

ΕΣΠΑ 2007-2013, Ε.Π. «ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ»
ΔΡΑΣΗ «ΑΡΙΣΤΕΙΑ II»

«Σύμφωνα Ασύρματα Οπτικά Συστήματα ως
Δομικά Στοιχεία για την Υλοποίηση Δικτύων
Οικίας και Επαγγελματικών Χώρων Επόμενης
Γενιάς, (COWS)»

Κωδικός Έργου : 3525

COWS

Παραδοτέο:	Π2.1
Τίτλος:	Σχήματα Κωδικοποίησης και Διαμόρφωσης Οπτικών Ασύρματων Συστημάτων Σύμφωνα Φώρασης
Ημερομηνία:	30/4/2015
Επιμέρους Δράση:	Π.Ε. 2
Υπεύθυνοι Σύνταξης:	Θωμάς Καμαλάκης, Βασίλειος Δαλάκας, Παναγιώτης Κανάκης, Κατερίνα Μαργαρίτη, Παναγιώτης Ριζομυλιώτης
Είδος:	Τεχνική Αναφορά



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

Περιεχόμενα	2
Ευρετήριο Εικόνων	3
1. Σκοπός του παραδοτέου	5
2. Αρχιτεκτονική του Συστήματος	5
2.1 Συνοπτικό Διάγραμμα	5
2.2 Περιγραφή του οπτικού συστήματος.....	6
3. Σχήματα Διαμόρφωσης.....	10
3.1 Διαμόρφωση με κλείδωμα μεταλλαγής φάσης (Phase Shift Keying – PSK)	10
3.2 M-DPSK.....	11
3.3 M-QAM.....	12
3.4 M-OOK.....	12
4. Κωδικοποίηση	12
4.1 Είδη κωδικοποίησης.....	12
4.2 Σύμπλεξη (Interleaving)	13
5. Προτεινόμενα Σχήματα Κωδικοποίησης	13
5.1 Συνελκτικοί κώδικες.....	13
5.2 BCH κώδικες	14
5.3 LDPC κώδικες	15
6. Εκτίμηση της Φάσης.....	16
6.1 Επίδραση της μεταβολής της φάσης	16
6.2 Μέθοδοι εκτίμησης φάσης.....	18
6.3 Εκτίμηση φάσης με PLL.....	18
6.4 Ψηφιακή εκτίμηση φάσης.....	19
7. Αποτελέσματα	23
7.1 Επιδόσεις Συστήματος Χωρίς Κωδικοποίηση	24
7.2 Συνελκτικοί κώδικες (CC).....	26
7.3 Υβριδικοί κώδικες (LDPC+BCH).....	29
8. Συμπεράσματα.....	31
Αναφορές	33
Παράρτημα.....	34

Ευρετήριο Εικόνων

Εικόνα 1: Το σύμφωνο οπτικό σύστημα που θεωρούμε στα πλαίσια του παρόντος παραδοτέου.	6
Εικόνα 2: Η χαρακτηριστική του οπτικού διαμορφωτή.....	7
Εικόνα 3: (a) Ο τεχνητά παραμορφωμένος αστερισμός τάσεων που αντιστοιχεί στο 16-PSK και (b) ο αστερισμός των συμβόλων που τελικά αποτυπώνεται στο πεδίο E_{out}	8
Εικόνα 4: (a) Ο τεχνητά παραμορφωμένος αστερισμός τάσεων που αντιστοιχεί στο 16-QAM και (b) ο αστερισμός των συμβόλων που τελικά αποτυπώνεται στο πεδίο E_{out}	8
Εικόνα 5: Οπτική ισχύς λήψης στο επίπεδο του δέκτη.....	9
Εικόνα 6: Σχήματα διαμόρφωσης PSK. Από αριστερά προς τα δεξιά BPSK, QPSK, 8-PSK. (Πηγή: Wikipedia).....	11
Εικόνα 7: DBPSK και DQPSK σήματα. Το πάνω είναι το DBPSK σήμα. Τα ανεξάρτητα bits από το DBPSK σήμα μπορούν να συνδυαστούν σε ζεύγη προκειμένου να προκύψει το DQPSK σήμα, με αλλαγές στη φάση κάθε $T_s = 2T_b$. (Πηγή: Wikipedia).....	11
Εικόνα 8: Συνελκτικός κωδικοποιητής με ρυθμό 2/3.....	14
Εικόνα 9: Μορφή των δεδομένων πριν την σύμπλεξη ($n_{ldpc} = 64\ 800$ bits for normal FECFRAME, $n_{ldpc} = 16\ 200$ bits for short FECFRAME).....	16
Εικόνα 10: Αρχικός και τελικός αστερισμός συμβόλων για την περίπτωση ενός PSK σήματος θεωρώντας την μετάδοση 500 συμβόλων.....	17
Εικόνα 11: Αρχικός και τελικός αστερισμός συμβόλων για την περίπτωση ενός PSK σήματος θεωρώντας την μετάδοση 5000 συμβόλων.....	17
Εικόνα 12: Παράδειγμα εξέλιξης φάσης στην περίπτωση που μεταδίδουμε 5000 και 50000 δείγματα.	17
Εικόνα 13: Η αρχή λειτουργίας του PLL.....	18
Εικόνα 14: Εκτίμηση της φάσης με ψηφιακή λογική.....	20
Εικόνα 15: Πολυπλεξία των συμβόλων.....	22
Εικόνα 16: Επιδόσεις QPSK χρησιμοποιώντας εκτίμηση φάσης για δύο διαφορετικά εύρη γραμμής πομπού και τοπικού ταλαντωτή.....	22
Εικόνα 17: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=2$	24
Εικόνα 18: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=4$	24
Εικόνα 19: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=8$	25
Εικόνα 20: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=16$	25
Εικόνα 21: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=32$	25
Εικόνα 22: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=64$	26
Εικόνα 23: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 1/2, $M=8$	26
Εικόνα 24: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 1/2, $M=16$	27
Εικόνα 25: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 1/2, $M=32$	27
Εικόνα 26: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 1/2, $M=64$	27
Εικόνα 27: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=8$	28
Εικόνα 28: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=16$	28
Εικόνα 29: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=32$	28

Εικόνα 30: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=64$	29
Εικόνα 31: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=8$	29
Εικόνα 32: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=16$	30
Εικόνα 33: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=32$	30
Εικόνα 34: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=64$	30
Εικόνα 35: Σύγκριση διαφόρων μεθόδων διαμόρφωσης για πιθανότητα σφάλματος 10^{-3}	31

1. Σκοπός του παραδοτέου

Στο παρόν παραδοτέο θα παρουσιάσουμε τα κύρια σχήματα κωδικοποίησης και διαμόρφωσης για οπτικά ασύρματα τηλεπικοινωνιακά συστήματα όταν χρησιμοποιείται σύμφωνη φάραση. Θα αναλύσουμε την αρχιτεκτονική της ζεύξης όπως διαμορφώνεται στο φυσικό επίπεδο, τα επιμέρους στοιχεία και τα αντίστοιχα μοντέλα που χρησιμοποιούνται ώστε να ληφθούν υπόψη καθώς επίσης και τον τρόπο υπολογισμού των επιδόσεων του συστήματος. Η απευθείας διόρθωση σφαλμάτων (forward acting error correction, FEC) χρησιμοποιείται για να βελτιωθεί η χωρητικότητα ενός διαύλου/καναλιού επικοινωνίας προσθέτοντας πλεονασμό στα πηγαία δεδομένα με στόχο “to achieve the Shannon bound”.

