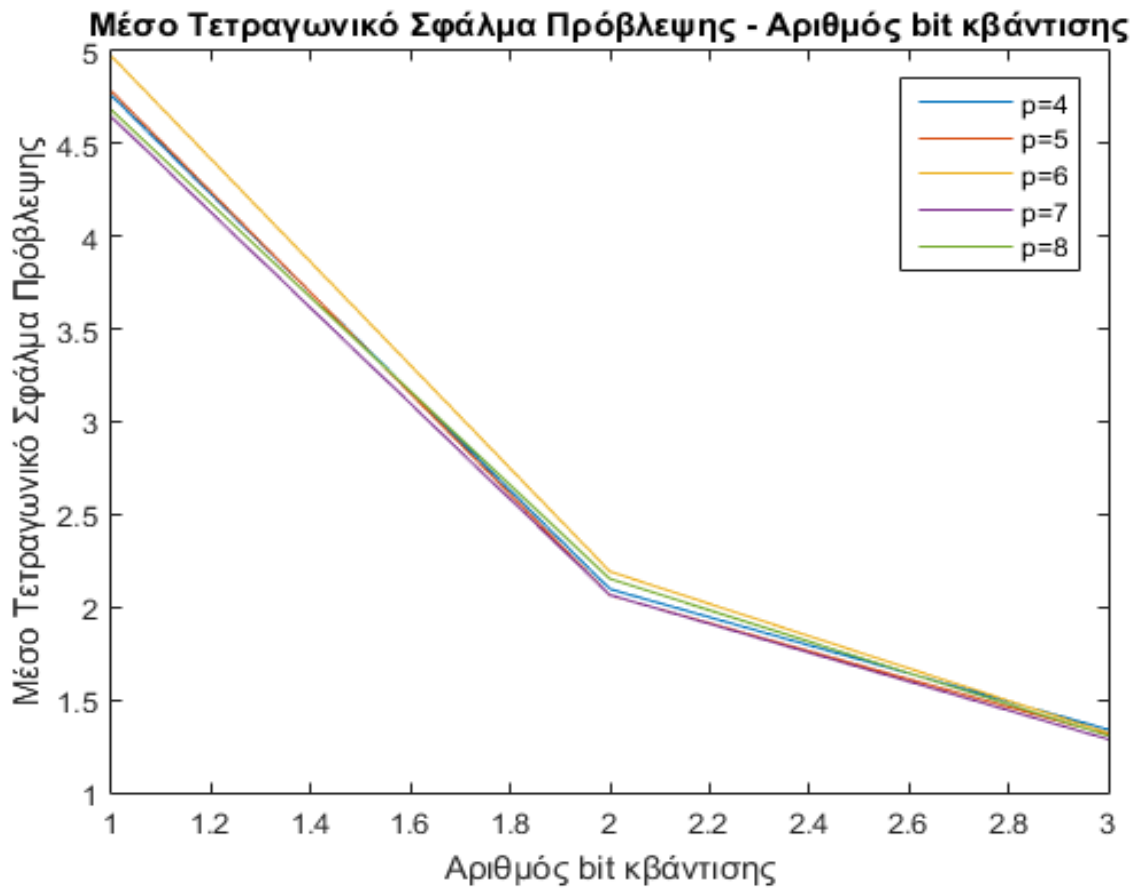
- Για $\text{bits} = 3$ & $p = 10$:



3.3 Ανάλυση Απόδοσης του Φίλτρου Πρόβλεψης

Για την περαιτέρω αξιολόγηση του προβλέπτη του συστήματος DPCM, λήφθηκαν οι αντίστοιχες τιμές του προβλεπόμενου σήματος για $p = 4 : 8$ και $N = 1, 2, 3 \text{ bits}$. Συγκεκριμένα για κάθε συνδυασμό των παραμέτρων p και N υπολογίστηκε και απεικονίστηκε αντίστοιχα το μέσο τετραγωνικό σφάλμα πρόβλεψης³ ως:

$$E(y^2) = E((x - y')^2)$$



³Όπου x το Αρχικό Σήμα & y' η Πρόβλεψη του x .

Μέσω του παραπάνω γραφήματος, είναι πλέον διακριτό το γεγονός ότι πράγματι η απόδοση του φίλτρου πρόβλεψης του σήματος επηρεάζεται σχεδόν αποκλειστικά από την ακρίβεια κβάντισης των δειγμάτων. Όσο περισσότερα bits κβάντισης χρησιμοποιούνται, τόσο περισσότερο αξιόπιστα προσεγγίζει το αρχικό σήμα ο προβλέπτης.

Είναι εμφανές επίσης, ότι η απόδοσή του προβλέπτη καθώς διακυμαίνεται για διαφορετικές τιμές του p παραμένει σχεδόν σταθερή. Ωστόσο, το γεγονός αυτό αντιφάσκει τη θεωρία της απόδοσης του φίλτρου, καθώς ήταν αναμενόμενο να βελτιώνεται η απόδοσή του όσο αυξάνεται η τιμή του p . Συγκεκριμένα εφόσον ο προβλέπτης χρησιμοποιεί περισσότερες τιμές δειγμάτων της μνήμης, όφειλε να προσεγγίζει αρκετά καλύτερα τις μελλοντικές τιμές του σήματος, επιτυγχάνοντας καλύτερη απόδοση.

Για το λόγο αυτό, για $p = 4 : 8$ καταγράφηκαν επίσης οι αντίστοιχες κβαντισμένες τιμές των συντελεστών του προβλέπτη a_i , $i = 1 : 8$, όπως παρουσιάζονται παρακάτω:

	p=4	p=5	p=6	p=7	p=8
a_1	1,3828125	1,3828125	1,3828125	1,3828125	1,3828125
a_2	-1,5234375	-1,5234375	-1,5234375	-1,5234375	-1,5234375
a_3	1,2109375	1,2109375	1,2109375	1,2109375	1,2109375
a_4	-0,3046875	-0,3046875	-0,3046875	-0,3046875	-0,3046875
a_5	-	-0,0078125	0,0078125	-0,0078125	0,0078125
a_6	-	-	-0,0078125	0,0078125	-0,0078125
a_7	-	-	-	-0,0078125	0,0078125
a_8	-	-	-	-	-0,0078125

Παρατηρώντας τους συντελεστές του προβλέπτη, είναι εμφανές ότι οι για $p = 1 : 4$ παρουσιάζει μεγάλες σχετικά τιμές σε σχέση με τις υπόλοιπες. Ενώ για $p \geq 5$ οι τιμές των συντελεστών παρουσιάζουν μικρές αλλά και κοινές τιμές κατά απόλυτη τιμή. Συνεπώς, το γεγονός αυτό έχει ως αποτέλεσμα ο προβλέπτης να δίνει μεγαλύτερη βαρύτητα στις 4 προηγούμενες τιμές των δειγμάτων, έχοντας μια μικρή επιρροή από όλες τις επόμενες.