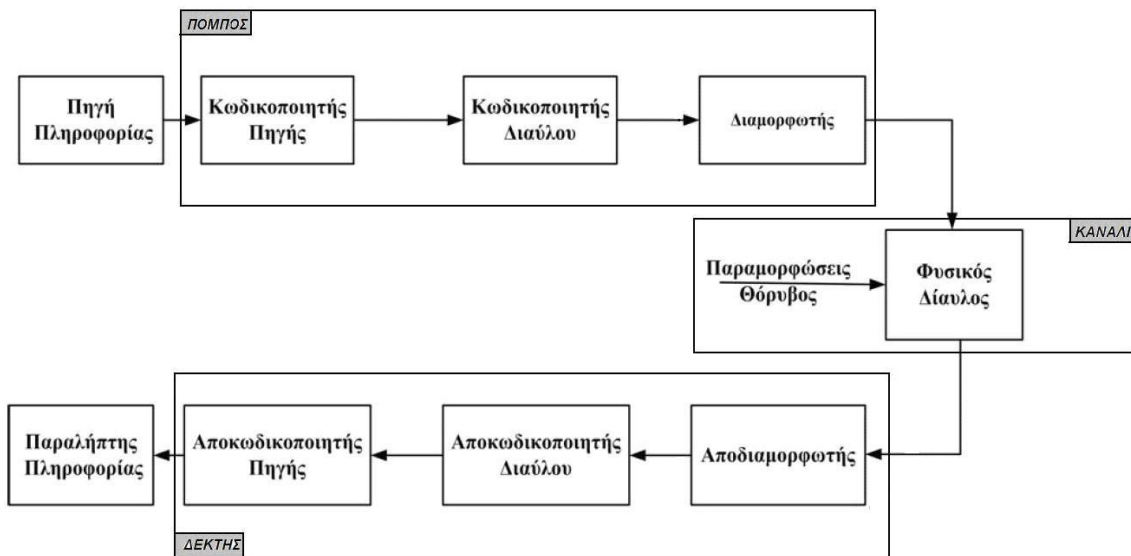
2. Αρχιτεκτονική του Συστήματος

2.1 Συνοπτικό Διάγραμμα

Η κλασική δομή ενός τηλεπικοινωνιακού συστήματος περιλαμβάνει τα εξής επιμέρους στοιχεία:

- Μια πηγή πληροφορίας από την οποία δημιουργείται η μεταβιβαζόμενη πληροφορία.
- Ένα δίαυλο/κανάλι επικοινωνίας μέσω του οποίου μεταφέρεται η πληροφορία.
- Έναν ή περισσότερους προορισμούς στους οποίους καταλήγει.

Νέες απαιτήσεις, όπως αυτές της πολύ υψηλής ταχύτητας μετάδοσης ή της ασφάλειας των αποστελλόμενων δεδομένων, υπαγορεύουν την εισαγωγή επιπλέον στοιχείων σε ένα σύγχρονο τηλεπικοινωνιακό σύστημα. Στο παρακάτω σχήμα απεικονίζεται μία απλοποιημένη μορφή της αρχιτεκτονικής της ζεύξης όταν περιλαμβάνεται κωδικοποίηση.



Σχήμα 1: Συνοπτικό διάγραμμα ενός ασύρματου τηλεπικοινωνιακού δικτύου όπου αναλύονται οι δυνατότητες κωδικοποίησης

Σκοπός μας είναι να καθορίσουμε τις επιδόσεις του δέκτη αν χρησιμοποιούμε σύγχρονες τεχνικές κωδικοποίησης όπως αυτές που εφαρμόζονται σε άλλα σύγχρονα συστήματα. Για το σκοπό αυτό υιοθετήσαμε στα πλαίσια αυτής της έρευνας ορισμένες από τις τεχνικές που περιλαμβάνονται στο DVB-S2X [8] ένα πρότυπο που δημοσιεύτηκε το 2014. Οι τεχνικές αυτές προσαρμόστηκαν και δοκιμάστηκαν στο πλαίσιο των οπτικών ασύρματων επικοινωνιών με τεχνικές σύμφωνης φάρασης. Για το σκοπό αυτό θα πρέπει να ξέρουμε μία σειρά από παραμέτρους όπως:

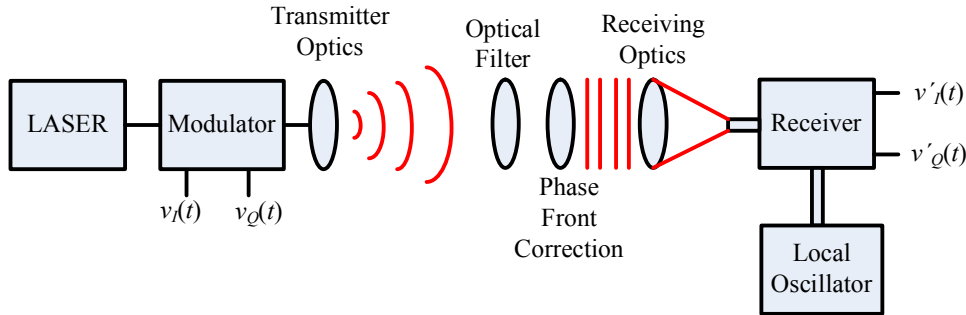
- Το είδος της κωδικοποίησης
- Τα επιμέρους στοιχεία κωδικοποίησης

- Το ρυθμό κωδικοποίησης

Κάποια από τα παραπάνω χαρακτηριστικά είναι σχετικά εύκολο να προσδιορισθούν όπως η θέση του πομπού και του δέκτη, ωστόσο ο προσδιορισμός κάποιων άλλων δεν είναι ιδιαίτερη απλή υπόθεση. Για παράδειγμα, ενώ η απόκριση του καναλιού όταν χρησιμοποιούμε διαμόρφωση πλάτους και άμεση ανίχνευση (intensity modulation / direct detection - IM/DD) η απόκριση του καναλιού έχει μελετηθεί εκτενώς [2] το σύμφωνο ασύρματο οπτικό κανάλι δεν έχει διερευνηθεί σε περιβάλλον εσωτερικού χώρου. Ωστόσο θα πρέπει να λάβουμε υπόψη μας ότι για να λάβει χώρα η μίξη του σήματος του τοπικού ταλαντωτή και του σήματος θα πρέπει τα δύο κύματα να διαθέτουν μεγάλο βαθμό συμφωνίας φάσης. Επομένως όπως θα δούμε και στο Π2.2 η ασύμφωνη ακτινοβολία που μπορεί να οφείλεται τόσο στον περιβάλλοντα φωτισμό όσο και σε διάχυτες ανακλάσεις του κύματος που μεταφέρει την πληροφορία στις επιφάνειες του δωματίου δεν μπορεί να αναμιχθεί με τον τοπικό ταλαντωτή επειδή έχει απωλεστεί η σύμφωνη φύση του φωτός. Αυτό έχει ως αποτέλεσμα το οπτικό κανάλι να διαθέτει μεγάλο εύρος ζώνης σε αντίθεση με την περίπτωση των IM/DD συστημάτων όπου το εύρος ζώνης περιορίζεται από το φαινόμενο της πολυδιόδευσης.

2.2 Περιγραφή του οπτικού συστήματος

Το οπτικό σύστημα που θα μελετήσουμε στον παρόν παραδοτέο είναι πανομοιότυπο με αυτό που αναλύσαμε στο Π1.1 [3] και στο Π1.2 [4] και αποτυπώνεται στο παρακάτω σχήμα:



Εικόνα 1: Το σύμφωνο οπτικό σύστημα που θεωρούμε στα πλαίσια του παρόντος παραδοτέου.

Όπως βλέπουμε στην Εικόνα 1 πρόκειται για ένα ασύρματο οπτικό σύστημα που χρησιμοποιεί σύμφωνη ομόδμη σύμφωνη φώραση. Το οπτικό σήμα ενός laser διαμορφώνεται από τον οπτικό διαμορφωτή ο οποίος αποτυπώνει τις τάσεις $v_I(t)$ και $v_Q(t)$ στο οπτικό φέρον. Οι τάσεις $v_I(t)$ και $v_Q(t)$ περιέχουν τις κυματομορφές της διαμόρφωσης για τις I και Q συνιστώσες αντίστοιχα και δίνονται από τις σχέσεις:

$$v_I(t) = \sum_{m=-\infty}^{+\infty} V_m p(t - mT_s) \quad (1)$$

$$v_Q(t) = \sum_{m=-\infty}^{+\infty} U_m p(t - mT_s) \quad (2)$$

όπου $p(t)$ είναι ένας κατάλληλος παλμός τάσης (π.χ. τύπου ανυψωμένου συνημίτονου), T_s είναι η διάρκεια του συμβόλου, V_m και U_m είναι κατάλληλα πλάτη τάσης τα οποία χρησιμοποιούνται για να αποτυπώσουν τον αστερισμό της επιθυμητής διαμόρφωσης πάνω στο πεδίο E_{out} σύμφωνο και με την σχέση

$$E_{\text{out}} = E_I + jE_Q = \frac{\sqrt{P_T}}{2} \cos\left(\pi \frac{v_I}{V_\pi}\right) + j \frac{\sqrt{P_T}}{2} \cos\left(\pi \frac{v_Q}{V_\pi}\right) \quad (3)$$

Αντικαθιστώντας τις (1)-(2) στην (3) και κάνοντας δειγματοληψία τις χρονικές στιγμές $t_m = mT_s$ θεωρώντας ότι $p(mT_s) = 1$ θα έχουμε,

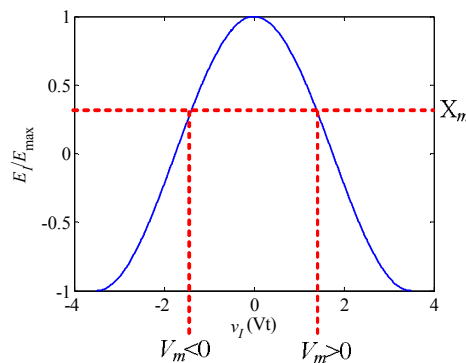
$$E_m = E_{\text{out}}(mT_s) = \frac{\sqrt{P_T}}{2} (X_m + jY_m) = \frac{\sqrt{P_T}}{2} \cos\left(\pi \frac{V_m}{V_\pi}\right) + j \frac{\sqrt{P_T}}{2} \cos\left(\pi \frac{U_m}{V_\pi}\right) \quad (4)$$

όπου με X_m και Y_m έχουμε συμβολίσει τα πλάτη του οπτικού πεδίου που αντιστοιχούν στην I και Q συνιστώσα,

$$X_m = \cos\left(\pi \frac{V_m}{V_\pi}\right) \quad (5)$$

$$Y_m = \cos\left(\pi \frac{U_m}{V_\pi}\right) \quad (6)$$

όπου P_T είναι η οπτική ισχύς εκπομπής και V_π είναι η χαρακτηριστική τάση του οπτικού διαμορφωτή. Όπως βλέπουμε από τις (5)-(6), ο διαμορφωτής είναι μη-γραμμικός δεδομένου ότι οι συνιστώσες του πεδίου E_I και E_Q δεν είναι απευθείας ανάλογες των τάσεων v_I και v_Q . Στην παρακάτω εικόνα βλέπουμε την χαρακτηριστική του διαμορφωτή από την οποία καταλαβαίνουμε ότι η απόσταση των συμβόλων τάσης $V_m + jU_m$ είναι διαφορετική από την απόσταση των συμβόλων που τελικά αποτυπώνονται στο οπτικό πεδίο $X_m + jY_m$. Αυτό μπορεί να είναι σημαντικό πρόβλημα στην περίπτωση συστημάτων διαμόρφωσης υψηλής τάξης όπως το M-QAM για $M=16$ ή 64 όπου επιθυμούμε η απόσταση των διαδοχικών γειτονικών συμβόλων όπως αυτά αποτυπώνονται στο οπτικό πεδίο E_{out} να είναι σταθερή (διότι αν δεν είναι σταθερή ορισμένα σύμβολα θα είναι πιο επιρρεπή στην επίδραση του προσθετικού θορύβου από κάποια άλλα).



Εικόνα 2: Η χαρακτηριστική του οπτικού διαμορφωτή.

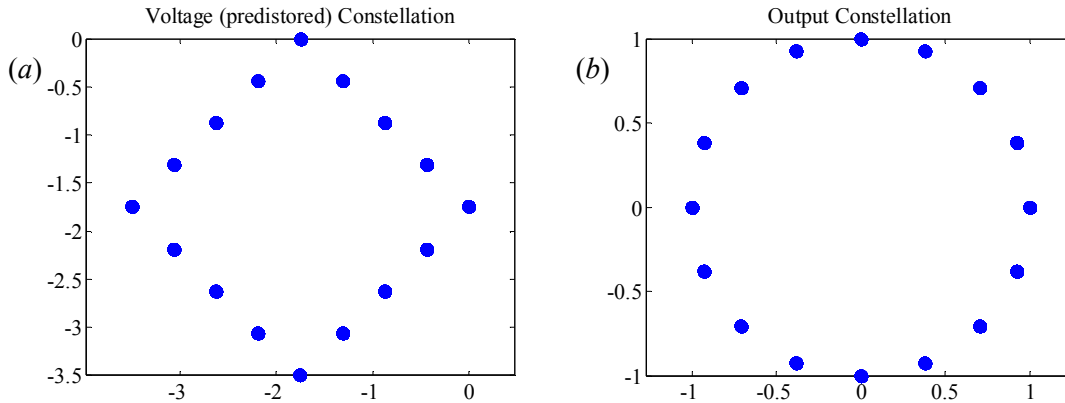
Στα πλαίσια του προγράμματος COWS θεωρήσαμε ένα απλό σχήμα τεχνητής παραμόρφωσης στον πομπό (predistortion) κατά το οποίο επιλέγουμε τα πλάτη των τάσεων V_m και U_m έτσι ώστε ο αστερισμός των συμβόλων $X_m + jY_m$ του οπτικού πεδίου να είναι ο επιθυμητός. Είναι εύκολο από τις (5)-(6) να δούμε ότι:

$$V_m = \frac{V_\pi}{\pi} \cos^{-1}(X_m) \quad (7)$$

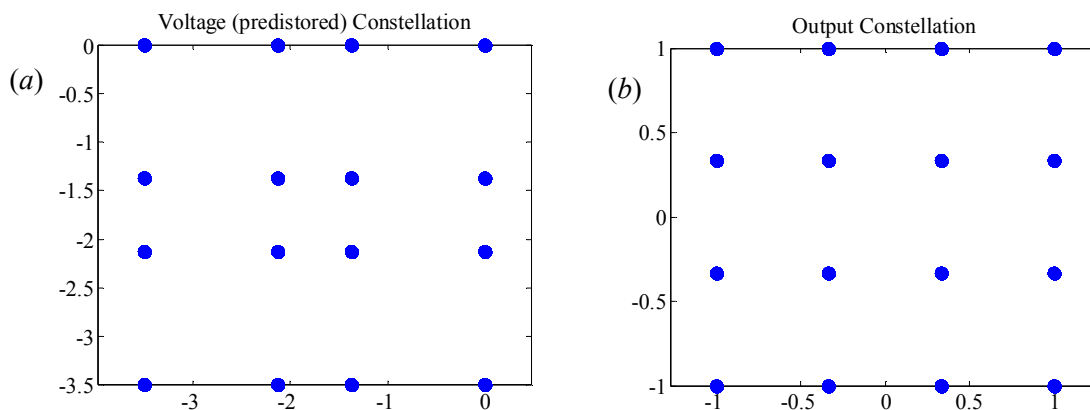
$$U_m = \frac{V_\pi}{\pi} \cos^{-1}(Y_m) \quad (8)$$

όπου με $\cos^{-1}(x)$ εννοούμε την αντίστροφη συνάρτηση του συνημίτονου. Στο σημείο αυτό ίσως να είναι χρήσιμο να επιλέξουμε σε πιο κλάδο την $\cos^{-1}(x)$ επιθυμούμε να επιλέξουμε επειδή δεδομένου ότι $\cos(-x) = \cos(x)$ έπεται ότι υπάρχουν δύο τιμές τις γωνίας $-\pi \leq x < \pi$ που έχουν το ίδιο συνημίτονο. Επιλέγουμε τις αρνητικές τιμές τάσης V_m και U_m που πληρούν τις (7)-(8). Η επιλογή αυτή γίνεται επειδή στις αρνητικές τιμές τάσης ($v_I < 0$) το πεδίο E_{out} παρουσιάζει αύξουσα συμπεριφορά όπως θα περίμενε κανείς από έναν κλασικό διαμορφωτή, σε αντίθεση με τις θετικές τιμές τάσης v_I όπου παρουσιάζει φθίνουσα συμπεριφορά. Φυσικά θα μπορούσαμε εξίσου να επιλέξουμε και τις θετικές τιμές τάσης V_m και U_m αλλά στην περίπτωση αυτή ένα σύμβολο με μεγάλο πλάτος X_m θα αντιστοιχούσε σε ένα πλάτος τάση V_m κοντά στο μηδέν κάτι που δεν συμφωνεί με την διαίσθηση μας και μπορεί να προκαλέσει παρανοήσεις στην ερμηνεία των αποτελεσμάτων. Θα πρέπει ωστόσο να σημειωθεί ότι οι εμπορικά διαθέσιμοι διαμορφωτές μπορεί να παρουσιάζουν διαφορετική συμπεριφορά από αυτή που περιγράφεται στην σχέση (3) εξαιτίας σφαλμάτων φάσης στους κυματοδηγούς των συμβολομέτρων τους, ωστόσο διαθέτουν ειδικούς ακροδέκτες τάσης που μπορούν να χρησιμοποιηθούν ώστε να διορθωθούν τα επιμέρους σφάλματα και να προσεγγίσουμε την συμπεριφορά που περιγράφει η εξίσωση (3).