Παρατηρώντας ωστόσο το γεγονός ότι το αρχικό σήμα εισόδου είναι κατασκευασμένο μέσω μιας Auto Regressive διαδικασίας τάξης 4, συνεπάγεται ότι κάθε τιμή της συγκεκριμένης διαδικασίας δημιουργήθηκε βασιζόμενη αποκλειστικά στις 4 άμεσα προηγούμενες τιμές. Επομένως είναι λογικό το φίλτρο πρόβλεψης να αναθέτει μεγαλύτερους συντελεστές βαρύτητας στις αντίστοιχες αυτές τιμές, προσφέροντας την καλύτερη δυνατή πρόβλεψη όταν στηρίζεται αποκλειστικά σε αυτές.

Αντίστοιχα για $p \geq 5$ οι συντελεστές του φίλτρου του προβλέπτη προσφέρουν μια τυχαία επιπλέον βαρύτητα στην εκτίμηση της πρόβλεψης, καθώς δεν προσφέρουν ουσιαστικά καμία βελτίωση σε αυτήν. Συγκεκριμένα, το μόνο που επιτυγχάνουν είναι να δυσχαιρένουν την απόδοση του προβλέπτη, καθώς τον τροφοδοτούν με θόρυβο.

3.4 Ανάλυση Αρχικού και Ανακατασκευασμένου Σήματος

Για την αξιολόγηση της απόδοσης του συνολικού συστήματος DPCM, λήφθηκαν οι αντίστοιχες τιμές του ανακατασκευασμένου σήματος στο δέκτη για $p = 4, 8$ και $N = 1, 2, 3 \text{ bits}$. Συγκεκριμένα για κάθε συνδυασμό των παραμέτρων p και N απεικονίστηκε αντίστοιχα το αρχικό και το ανακατασκευασμένο σήμα σε κοινά γραφήματα. Για την καλύτερη εξαγωγή συμπερασμάτων, επειδή τα δύο σήματα σχεδόν ταυτίζονται, επιλέχθηκε να αναπαρασταθεί σε κάθε περίπτωση μία μικρή αντιπροσωπευτική περιοχή των δειγμάτων του σήματος. Συγκεκριμένα επιλέχθηκε η αναπαράσταση 150 δειγμάτων (2000:2149), όπως παρουσιάζονται αντίστοιχα παρακάτω.

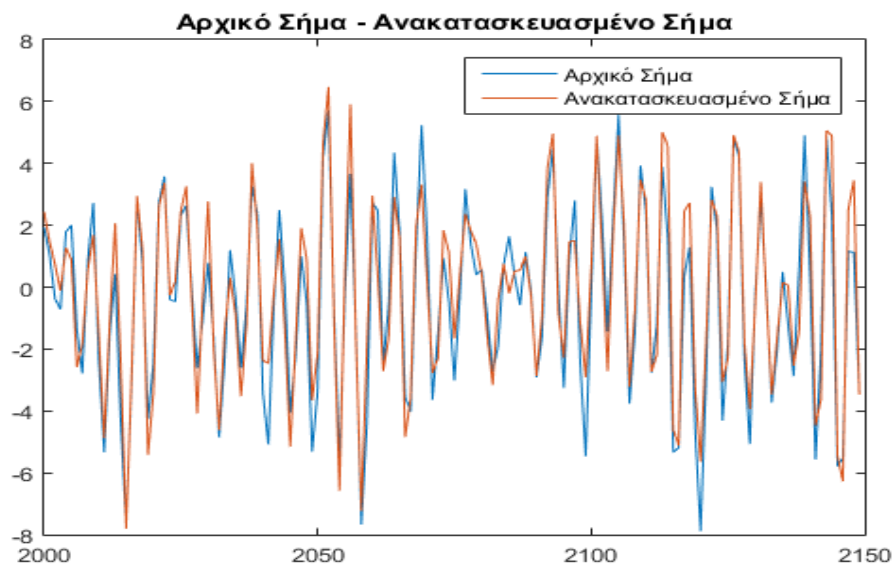
Είναι εμφανές λοιπόν μέσω των γραφημάτων πως το πλήθος των bits κβάντισης αποτελεί τον καθοριστικό παράγοντα για την απόδοση του συνολικού συστήματος DPCM. Σε αντίθεση με το πλήθος των προηγούμενων τιμών p , το οποίο επηρεάζει ελάχιστα την ανακατασκευή του σήματος (εφόσον $p > 4$), παρουσιάζοντας ωστόσο πάντα την καλύτερη δυνατή πρόβλεψη για $p = 4$.

Αναλυτικότερα, παρατηρείται ότι χρησιμοποιώντας μικρό πλήθος bits κβάντισης ($\text{bits} = 1, 2$), το σύστημα DPCM προσφέρει μία αξιόλογη ανακατασκευή του σήματος. Ωστόσο παρατηρείται πως στην περίπτωση αυτή δεν καταφέρνει να προσεγγίσει αρκετά καλά το αρχικό σήμα, ειδικά όταν αυτό παρουσιάζει απότομες μεταβολές. Ωστόσο στην περίπτωση που χρησιμοποιούνται 3 bits κβάντισης, παρατηρείται ότι η ανακατασκευή του σήματος στο δέκτη προσεγγίζει σχεδόν ακριβώς την αρχική μορφή του σήματος, προσφέροντας μία αρκετά αξιόπιστη μετάδοση του σήματος μεταξύ πομπού και δέκτη.