Στην Εικόνα 3 και στην Εικόνα 4 δείχνουμε τους τεχνητά παραμορφωμένους αστερισμούς της τάσης και τους αντίστοιχους αστερισμούς του οπτικού πεδίου στην περίπτωση του 16-PSK και του 16-QAM αντίστοιχα. Ο εν λόγω τρόπος παραμόρφωσης είναι σχετικά απλός και αντιστοιχεί απλά σε τροποποίηση του πλάτους της τάσης των σημάτων που οδηγούν τον διαμορφωτή. Στην περίπτωση του 16-PSK παρατηρούμε ότι ο αστερισμός των τάσεων έχει σχήμα ρόμβου ενώ στην περίπτωση του 16-QAM παρατηρούμε ότι ο αστερισμός τάσης προκύπτει από τον κλασικό 16-QAM αστερισμό με μία απλή αλλαγή κλίμακας στα κεντρικά σύμβολα.

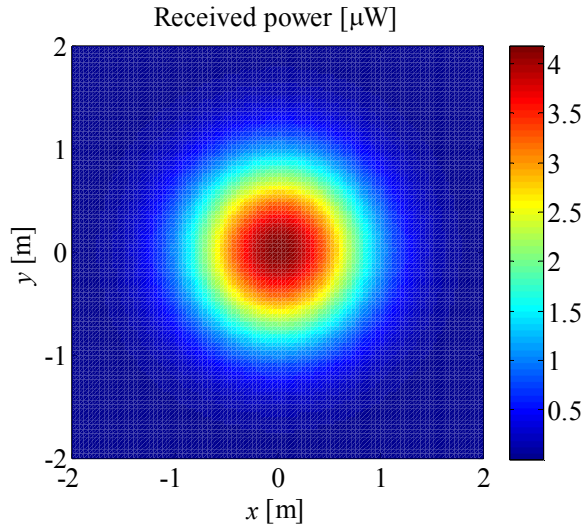


Εικόνα 3: (a) Ο τεχνητά παραμορφωμένος αστερισμός τάσεων που αντιστοιχεί στο 16-PSK και (b) ο αστερισμός των συμβόλων που τελικά αποτυπώνεται στο πεδίο E_{out} .



Εικόνα 4: (a) Ο τεχνητά παραμορφωμένος αστερισμός τάσεων που αντιστοιχεί στο 16-QAM και (b) ο αστερισμός των συμβόλων που τελικά αποτυπώνεται στο πεδίο E_{out} .

Στις υπόλοιπες πτυχές του, το οπτικό σύστημα διέπεται από τις παραμέτρους που αναλύσαμε στο Π1.2 και ο Πίνακας 1 συνοψίζει τις τιμές του. Όπως δείχνει και ο πίνακας θεωρούμε ότι η απόσταση μεταξύ πομπού/δέκτη είναι 3m και τα δύο στοιχεία είναι προσανατολισμένα το ένα προς το άλλο. Ο ρυθμός μετάδοσης είναι της τάξης του 1Gb/s ενώ η ισχύς εκπομπής είναι της τάξης του 1mW. Στην Εικόνα 5 δείχνουμε την κατανομή της οπτικής ισχύος λήψης P_R στο επίπεδο του δέκτη. Παρατηρούμε ότι για τις εν λόγω παραμέτρους, η μέγιστη ισχύς λήψης είναι περίπου 4.1μW ενώ η οπτική ισχύς μειώνεται στο μισό σε μία απόσταση περίπου 0.7m από την θέση που έχουμε θεωρήσει τον δέκτη. Επομένως το εν λόγω σύστημα έχει ικανοποιητική κάλυψη που αντιστοιχεί σε μία κυκλική μικροκυψέλη με ακτίνα περίπου 1m. Θα πρέπει επίσης να σημειωθεί ότι στο συγκεκριμένο σύστημα δεν έχουμε θεωρήσει κάποια εξειδικευμένα οπτικά στοιχεία συστήματα εκπομπής (φακούς, κτλ). Στο σημείο αυτό θα πρέπει να σημειωθεί ότι έχουμε θεωρήσει ένα απλοποιημένο μοντέλο λευκού θερμικού θορύβου επειδή συνήθως χρησιμοποιούμε πολύ λίγα δείγματα ανά σύμβολο επειδή σε μεγάλους ρυθμούς μετάδοσης η μεγαλύτερη δειγματοληψία απαιτεί γρήγορους μετατροπείς από αναλογικό σε ψηφιακό και αντίστροφα. Επομένως αν θεωρήσουμε ένα δείγμα ανά σύμβολο, ο θερμικός θόρυβος μπορεί να θεωρηθεί λευκός. Σε κάθε περίπτωση μπορούμε να χρησιμοποιήσουμε το μοντέλο του έγχρωμου θορύβου όπου απαιτείται σύμφωνα και με τα όσα συζητήσαμε στο Π1.1 και Π1.2.



Εικόνα 5: Οπτική ισχύς λήψης στο επίπεδο του δέκτη.

Πίνακας 1: Παράμετροι του οπτικού συστήματος

Διαρρύθμιση χώρου

Χαρακτηριστικό	Τιμή	Παρατηρήσεις
Μήκος δωματίου	5m	Καθορίζουν την κατανομή του ηλιακού φωτός και του φωτός από τις λάμπες εντός του δωματίου
Πλάτος δωματίου	7.5m	
Ύψος δωματίου	3m	
Θέση πομπού \mathbf{r}_T	[2.5, 3.75 3]	Θεωρείται στο κέντρο του ταβανιού του δωματίου
Θέση δέκτη \mathbf{r}_R	$[x_R, y_R, 0]$	Θεωρείται σε ύψος μισό μέτρο από τον πομπό
Πλήθος παραθύρων	1	Παίζει ρόλο στον υπολογισμό της φασματικής ισχύος του ηλιακού φωτός P_{sun}
Θέση παραθύρου	Στο επίπεδο $y=7.5$	
Προσανατολισμός πομπού \mathbf{n}_T	[0, 0, -1]	Ο πομπός είναι προσανατολισμένος κάθετα προς τα κάτω
Προσανατολισμός πομπού \mathbf{n}_R	[0, 0, 1]	Ο δέκτης είναι προσανατολισμένος κάθετα προς τα πάνω

Συνθήκες Φωτισμού

Χαρακτηριστικό	Τιμή	Παρατηρήσεις
Είδος λαμπτήρων του δωματίου	Αλογόνου, Φωσφορισμού, LED	Καθορίζει σε μεγάλο βαθμό το φάσμα εκπομπής
Φασματική πυκνότητα εκπομπής λαμπτήρων p_{lamp}	Μερικές δεκάδες $mW/nm/m^2$	Εξαρτάται από το είδος του λαμπτήρα
Φασματική πυκνότητα του ηλιακού φωτός p_{sun}	Μερικές εκατοντάδες $mW/nm/m^2$	Εξαρτάται από το μήκος κύματος. Υπάρχει σημαντική εκπομπή και κοντά στο $\lambda=1550nm$
Φασματική πυκνότητα περιβάλλοντος φωτός στο δέκτη p_n	μερικά $mW/nm/m^2$	Μετρίεται πειραματικά ή υπολογίζεται

Παράμετροι πομπού

Χαρακτηριστικό	Τιμή	Παρατηρήσεις
Ισχύς εκπομπής P_T	$\geq 1mW$	Δεν μπορεί να ξεπερνά τις μερικές δεκάδες mW στην περίπτωση ημιαγωγικών laser με μικρό εύρος γραμμής. Καθορίζεται έτσι ώστε να

	$\leq 40\text{mW}$	επιτύχουμε το επιθυμητό SNR
Μήκος κύματος	$\lambda=780\text{nm}$ ή $\lambda=1550\text{nm}$	Η επιλογή του μήκους κύματος καθορίζεται από το κόστος, τις επιθυμητές συνθήκες φωτισμού και τα πρότυπα ασφαλείας.
Εύρος γραμμής $\Delta\nu_T$	$\geq 100\text{kHz}$	Θα πρέπει να επιλεγεί πολύ μικρότερο από τον ρυθμό συμβόλων
Διαμόρφωση	Εξωτερική QPSK	Η εξωτερική διαμόρφωση μπορεί να χρησιμοποιηθεί για να υλοποιηθούν σχήματα διαμόρφωσης όπως το QPSK και το DQPSK
Εύρος δέσμης ω_0	$\sim 5\mu\text{m}$	Καθορίζεται από την οπτική ίνα στην έξοδο του διαμορφωτή αν αυτός είναι fiber-coupled.
Πλήθος φακών πομπού	2	
Απόσταση μεταξύ φακών d_i	$\sim 13\text{mm}$	
Διάμετρος φακών	$\sim 20\text{mm}$	Ο καθορισμός των παραμέτρων του οπτικού συστήματος του δέκτη στηρίζεται σε όσα θα αναλυθούν και στο Π2.2
Εστιακή απόσταση φακών	$f=-1\text{cm}$	

Παράμετροι δέκτη

Χαρακτηριστικό	Τιμή	Παρατηρήσεις
Ισχύς τοπικού ταλαντωτή P_{LO}	$\geq 1\text{mW}$ $\leq 40\text{mW}$	Δεν μπορεί να ξεπερνά τις μερικές δεκάδες mW στην περίπτωση ημιαγωγικών laser με μικρό εύρος γραμμής. Καθορίζεται έτσι ώστε να επιτύχουμε το επιθυμητό SNR
Εύρος γραμμής $\Delta\nu_R$	$\geq 100\text{kHz}$	Θα πρέπει να επιλεγεί πολύ μικρότερο από τον ρυθμό συμβόλων
Αποκρισιμότητα δέκτη R	$\cong 0.6\text{A/W}$ ($\lambda=780\text{nm}$) $\cong 1\text{A/W}$ ($\lambda=1550\text{nm}$)	Καθορίζεται από τα χαρακτηριστικά των φωτοдиодων του δέκτη
Φασματική πυκνότητα θερμικού θορύβου S_{TH}	$\sim 10^{-23}\text{A}^2/\text{Hz}$	Εξαρτάται από τον σχεδιασμό του ενισχυτή όπως είδαμε στο Π1.1
Φασματική πυκνότητα θορύβου βολής S_{SH}	$\sim 10^{-24}\text{A}^2/\text{Hz}$	Εξαρτάται τις συνθήκες φωτισμού όπως δείξαμε στο Π1.1
Εύρος ζώνης δέκτη B	$\sim 1\text{GHz}$	Στην περίπτωση του QPSK αντιστοιχεί σε ένα ρυθμό $\sim 2\text{Gb/s}$
Εύρος ζώνης οπτικού φίλτρου $\Delta\lambda$	$\geq 10\text{nm}$	Στο εμπόριο κυκλοφορούν διηλεκτρικά φίλτρα με σχετικά μικρό κόστος και φασματικά εύρος μερικών nm
Ελάχιστο ηηλικό σήμα-προς-θόρυβο ανά bit SNR_b	8dB	Προκύπτει από την ευαισθησία του θορύβου φάσης όπως είδαμε στο Π2.1
Ελάχιστο ηηλικό σήμα-προς-θόρυβο ανά σύμβολο SNR	11dB	$\text{SNR}=\log_2 M \times \text{SNR}_b$ όπου M είναι η τάξη διαμόρφωσης.
Επιφάνεια φακού λήψης A_R	$\sim 75\text{cm}^2$	
Αριθμός F_R του φακού του δέκτη	~ 1.3	Πληροφορίες για τον καθορισμό των παραμέτρων αυτών μπορούν να βρεθούν στο Π2.2
Διάμετρος φακού λήψης D_R	$\sim 10\text{cm}$	
Εστιακή απόσταση f_R	$\sim 13\text{cm}$	

3. Σχήματα Διαμόρφωσης

Ακολούθως θα αναφέρουμε σύντομα τα σημαντικότερα χαρακτηριστικά των σχημάτων διαμόρφωσης που χρησιμοποιήθηκαν στα πλαίσια της έρευνας που παρουσιάζεται στο συγκεκριμένο κεφάλαιο του παραδοτέου βασισμένοι σε όσα αναφέρονται στην βιβλιογραφία [15].

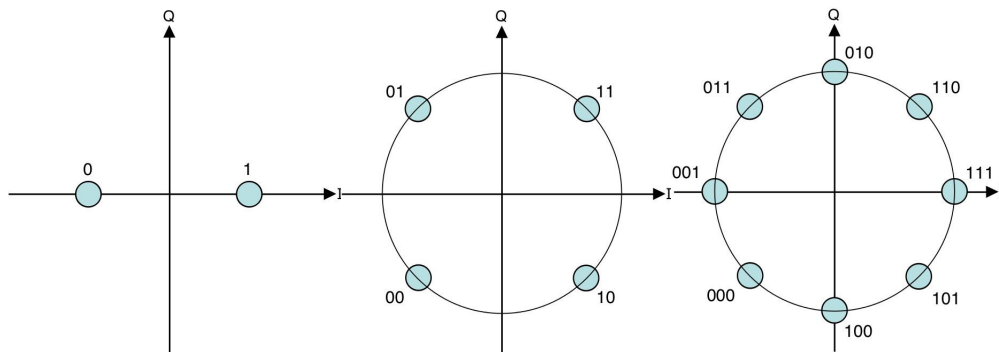
3.1 Διαμόρφωση με κλείδωμα μεταλλαγής φάσης (Phase Shift Keying - PSK)

Η διαμόρφωση με κλείδωμα μεταλλαγής φάσης PSK (Phase-shift Keying) είναι ο τύπος διαμόρφωσης στην οποία η πληροφορία περιέχεται στη στιγμιαία φάση του διαμορφωμένου φέροντος σήματος. Στη μέθοδο αυτή, η φάση του φέροντος σήματος μεταβάλλεται για να απεικονιστεί το bit ενώ η συχνότητα και το πλάτος παραμένουν σταθερά. Για παράδειγμα αν θεωρηθεί ότι η φάση 0° είναι το 0 τότε μεταβάλλοντας τη φάση

κατά 180° είναι το 1. Η φάση μένει σταθερή για κάθε διάστημα μετάδοσης ενός bit. Τα κυριότερα σχήματα PSK είναι

- $M=2$ Binary phase-shift keying (BPSK)
- $M=4$ Quadrature phase-shift keying (QPSK)
- $M=8$ Οκταδικό phase shift keying (8-PSK)

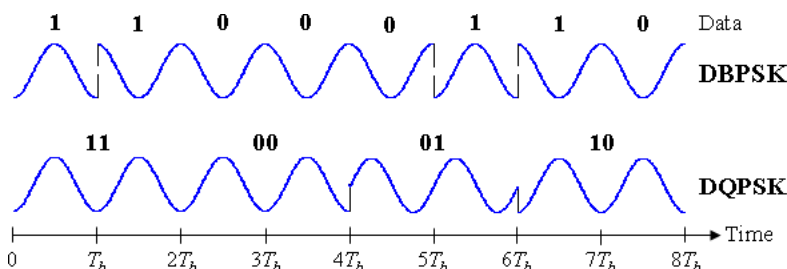
Αν και οποιοσδήποτε αριθμός επιπέδων φάσεων M μπορεί να θεωρητικά να χρησιμοποιηθεί για την υλοποίηση ενός PSK αστερισμού, το 8-PSK στις επικοινωνίες RF είναι το πιο συνηθισμένο καθώς για τα μεγαλύτερης τάξης PSK με περισσότερες φάσεις ο ρυθμός σφάλματος γίνεται τόσο μεγάλος που είναι προτιμότερο να χρησιμοποιηθούν άλλα συστήματα όπως η τετραγωνική διαμόρφωση πλάτους (Quadrature Amplitude Modulation, QAM). Ωστόσο θα πρέπει να σημειωθεί ότι στην περίπτωση των οπτικών συστημάτων με σύμφωνο δέκτη, η διαμόρφωση QPSK ($M=4$) είναι ιδιαίτερα δημοφιλής εξαιτίας της απλότητας υλοποίησης.