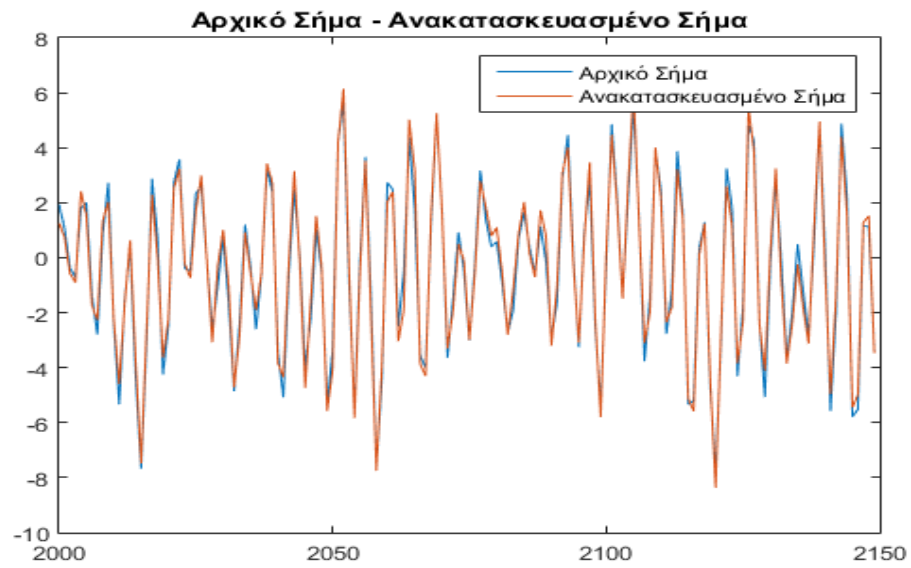
- Για $\text{bits} = 1$ & $p = 4$:



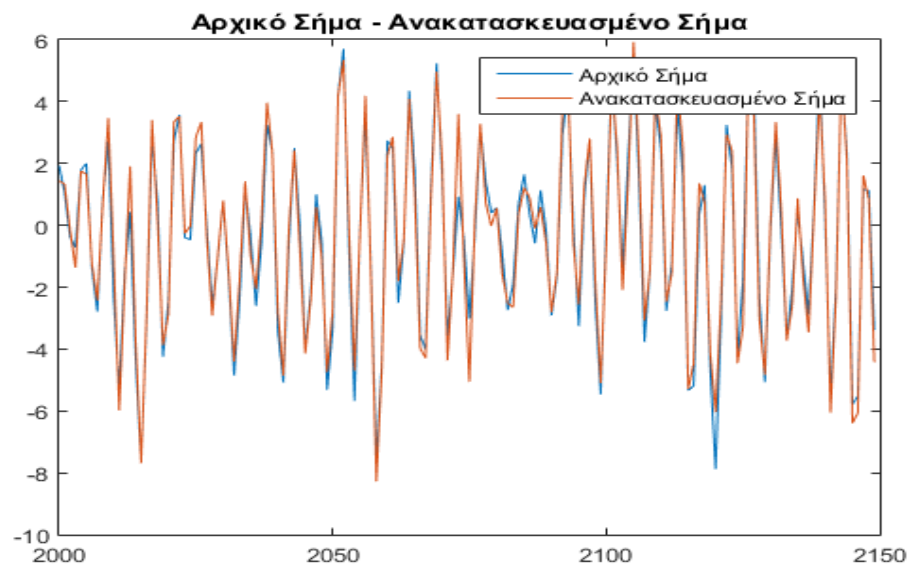
- Για $\text{bits} = 1$ & $p = 8$:



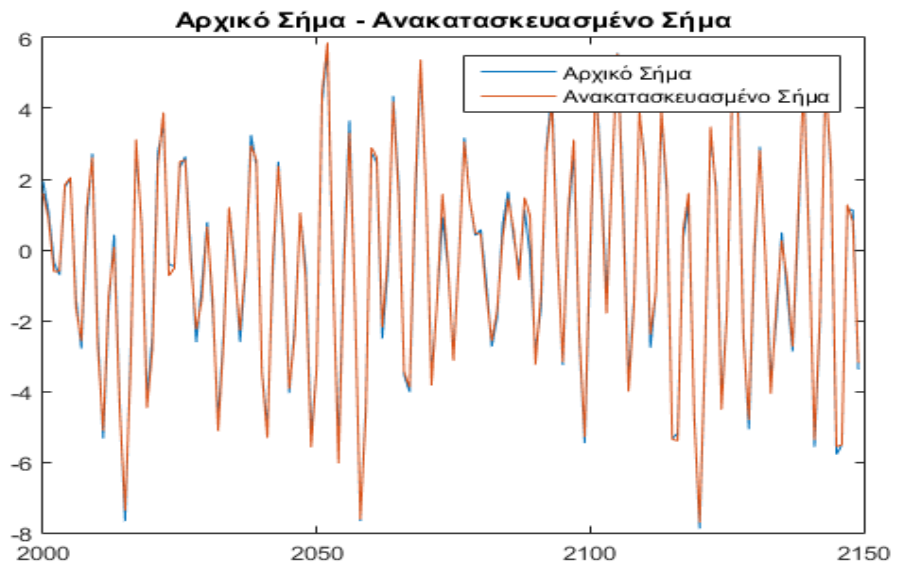
- Για $\text{bits} = 2$ & $p = 4$:



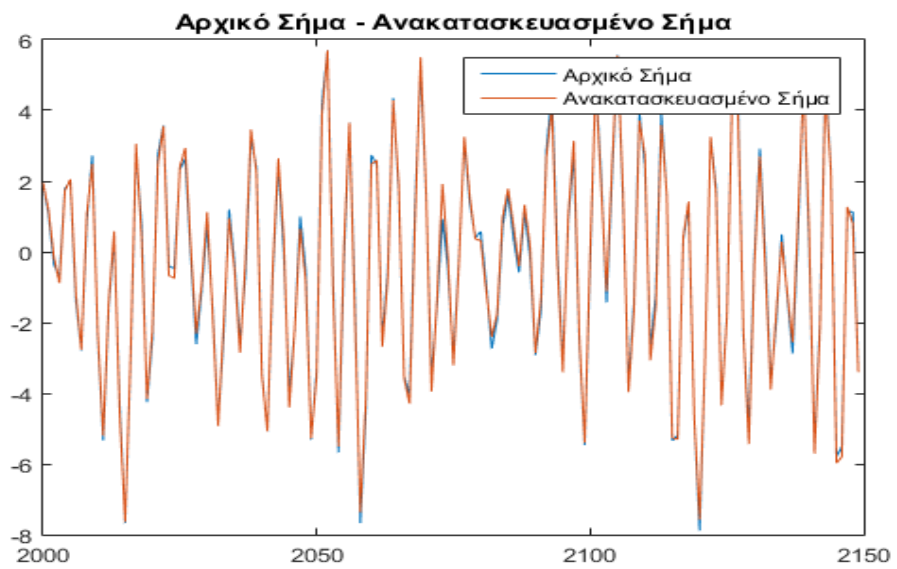
- Για $\text{bits} = 2$ & $p = 8$:



- Για $\text{bits} = 3$ & $p = 4$:



- Για $\text{bits} = 3$ & $p = 8$:



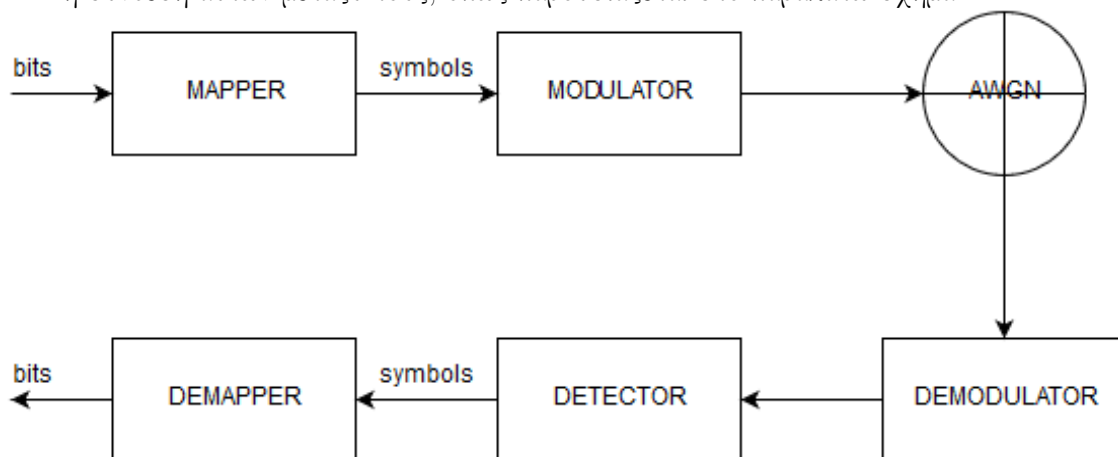
4 Προσομοίωση Ομόδυνου Ζωνοπερατού Συστήματος M-PSK

4.1 Περιγραφή

Το PSK αποτελεί αποκλειστικά ζωνοπερατή διαμόρφωση, εφόσον η πληροφορία μεταφέρεται στη φάση ημιτονοειδών κυματομορφών, της ίδιας συχνότητας και πλάτους (ενέργειας). Όπως κάθε τηλεπικοινωνιακό σύστημα, έτσι και το PSK αποτελείται από τρία βασικά μέρη:

- **Δέκτη:** Αποτελείται από τον *mapper* και τον *modulator*.
- **Κανάλι:** Αποτελείται από ένα *Additive white Gaussian noise (AWGN)* κανάλι.
- **Πομπό:** Αποτελείται από τον *demapper*, τον *decoder* και τον *demodulator*.

Επομένως για την υλοποίηση της προσομοίωσης ενός συστήματος PSK απαιτήθηκε η υλοποίηση του καθενός υποσυστήματος ξεχωριστά⁴, και στη συνέχεια η σύνδεση αυτών μεταξύ τους, όπως παρουσιάζεται στο παρακάτω σχήμα:



⁴Κάθε υποσύστημα υλοποιήθηκε ως μια ξεχωριστή συνάρτηση.

4.1.1 Mapper

Αρχικά υλοποιήθηκε το υποσύστημα του Mapper, ο οποίος αναλαμβάνει την μετατροπή της δυαδικής ακολουθίας του σήματος σε M σύμβολα. Ο Mapper υλοποιήθηκε με τέτοιο τρόπο ώστε να υποστηρίζει δύο τρόπους κωδικοποίησης, την απλή και την Gray. Με αυτόν τον τρόπο ο χρήστης μπορεί μεταβάλλοντας την τιμή της μεταβλητής `gray_encoding` να επιλέξει τον επιθυμητό τρόπο κωδικοποίησης.

Επίσης, ανάλογα του μεγέθους του σήματος εισόδου υπάρχει πιθανότητα να υπάρξουν κάποια bits τα οποία να μην αρκούν ώστε να σχηματίσουν ομάδα των $\log_2(M)$ bits. Για τον λόγο αυτό, κατά την κωδικοποίηση του σήματος, προστίθεται ο απαραίτητος αριθμός από bits μηδενικών στα MSB bits⁵ του σήματος, ώστε όλα τα bits να δημιουργούν αντίστοιχα ομάδα των $\log_2(M)$ bits. Με αυτόν τον τρόπο όλη η δυαδική πληροφορία του σήματος κωδικοποιείται σωστά, δίχως να υπάρξει πιθανώς κάποια απώλεια των τελευταίων δειγμάτων.

4.1.2 Modulator

Στη συνέχεια υλοποιήθηκε το υποσύστημα του διαμορφωτή, ο οποίος αναλαμβάνει να πολλαπλασιάσει το κωδικοποιημένο σήμα με τον ορθογώνιο παλμό και στη συνέχεια να μεταφέρει το σήμα στη ζώνη μετάδοσης, ώστε τελικά να προκύψει το ζωνοπερατό σήμα `s_m`.

⁵Ως MSB bits θεωρήθηκαν τα δεξιότερα bits.