Εικόνα 6: Σχήματα διαμόρφωσης PSK. Από αριστερά προς τα δεξιά BPSK, QPSK, 8-PSK. (Πηγή: Wikipedia)

3.2 M-DPSK

Η διαφορική διαμόρφωση μετατόπισης φάσης (DPSK) είναι μια κοινή μορφή διαμόρφωσης φάσης που μεταβιβάζει τα δεδομένα με την αλλαγή της φάσης του φέροντος κύματος. Είναι γνωστή και ως διαφορική κωδικοποίηση. Ένα διακριτικό γνώρισμα της είναι το γεγονός ότι δεν υπάρχουν προκαθορισμένες περιοχές απόφασης, όπου η κάθε περιοχή συνδέεται με συγκεκριμένο σημείο πληροφορίας. Αντί αυτού, η απόφαση στηρίζεται στη διαφορά φάσης μεταξύ διαδοχικά λαμβανόμενων σημάτων. Έτσι ενώ στην περίπτωση του PSK είναι η φάση που καθορίζει την πληροφορία που μεταδίδεται, στο DPSK η πληροφορία κωδικοποιείται στις διαφορές της φάσης. Η δυαδική πληροφορία εισόδου περιέχεται στη διαφορά φάσης μεταξύ δύο διαδοχικών σημάτων παρά στην απόλυτη φάση. Με την DPSK, δεν είναι απαραίτητο να ανακτηθεί η φάση του φέροντος κύματος. Αντίθετα, ένα λαμβανόμενο σύμβολο υστερεί για μια χρονική στιγμή και στη συνέχεια αυτό συγκρίνεται με το επόμενο λαμβανόμενο σύμβολο. Η διαφορά στη φάση μεταξύ των δυο συμβόλων καθορίζει τη λογική κατάσταση των δεδομένων. Για παράδειγμα, σε διαφορικά κωδικοποιημένο BPSK ένα δυαδικό "1" μπορεί να μεταδοθεί με την προσθήκη 180° ως προς την τρέχουσα φάση του σήματος και ένα δυαδικό "0" με την προσθήκη 0° ως προς την τρέχουσα φάση.



Εικόνα 7: DBPSK και DQPSK σήματα. Το πάνω είναι το DBPSK σήμα. Τα ανεξάρτητα bits από το DBPSK σήμα μπορούν να συνδυαστούν σε ζεύγη προκειμένου να προκύψει το DQPSK σήμα, με αλλαγές στη φάση κάθε $T_s = 2T_b$. (Πηγή: Wikipedia)

3.3 M-QAM

Η διαμόρφωση QAM (Quadrature Amplitude Modulation) είναι ουσιαστικά συνδυασμός των τεχνικών διαμόρφωσης PSK (Phase Shift Keying) και ASK (Amplitude Shift Keying). Συγκεκριμένα, η μετάδοση της πληροφορίας γίνεται με τη βοήθεια σήματος διαμορφωμένο κατά πλάτος και κατά φάση. Όταν εκπέμπουμε 2 σήματα διαμορφώνοντάς τα με QAM, το εκπεμπόμενο σήμα θα έχει την ακόλουθη μορφή:

$$u_m(t) = A_m \cos(t) \cos 2\pi f_c t + A_m \sin(t) \sin 2\pi f_c t$$

για $m = 1, 2, \dots, M$

Στις ψηφιακές επικοινωνίες κάνουμε λόγο για M-QAM με το M να παίρνει τιμές των δυνάμεων του 2. Στις δύο πρώτες περιπτώσεις αναφερόμαστε ουσιαστικά στην BPSK και QPSK αντίστοιχα καθώς το πλάτος του σήματος παραμένει σταθερό κατά τη διαμόρφωση και χρησιμοποιείται μόνο η διαφορά φάσης για την μετάδοση της πληροφορίας.

3.4 M-OOK

Η διαμόρφωση on-off keying (OOK) είναι η απλούστερη μορφή της διαμόρφωσης πλάτους ASK για ψηφιακά δεδομένα, όπως η παρουσία ή απουσία ενός φέροντος κύματος. Στην απλούστερη μορφή της, η παρουσία ενός φέροντος για ένα συγκεκριμένο διάστημα αναπαριστά το δυαδικό ένα, ενώ η απουσία για το ίδιο διάστημα αναπαριστά το δυαδικό μηδέν. Κάποια πιο εξελιγμένα συστήματα παρουσιάζουν ποικιλία στην χρονική διάρκεια αυτών των διαστημάτων προκειμένου να μεταφέρουν περισσότερες πληροφορίες. OOK χρησιμοποιείται για τη μετάδοση κώδικα Μορς μέσω ραδιοσυχνοτήτων και αναφέρεται ως λειτουργία CW (συνεχούς κύματος), αν και θεωρητικά μπορεί να χρησιμοποιηθεί οποιοδήποτε ψηφιακό σύστημα κωδικοποίησης. Το OOK είναι πιο αποδοτικό φασματικά από διαμόρφωση μετατόπισης συχνότητας, αλλά πιο ευαίσθητο στο θόρυβο κατά τη χρήση ενός αναγεννητικού δέκτη ή όταν εφαρμόζεται υπερτεροδύναμος δέκτης. Για μια δεδομένη ταχύτητα δεδομένων, το εύρος ζώνης ενός BPSK σήματος και το εύρος ζώνης του OOK σήματος είναι ίσες.

4. Κωδικοποίηση

Στα πλαίσια του παρόντος παραδοτέου, εκτός από την μελέτη των συστημάτων διαμόρφωσης, έγινε και συγκριτική μελέτη με χρήση κωδικοποίησης. Σε ένα ρεαλιστικό σύστημα είναι δεδομένο ότι θα χρησιμοποιηθεί κάποιο είδος κωδικοποίησης για την πιο αποτελεσματική φασματικά εκμετάλλευση του διαύλου επικοινωνίας. Ακολούθως αναφέρονται συνοπτικά οι βασικές έννοιες για τα είδη κωδικοποίησης που εφαρμόστηκαν στα πλαίσια της συγκεκριμένης έρευνας. Στις παραγράφους που ακολουθούν θα αναλύσουμε περαιτέρω αυτά τα σημεία και θα δείξουμε πως μπορούμε να εκτιμήσουμε τις επιδόσεις του σύμφωνου δέκτη σε σχέση με τα σχήματα κωδικοποίησης. Παράλληλα θα μελετήσουμε πως το όλο σύστημα μπορεί να προσομοιωθεί με την βοήθεια του MATLAB ή του GNU-OCTAVE που αποτελεί ένα ευρέως διαδεδομένο πακέτο ελεύθερου λογισμικού.

4.1 Είδη κωδικοποίησης

Στην κωδικοποίηση υπάρχουν δύο κύριες μορφές και οι συνδυασμοί αυτών

- Αλγεβρικοί (block/algebraic codes)
- Συνελκτικοί ή συγκεραστικοί (convolutional codes)

Στην πρώτη κατηγορία, μια ομάδα k συμβόλων αντιστοιχείται, με την προσθήκη επιπλέον συμβόλων, σε n σύμβολα ($n > k$) με στόχο την αύξηση της ελάχιστης απόστασης Hamming, δηλαδή του ελάχιστου αριθμού διαφορετικών συμβόλων ανάμεσα σε οποιοδήποτε ζευγάρι κωδικών λέξεων. Οι πιο διαδεδομένοι κώδικες block είναι οι κώδικες Reed-Solomon, οι οποίοι ενδείκνυνται για κανάλια με ριπές λαθών, όπως π.χ. σε μια OFDM ζεύξη, και μπορούν να διορθώσουν ένα συγκεκριμένο αριθμό λανθασμένων bit, όσο όμως αυτά βρίσκονται εντός ορισμένου αριθμού συμβόλων [15]. Οι συνελκτικοί κώδικες, οι οποίοι θα αναλυθούν

περαιτέρω ακολούθως, αντιστοιχούν μέσω συνέλιξης k ψηφία μιας συνεχόμενης ακολουθίας σε n ψηφία εξόδου και υλοποιούνται με απλούς καταχωρητές ολίσθησης και modulo-2 αθροιστές. Συγκριτικά όμως με ένα σύστημα αλγεβρικού κώδικα παρουσιάζεται μεγαλύτερη καθυστέρηση που επιδρά ιδιαίτερα αρνητικά στις επικοινωνίες μεταγωγής πακέτου.