4.1.3 AWGN Κανάλι

Επιπλέον υλοποιήθηκε το υποσύστημα του AWGN καναλιού, μέσω του οποίου διέρχεται το σήμα. Συγκεκριμένα θεωρήθηκε ότι το σήμα διέρχεται μέσω ενός ιδανικού καναλιού. Επομένως εφόσον παράχθηκε ο τυχαίος θόρυβος, με βάση την αντίστοιχη τιμή του SQNR, τότε προστέθηκε στις αντίστοιχες τιμές του διαμορφωμένου σήματος.

4.1.4 Demodulator

Μετά υλοποιήθηκε το υποσύστημα του αποδιαμορφωτή, μέσω του οποίου το σήμα αποδιαμορφώνεται σε ένα δισδιάστατο διάνυσμα r . Το συγκεκριμένο διάνυσμα αντιστοιχεί στη θέση του ληφθέντος σήματος πάνω στο επίπεδο του αστερισμού M-PSK. Μέσω του συγκεκριμένου επιπέδου θα επιτευχθεί ο προσδιορισμός του αντίστοιχου συμβόλου της κάθε ομάδας των $\log_2(M)$ bits.

4.1.5 Detector

Στη συνέχεια υλοποιήθηκε το υποσύστημα του φωρατή, ο οποίος αξιοποιώντας το επίπεδο αστερισμού του M-PSK και την αντίστοιχη θέση του διανύσματος r πάνω σε αυτό, αποφασίζει σε ποιο σύμβολο βρίσκεται εγγύτερα. Για να επιτευχθεί αυτό εξετάζονται όλες οι αποστάσεις μεταξύ των διανυσμάτων r και s_m , χρησιμοποιώντας την μετρική της ευκλείδειας απόστασης, και εφόσον βρεθεί η μικρότερη απόσταση, τότε το αντίστοιχο διάνυσμα s_m αντιστοιχεί στο σύμβολο που στάλθηκε.

4.1.6 Demapper

Τελευταίο υλοποιήθηκε το υποσύστημα του Demapper, ο οποίος δέχεται ως είσοδο το σύμβολο που έχει ανιχνεύσει ο φωρατής και στη συνέχεια παράγει την αντίστοιχη ομάδα από $\log_2(M)$ bits. Επιπλέον ανάλογα με την αρχική κωδικοποίηση που έχει υποστεί το σήμα, επιλέγεται και ο αντίστοιχα κατάλληλος τρόπος αποκωδικοποίησής του.

4.1.7 Script - M-PSK

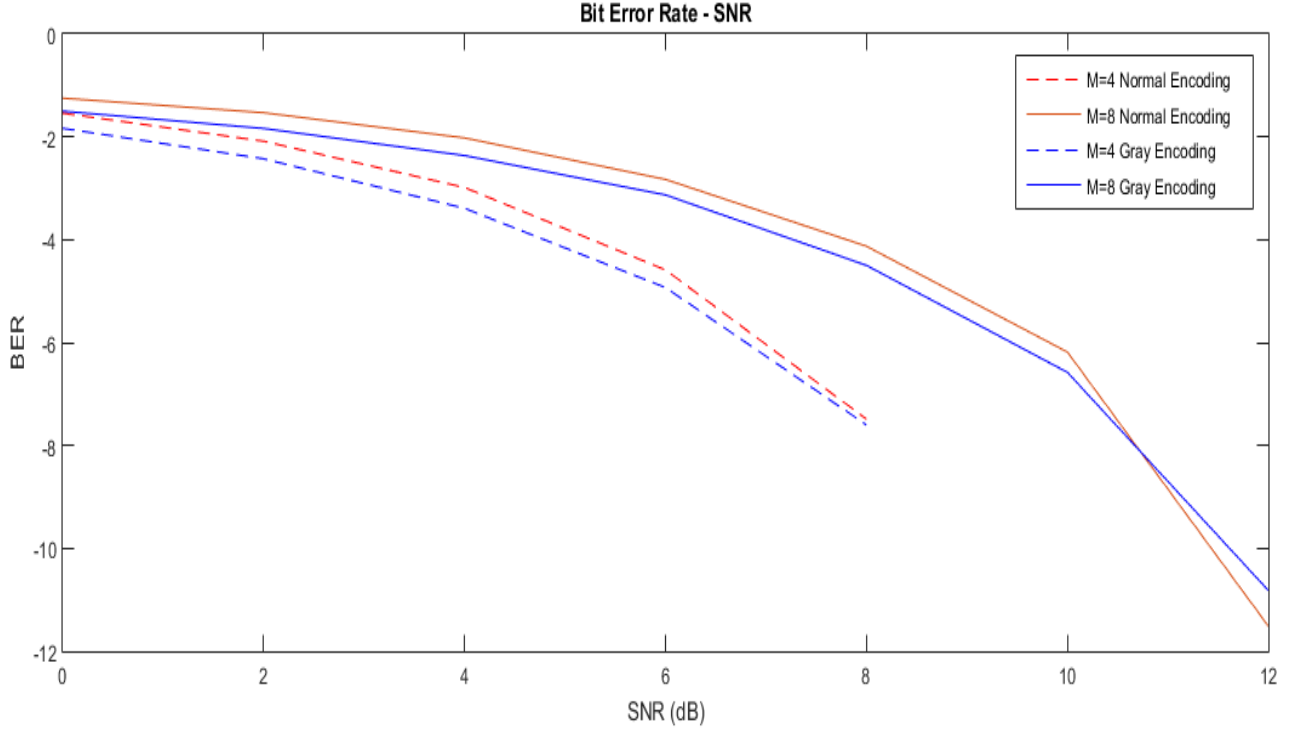
Για την ένωση όλων των υποσυστημάτων που υλοποιήθηκαν δημιουργήθηκε ένα τελευταίο κομμάτι κώδικα, που συμβάλει στην πλήρη και ομαλή λειτουργία της προσομίωσης του συστήματος M-PSK.

Συγκεκριμένα μέσω αυτού δημιουργείται αρχικά ένα τυχαίο δυαδικό σήμα εισόδου, με ισοπίθανες εμφανίσεις των 0 και 1. Στη συνέχεια αναλαμβάνει να χρησιμοποιήσει κατάλληλα τα αντίστοιχα υλοποιημένα υποσυστήματα, ώστε τελικά να προκύψει το αντίστοιχο δυαδικό σήμα εξόδου. Τέλος, αναλαμβάνει να αφαιρέσει τυχόν επιπλέον bits που προστέθηκαν μέσω του Mapper, τα οποία δεν προσφέρουν πια καμία πληροφορία στο σήμα.

4.2 Μετρήσεις BER

Για την αξιολόγηση της απόδοσης του συστήματος M-PSK υπολογίστηκε η πιθανότητα εμφάνισης σφάλματος bit (BER), συγκρίνοντας την τιμή bit που λήφθηκε με αυτήν που στάλθηκε. Για την επίτευξη αξιόπιστων μετρήσεων BER, χρησιμοποιήθηκε ένα αρκετά μεγάλος αριθμός δεδομένων εισόδου, συγκεκριμένα οι μετρήσεις λήφθηκαν σε δεδομένα εισόδου μεγέθους ίσο με 10^5 . Παρακάτω παρουσιάζονται οι αντίστοιχες τιμές BER που υπολογίστηκαν, εξετάζοντας διάφορες τιμές των παραμέτρων M (πλήθος συμβόλων) και SNR (Signal-Noise-Ratio).

<div><div><div><div><div></div><div>SNR</div></div></div><div><div><div># Συμβόλων</div></div></div></div></div>	M=4 Normal Encoding	M=4 Gray Encoding	M=8 Normal Encoding	M=8 Gray Encoding
0 dB	0,21410	0,15950	0,28672	0,22266
2 dB	0,12416	0,08820	0,21612	0,15912
4 dB	0,05045	0,03370	0,13240	0,09397
6 dB	0,01017	0,00722	0,05910	0,04371
8 dB	5,6e-04	5,0e-04	0,01612	0,01111
10 dB	0	0	0,00206	0,00139
12 dB	0	0	1,0e-05	2,0e-05
14 dB	0	0	0	0
16 dB	0	0	0	0



Από τις παραπάνω μετρήσεις παρατηρείται ότι η απόδοση του συστήματος M-PSK παρουσιάζει ιδιαίτερη βελτίωση, όσο αυξάνεται το SNR. Το γεγονός αυτό είναι λογικό, καθώς όσο αυξάνεται η τιμή του SNR τόσο μειώνεται ο θόρυβος του AWGN καναλιού, καθώς μειώνεται αντίστοιχα η διασπορά του θορύβου⁶. Ήταν αναμενόμενο επίσης, εφόσον αυξάνεται το SNR να βελτιώνεται η λήψη του σήματος, καθώς η αύξηση του SNR συνεπάγεται αύξηση της ισχύος του σήματος⁷.

Τέλος, παρατηρείται επίσης ότι η χρήση της κωδικοποίησης Gray, επιτυγχάνει καλύτερη απόδοση σε σχέση με την απλή κωδικοποίηση. Το γεγονός αυτό

⁶ $\sigma^2 = \frac{E_b}{2 \cdot 10^{\frac{SNR}{10}}}$.

⁷ Εξ' ορισμού $SNR = \frac{P_{signal}}{P_{noise}}$.

οφείλεται στον σχεδιασμό του συστήματος PSK και συγκεκριμένα οφείλεται στη γεωμετρική αναπαράστασή του, δηλαδή τους αστερισμούς των σημάτων.

Συγκεκριμένα, κατά τη λήψη του σήματος στον δέκτη, ο φωρατής μπορεί να πάρει εσφαλμένη απόφαση για το σύμβολο που στάλθηκε. Στην περίπτωση όμως αυτή, λόγω της δομής των αστερισμών, το λανθασμένο σύμβολο είναι συνήθως κάποιο από τα γειτονικά του πραγματικού συμβόλου⁸. Επομένως, εφόσον στην κωδικοποίηση Gray κάθε αριθμός διαφέρει με τον επόμενο του και τον προηγούμενό του κατά 1 bit, συνεπάγεται αντίστοιχα ότι μια εσφαλμένη απόφαση του φωρατή οδηγεί σε λάθος της τάξης του ενός μόλις bit. Συνεπώς η κωδικοποίηση Gray υπερτερεί της απλής κωδικοποίησης, όταν ο γεωμετρικός χώρος παρουσιάζει τέτοια δομή, με αποτέλεσμα να εξασφαλίζει μεγαλύτερη αξιοπιστία και απόδοση στο συνολικό τηλεπικοινωνιακό σύστημα PSK.

⁸Η πιθανότητα να είναι παρα-γειτονικά είναι πολύ μικρή.

Αναφορές

- [1] John G.Proakis, Masoud Salehi (2002), *Συστήματα Τηλεπικοινωνιών*, Εκδόση Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών.
- [2] Μπερμπερίδης Κωνσταντίνος, *Ψηφιακές Τηλεπικοινωνίες*, <https://eclass.upatras.gr/courses/CEID1025/>, τελευταία πρόσβαση στις 03/01/2017.

my_quantizer.m

```
1  %Ομοιόμορφος κβαντιστής N δυαδικών ψηφίων
2  %y <- Τρέχων δείγμα
3  %N <- Αριθμός των bits
4  %max/min_value <- Δυναμική περιοχή
5  function y2 = my_quantizer(y, N, min_value, max_value)
6  % Υπολογισμός κέντρων κβαντισμού
7  D = max_value/2^(N-1);
8  %Υπολογισμός των κέντρων κάθε περιοχής
9  centers(1) = max_value-D/2;
10 centers(2^N) = min_value+D/2;
11 for i = 2:(2^N-1)
12     centers(i) = centers(i-1)-D;
13 end
14 %Υπολογισμός περιοχής στην οποία ανήκει το δείγμα
15 for j = 1:length(y)
16     if y(j) <= min_value
17         y2(j) = 2^N;
18     elseif y(j) >= max_value
19         y2(j) = 1;
20     else
21         if y(j) < 0
22             y(j) = max_value + abs(y(j));
23         elseif y(j) >= 0
24             y(j) = max_value - y(j);
25         end
26         y2(j) = floor(y(j)/D)+1;
27     end
28     y2(j) = centers(y2(j));
29 end
30 end
```