Μέτρο της απόδοσης ενός κώδικα αποτελεί το κέρδος κωδικοποίησης, που ορίζεται ως το κέρδος στο λόγο πυκνότητας της ενέργειας ψηφίου προς το θόρυβο (E_b/N_0) σε σχέση με ένα μη κωδικοποιημένο σύστημα για συγκεκριμένο BER . Παράλληλα υπάρχει η δυνατότητα συνδυασμού των δύο ειδών κωδικοποίησης, βασικό πλεονέκτημα του οποίου είναι η επίτευξη μεγάλου κέρδους κωδικοποίησης με μικρότερη πολυπλοκότητα υλοποίησης.

Κατά την εφαρμογή της κωδικοποίησης σε ένα σήμα σημαντικό είναι να ληφθεί υπόψιν η σχέση της με το χρησιμοποιούμενο σχήμα διαμόρφωσης έτσι, ώστε να προκύψει το καλύτερο αποτέλεσμα. Στις περισσότερες περιπτώσεις επιλέγεται ένας εκ των $1/2$, $2/3$, $1/4$, $3/4$ ρυθμών κωδικοποίησης σε συνδυασμό με το κατάλληλο είδος διαμόρφωσης, με τη βοήθεια διαγραμμμάτων που απεικονίζουν τη σχέση του BER με το λόγο E_b/N_0 .

4.2 Σύμπλεξη (Interleaving)

Οι περισσότεροι κώδικες διόρθωσης λαθών δεν είναι σχεδιασμένοι ώστε να αντιμετωπίζουν ριπές σφαλμάτων. Έτσι αποδίδουν καλύτερα όταν τα λάθη κατά την λήψη είναι ομοιόμορφα κατανομημένα στο σύνολο των φερουσών. Για την αντιμετώπιση ριπών σφαλμάτων, τα τηλεπικοινωνιακά συστήματα υιοθετούν τη μέθοδο της σύμπλεξης (interleaving) στη σειριακή ακολουθία των bits με σκοπό την τυχαία εμφάνιση των λανθασμένων bits. Στον πομπό, τα κωδικοποιημένα bits συνδυάζονται με τέτοιο τρόπο που εξασφαλίζει ότι, μετά την σύμπλεξη (interleaving), ανάμεσα σε γειτονικά bits παρεμβάλλονται αρκετά άλλα. Στο δέκτη πραγματοποιείται ο αντίστροφος συνδυασμός πριν την αποκωδικοποίηση για την ανάκτηση της αρχικής ακολουθίας με την ταυτόχρονη όμως διασπορά των λαθών στο χρόνο. Ένα ευρέως διαδεδομένο σχήμα σύμπλεξης (interleaving) είναι ο block interleaver, στον οποίο τα δεδομένα εισάγονται σε ένα πίνακα κατά στήλες και εξάγονται κατά γραμμές. Εν γένει το interleaving είναι δυνατό να πραγματοποιηθεί σε επίπεδο συμβόλου ή bit με block ή συνελκτικούς interleavers. Στα πλαίσια της παρούσας μελέτης χρησιμοποιήσαμε ρεαλιστικούς συμπλέκτες από το πρότυπο DVB-S2 [9] αν και αυτό δεν ήταν απαραίτητο λόγω του ότι δεν εφαρμόσαμε OFDM προκειμένου να έχουμε ριπές σφαλμάτων.

5. Προτεινόμενα Σχήματα Κωδικοποίησης

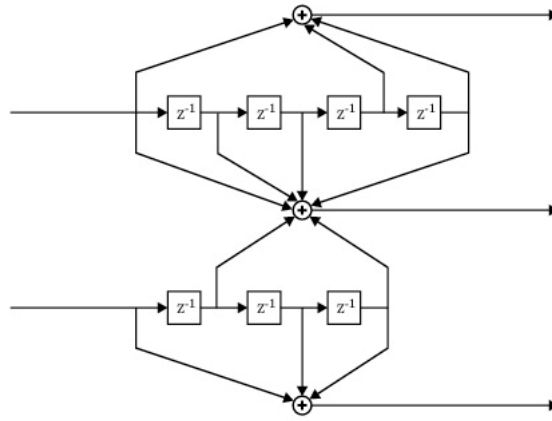
Στα πλαίσια της παρούσας μελέτης χρησιμοποιήσαμε ένα σχήμα κωδικοποίησης από συνελκτικούς κώδικες [14] και ένα σχήμα προερχόμενο από το πρότυπο DVB-S2 που συνδυάζει BCH με LDPC κώδικες [9]. Ακολούθως θα αναφέρουμε εν συντομία κάποια από τα χαρακτηριστικά τους σε σχέση με την παρούσα μελέτη.

5.1 Συνελκτικοί κώδικες

Ο Πίνακας 2 παρεθέτει τα χαρακτηριστικά που χρησιμοποιήθηκαν στα πλαίσια της έρευνας μας σε σχέση με τους συνελκτικούς κώδικες [14], ενώ στην Εικόνα 8 παρατίθεται το διάγραμμα για έναν εκ των δύο κωδικοποιητών που χρησιμοποιήσαμε στα πλαίσια της παρούσας μελέτης.

Πίνακας 2: Παράμετροι συνελκτικών κωδίκων

Ρυθμός R_C	ConstraintLength vector	CodeGenerator matrix
0.5	3	[5 7]
2/3	[5, 4]	[23,35,0; 0,5,13]



Εικόνα 8: Συνελκτικός κωδικοποιητής με ρυθμό 2/3.

Στους συνελκτικούς κώδικες, υπάρχει ένας ολισθητής – καταχωρητής μήκους k_0L . Τα bits πληροφορίας μπαίνουν στον ολισθητή – καταχωρητή, k_0 bits κάθε φορά και στην έξοδο του κωδικοποιητή βγαίνουν n_0 bits, τα οποία είναι ένας γραμμικός συνδυασμός διαφόρων bits του ολισθητή. Στη συνέχεια μεταδίδονται μέσω του διαύλου. Τα n_0 αυτά bits εξαρτώνται όχι μόνο από τα πιο πρόσφατα k_0 bits, που μόλις μπήκαν στον ολισθητή, αλλά και από τα $(L - 1)k_0$ προηγούμενα, που περιέχονται στον ολισθητή και αποτελούν την κατάσταση του (state). Η ποσότητα L ορίζεται ως μήκος εξαναγκασμού (constraint length) του συνελκτικού κώδικα και ο αριθμός δυνατών καταστάσεών του είναι $2^{(L-1)k_0}$. Ο ρυθμός κωδικοποίησης ορίζεται ως [15]

$$R_c = \frac{k_0}{n_0} \quad (9)$$

Η αποκωδικοποίηση επιλέχθηκε να γίνει μέσω της μεθόδου Hard Απόφασης (Hard Decision) και του αλγορίθμου Viterbi [15]. Πρόκειται για τον απλούστερο και συχνότερα χρησιμοποιούμενο τρόπο αποκωδικοποίησης σύμφωνα με τον οποίο λαμβάνουμε δυαδικές αποφάσεις ως προς τις συνιστώσες του λαμβανόμενου σήματος και στην συνέχεια βρίσκουμε την κωδική λέξη που βρίσκεται πλησιέστερα υπό την έννοια της απόστασης Hamming [15].