DPCM.m

```
1 %Κωδικοποίηση DPCM
2 %Ανάκτηση δεδομένων εισόδου
3 load source.mat
4 %Αριθμός δειγμάτων στη μνήμη
5 p = 8;
6 %Αριθμός bits κβάντισης
7 bits = 3;
8 %Μέγεθος του αρχείου εισόδου
9 N = length(x);
10
11 %Υπολογισμός διανύσματος αυτοσυσχέτισης
12 for i = 1:p
13     sum = 0;
14     for n = p+1:N
15         sum = sum+x(n)*x(n-i);
16     end
17     r(i) = (1/(N-p))*sum;
18 end
19 r = r';
20
21 %Υπολογισμός πίνακα αυτοσυσχέτισης
22 for i = 1:p
23     for j = 1:p
24         sum = 0;
25         for n = p:N
26             sum = sum+x(n-j+1)*x(n-i+1);
27         end
28         R(i,j) = (1/(N-p+1))*sum;
29     end
30 end
```

```

31 %Υπολογισμός των συντελεστών του φίλτρου πρόβλεψης
32 a = inv(R)*r;
33 %Κβάντιση των συντελεστών
34 a_quantum = my_quantizer(a,8,-2,2)';
35
36 %Θεωρούμε ότι τα p πρώτα δείγματα μεταδίδονται
37 %μη κβαντισμένα και χωρίς σφάλματα
38 %Αποθήκευση p αρχικών τιμών στη μνήμη
39 mem(1:p) = x(1:p)';
40 %Αποθήκευση p αρχικών τιμών που θα ληφθούν
41 y2(1:p) = mem(1:p);
42 %Υλοποίηση φίλτρου πρόβλεψης
43 for j = p+1:N
44     sum = 0;
45     for i = 1:p
46         sum = sum + a_quantum(i)*mem(j-i);
47     end
48     %Πρόβλεψη του δείγματος
49     prediction(j) = sum;
50     %Υπολογισμός σφάλματος πρόβλεψης
51     y1(j) = x(j)-prediction(j);
52     %Κβαντοποίηση του σφάλματος πρόβλεψης
53     y1_quantum(j) = my_quantizer(y1(j),bits,-3,3)';
54     %Ανακατασκευή του δείγματος στο δέκτη
55     y2(j) = y1_quantum(j)+ prediction(j);
56     mem(j) = y2(j);
57 end

```

my_mapper.m

```
1 %Υλοποίηση του mapper
2 %Αντιστοίχιση των log2(M) bits εισόδου σε M σύμβολα
3 %input_binary <- Δυαδική ακολουθία σήματος
4 %bits <- Αριθμός bits κάθε ομάδας συμβόλου
5 %gray_encoding <- =1 Για κωδικοποίηση Gray / =0 Για απλή κωδικοποίηση
6 function [input_symbols, input_size, modulo] = my_mapper(
    input_binary, bits, gray_encoding)
7 input_size = length(input_binary);
8
9 %Υπολογισμός του πλήθους των bits που δεν σχηματίζουν ομάδα από log2(M) bits
10 modulo = mod(input_size, bits) ;
11 %Τροποποίηση των τελευταίων bits του σήματος εισόδου, ώστε
12 %όλα τα bits να σχηματίζουν ομάδα από log2(M) bits
13 %=> mod(input_size, bits) == 0
14 if (modulo ~= 0)
15     input_size = input_size - modulo;
16     for i = 1:modulo
17         temp(i) = input_binary(input_size+1);
18     end
19     %Συμπλήρωση με μηδενικά στα MSB του σήματος
20     for i = (modulo+1):bits
21         temp(i) = 0;
22     end
23     input_binary((input_size+1):(input_size+bits)) = temp(:);
24     %Επαναυπολογισμός του μήκους του διανύσματος εισόδου
25     %καθώς αυξήθηκε το μήκος του, προσθέτοντας bits μηδενικών
26     input_size = length(input_binary);
27 end
28
29 %Μετατροπή των δυαδικών log2(M) bits στις αντίστοιχες δυαδικές τιμές
```

```

30 j = 1;
31 for i = 1:bits:input_size
32     input_symbols(j) = bi2de(input_binary(i:i+bits-1)', 'right-
        msb');
33     j = j+1;
34 end
35
36 %Μετατροπή των δεκαδικών τιμών στις αντίστοιχες κωδικοποιήσεις gray
37 if (gray_encoding == 1)
38     input_symbols = bin2gray(input_symbols, 'psk', 2^bits);
39 end
40 end

```

my_modulator.m

```
1 %Υλοποίηση του διαμορφωτή
2 %input_symbols <- Συμβολική ακολουθία του σήματος
3 %M <- Το πλήθος των συμβόλων
4 function [s_m] = my_modulator(input_symbols, M)
5 %Σε κάθε περίοδο φέρουσας κρατάμε 4 δείγματα και
6 %κάθε περίοδο συμβόλου περιλαμβάνει 40 δείγματα
7 t_sumbol = 40;
8 t_c = 4;
9 f_c = 1/t_c;
10 %Ορθογώνιος παλμός
11 g_t = sqrt(2/t_sumbol);
12
13 %Υπολογισμός 2 συνιστωσών για κάθε σύμβολο
14 for m = 0:M-1
15     s(m+1,1) = cos(2*pi*m/M);
16     s(m+1,2) = sin(2*pi*m/M);
17 end
18
19 %Υπολογισμός του ζωνοπερατού σήματος
20 for i = 1:length(input_symbols)
21     for t = 1:t_sumbol
22         s_m(i,t) = s((input_symbols(i)+1),1)*g_t*cos(2*pi*f_c*t) +
23                     s((input_symbols(i)+1),2)*g_t*sin(2*pi*f_c*t);
24     end
25 end
```

my_AWGN.m

```
1 function signal_AWGN = my_AWGN(M, SNR, input_size, s_m)
2 %M <- Το πλήθος των συμβόλων
3 %SNR <- Signal-to-Noise ratio
4 %input_size <- Μέγεθος διανύσματος του σήματος
5 %s_m <- Διαμορφωμένο Σήμα
6 %Ενέργεια ανά bit
7 E_b = 1/log2(M);
8 %Διασπορά θορύβου
9 var = E_b / ((2*10)^(SNR/10));
10 %Υπολογισμός του AWGN
11 noise = sqrt(var)*randn((input_size/log2(M))*40,1);
12 %Υπολογισμός του σήματος που θα παραληφθεί από τον δέκτη
13 %Αρχικό Σήμα + AWGN
14 dim = size(s_m);
15 signal_AWGN = s_m + reshape(noise, dim(1),dim(2));
16 end
17 end
```

my_demodulator.m

```
1  %Υλοποίηση του αποδιαμορφωτή
2  %signal_AWGN <- Το ληφθέν σήμα
3  function r = my_demodulator(signal_AWGN)
4  %Σε κάθε περίοδο φέρουσας κρατάμε 4 δείγματα και
5  %κάθε περίοδο συμβόλου περιλαμβάνει 40 δείγματα
6  t_symbol = 40;
7  t_c = 4;
8  f_c = 1/t_c;
9  %Ορθογώνιος παλμός
10 g_t = sqrt(2/t_symbol);
11
12 %Συσχέτιση της φέρουσας με τον ορθογώνιο παλμό
13 for t = 1:t_symbol
14     y_1(t,1) = g_t*cos(2*pi*f_c*t);
15     y_2(t,1) = g_t*sin(2*pi*f_c*t);
16 end
17
18 %Υπολογισμός του διανύσματος r
19 r_1 = signal_AWGN*y_1;
20 r_2 = signal_AWGN*y_2;
21 r = [r_1,r_2];
22 end
```

my_detector.m

```
1 %Υλοποίηση του φωρατή
2 %r <- Διάνυσμα θέσης του ληφθέντος σήματος στο επίπεδο
3 %M <- Το πλήθος των συμβόλων
4 function output_symbols = my_detector(r, M)
5 %Υπολογισμός των 2 συνιστωσών του κάθε συμβόλου
6 for m = 0:M-1
7     s_m(m+1, 1) = cos((2*pi*m)/M);
8     s_m(m+1, 2) = sin((2*pi*m)/M);
9 end
10
11 %Εύρεση του συμβόλου που στάλθηκε
12 %Υπολογισμός της μικρότερης απόστασης μεταξύ των διανυσμάτων r & s_m
13 %με τη χρήση της ευκλείδειας απόστασης
14 for i = 1:length(r)
15     %Αρχικοποίηση με τη μέγιστη τιμή
16     min = realmax;
17     %Αρχικοποίηση με την μικρότερη θέση
18     pos = 1;
19     for j = 0:M-1
20         euclidean_dist(j+1, 1) = sqrt((r(i, 1)-s_m(j+1, 1))^2+(r(i
21             , 2)-s_m(j+1, 2))^2);
22         if (euclidean_dist(j+1, 1) < min)
23             min = euclidean_dist(j+1, 1);
24             pos = j;
25         end
26     end
27     output_symbols(i, 1) = pos;
28 end
```


my_demapper.m

```
1 %Υλοποίηση του demapper
2 %output_symbols <- Συμβολική ακολουθία του ληφθέν σήματος
3 %input_size <- Μέγεθος διανύσματος του αρχικού σήματος
4 %bits <- Αριθμός bits κάθε ομάδας συμβόλου
5 %gray_encoding <- =1 Για κωδικοποίηση Gray / =0 Για απλή κωδικοποίηση
6 function [output_binary] = my_demapper(output_symbols, input_size,
    bits, gray_encoding)
7
8 %Μετατροπή των δεκαδικών τιμών στα αντίστοιχα δυαδικά log2(M) bits
9 output_binary(1:input_size) = reshape(de2bi(output_symbols(:)', '
    right-msb'), input_size, 1);
10
11 %Μετατροπή των δεκαδικών τιμών στα αντίστοιχα δυαδικά log2(M) bits (Gray)
12 %και μετατροπή των κωδικοποιήσεων Gray στις αντίστοιχες δυαδικές
13 if (gray_encoding == 1)
14     output_gray = gray2bin(output_symbols, 'psk', 2^bits);
15     output_binary(1:input_size) = reshape(de2bi(output_gray(:)', '
        right-msb'), input_size, 1);
16 end
17 end
```

M_PSK.m

```
1 %Προσομοίωση Ομόδυνου Ζωνοπερατού Συστήματος M_PSK
2 %gray_encoding <- =1 Για κωδικοποίηση Gray / =0 Για απλή κωδικοποίηση
3 gray_encoding = 0;
4 %input_size <- Μέγεθος διανύσματος του σήματος
5 input_size= 10^4;
6
7 %Αλφάβητο σήματος εισόδου
8 alphabet = [0 1];
9 %Δημιουργία σήματος με ισοπίθανη εμφάνιση συμβόλων αλφαβήτου
10 input_binary = randsrc(input_size , 1, alphabet);
11 for M = 4:4:8
12
13     %Ομάδα απο m bits
14     bits = log2(M);
15     for SNR = 0:2:16
16         %Υποσύστημα Mapper
17         [input_symbols, input_size, modulo] = my_mapper(
18             input_binary, bits, gray_encoding);
19         %Υποσύστημα Modulator
20         [s_m] = my_modulator(input_symbols, M);
21         %Υποσύστημα Καναλιού AWGN
22         [signal_AWGN] = my_AWGN(M, SNR, input_size, s_m);
23         %Υποσύστημα Demodulator
24         [r] = my_demodulator(signal_AWGN);
25         %Υποσύστημα Detector
26         [output_symbols] = my_detector(r, M);
27         %Υποσύστημα Demapper
28         [output_binary] = my_demapper(output_symbols, input_size,
29             bits, gray_encoding);
```

```

29      %Αφαίρεση των επιπλέον bits μηδενικών που προστέθηκαν
30      if (modulo ~=0)
31          output_binary(length(input_binary)+1:input_size) = [];
32      end
33      output_binary = output_binary';
34
35      %Υπολογισμός BER
36      bits_error = 0;
37      for i = 1:length(input_binary)
38          if (input_binary(i) ~= output_binary(i))
39              bits_error = bits_error + 1;
40          end
41      end
42      PSK_BER(M/4, SNR/2+1) = bits_error/length(input_binary);
43  end
44 end

```