5.2 BCH κώδικες

Οι BCH είναι από τους πιο γνωστούς και αποτελεσματικούς κυκλικούς κώδικες. Εφευρέθηκαν το 1959 από τον Γάλλο μαθηματικό Alexis Hocquenghem και το 1960 από τους Raj Bose και D.K.-Ray Chaudhuri. Πρόκειται για μια υποκατηγορία κυκλικών κωδίκων, η οποία εν γένει μπορεί να σχεδιαστεί για τη διόρθωση συγκεκριμένου αριθμού σφαλμάτων ανά κωδική λέξη. Είναι ο τρόπος κατασκευής τους, αυτός ο οποίος επιτρέπει για δεδομένο μήκος κώδικα να προκύπτει κώδικας με δεδομένη διορθωτική ικανότητα. Η διαδικασία για την κατασκευή ενός BCH κώδικα επί πεδίου $GF(q)$ μήκους κώδικα n και διορθωτικής ικανότητας t είναι η εξής:

- Προσδιορίζεται πεδίο επέκτασης του $GF(q)$, $GF(qm)$, ελάχιστης δυνατής τάξης που να περιέχει ρίζα α του πολυωνύμου x^{n-1} .
- Για ακέραιο b , προσδιορίζονται τα στοιχεία του πεδίου $GF(qm)$ $\alpha^{b+1}, \alpha^{b+2}, \alpha^{b+3}, \dots, \alpha^{b+2t-1}$
- Προσδιορίζονται τα ελάχιστα πολυώνυμα των παραπάνω στοιχείων ως προς το πεδίο $GF(q)$.
- Προσδιορίζεται το ελάχιστο κοινό πολλαπλάσιο πολυώνυμο $g(x)$ των ελάχιστων πολυωνύμων των παραπάνω στοιχείων.
- Το πολυώνυμο $g(x)$ αποτελεί γεννήτορα πολυώνυμο του κώδικα C μήκους n διάστασης $k=n-q$, (όπου q ο βαθμός του πολυωνύμου g) και διορθωτικής ικανότητας t .

Η ευελιξία στη σχεδιάσή τους και η ύπαρξη ενός δραστικού αλγορίθμου αποκωδικοποίησης, γι' αυτούς τους κώδικες, τους καθιστά ιδιαίτερα ελκυστικούς. Πιο συγκεκριμένα, αποδεικνύεται ότι για οποιαδήποτε m και t , υπάρχει ένας BCH κώδικας με παραμέτρους τις

$$n = 2^m - 1 \quad (10)$$

$$n - k \leq mt \quad (11)$$

$$d_{\min} \geq 2t + 1 \quad (12)$$

Εφόσον τα m και t είναι αυθαίρετα, ο σχεδιαστής του συστήματος έχει μεγάλο αριθμό επιλογών σε αυτήν την οικογένεια κωδίκων. Οι BCH κώδικες έχουν μάλιστα καταχωρηθεί σε πίνακα. Ακολουθεί ο Πίνακας 3 με τα πολυώνυμα που χρησιμοποιήθηκαν στα πλαίσια της παρούσας μελέτης.

Πίνακας 3: Πολυώνυμα BCH κωδίκων

$g_1(x)$	$1+x^2+x^3+x^5+x^{16}$
$g_2(x)$	$1+x+x^4+x^5+x^6+x^8+x^{16}$
$g_3(x)$	$1+x^2+x^3+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{11}+x^{16}$
$g_4(x)$	$1+x^2+x^4+x^6+x^9+x^{11}+x^{12}+x^{14}+x^{16}$
$g_5(x)$	$1+x+x^2+x^3+x^5+x^8+x^9+x^{10}+x^{11}+x^{12}+x^{16}$
$g_6(x)$	$1+x^2+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{12}+x^{13}+x^{14}+x^{15}+x^{16}$
$g_7(x)$	$1+x^2+x^5+x^6+x^8+x^9+x^{10}+x^{11}+x^{13}+x^{15}+x^{16}$
$g_8(x)$	$1+x+x^2+x^5+x^6+x^8+x^9+x^{12}+x^{13}+x^{14}+x^{16}$
$g_9(x)$	$1+x^5+x^7+x^9+x^{10}+x^{11}+x^{16}$
$g_{10}(x)$	$1+x+x^2+x^5+x^7+x^8+x^{10}+x^{12}+x^{13}+x^{14}+x^{16}$
$g_{11}(x)$	$1+x^2+x^3+x^5+x^9+x^{11}+x^{12}+x^{13}+x^{16}$
$g_{12}(x)$	$1+x+x^5+x^6+x^7+x^9+x^{11}+x^{12}+x^{16}$

5.3 LDPC κώδικες

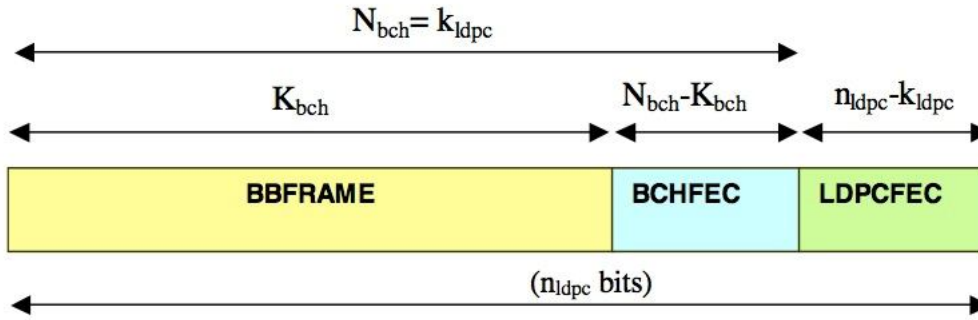
Οι LDPC (κώδικες ελέγχου ισοτιμίας χαμηλής πυκνότητας) χρησιμοποιούνται σε τηλεπικοινωνιακά συστήματα με σκοπό τον εντοπισμό και τη διόρθωση σφαλμάτων. Ανακαλύφθηκαν από τον Gallager το 1962. Ξεχάστηκαν όμως για δεκαετίες εξαιτίας της περιορισμένης τεχνολογικής τεχνογνωσίας. Η χρήση τους προτάθηκε εκ νέου στα μέσα της δεκαετίας του '90 από τον R. Neal και D. Mackay στο Πανεπιστήμιο του Cambridge.

Ο αποκωδικοποιητής που χρησιμοποιήσαμε κάνει 50 επαναλήψεις κατά μέγιστο ενώ χρησιμοποιεί τον βέλτιστο sum-product algorithm [10]. Εφαρμόστηκε soft-demodulation υπολογίζοντας τον λόγο bit Log-Likelihood Ratio (LLR) με χρήση του τελεστή \max^* [11].

Όπως προαναφέραμε το σχήμα που χρησιμοποιήθηκε συνδυάζει BCH με LDPC κώδικες με σύμπλεξη [9]. Το υποσύστημα που υλοποιήθηκε εφαρμόζει το συνδυασμό ενός εξωτερικού (BCH), και ενός εσωτερικού κωδικοποιητή (LDPC) και σύμπλεξης. Η είσοδος αποτελείται από πλαίσια βασικής ζώνης (BBFRAMEs) και η έξοδος από κωδικοποιημένα πλαίσια (FECFRAMEs).

Κάθε πλαίσιο βασικής ζώνης (K_{bch} bits) θα πρέπει να υποβάλλεται σε επεξεργασία από το υποσύστημα κωδικοποίησης, για να δημιουργήσει ένα FECFRAME (n_{ldpc} bits).

Τα bits ελέγχου ισοτιμίας (BCHFEC) του συστηματικού BCH εξωτερικού κώδικα πρέπει να προσαρτηθούν μετά την ακολουθία των bits που αποτελούν το BBFRAME, και τα bits ελέγχου ισοτιμίας (LDPCFEC) του εσωτερικού κωδικοποιητή LDPC προσαρτώνται μετά την ακολουθία των bits BCHFEC, όπως φαίνεται στην Εικόνα 9.



Εικόνα 9: Μορφή των δεδομένων πριν την σύμπλεξη ($n_{ldpc} = 64\ 800$ bits for normal FECFRAME, $n_{ldpc} = 16\ 200$ bits for short FECFRAME).

6. Εκτίμηση της Φάσης

6.1 Επίδραση της μεταβολής της φάσης

Η εκτίμηση της φάσης του οπτικού σήματος είναι ιδιαίτερα σημαντικό στοιχείο για την σωστή λειτουργία του δέκτη ιδιαίτερα για σχήματα διαμόρφωσης τα οποία είναι ευαίσθητα σε μεταβολές της φάσης όπως για παράδειγμα το PSK. Οι μεταβολές της φάσης προέρχονται κυρίως από δύο αιτίες: η πρώτη είναι ο θόρυβος φάσης του laser εκπομπής και του laser του τοπικού ταλαντωτή. Η δεύτερη αιτία είναι η διαφορά στο μήκος κύματος εκπομπής μεταξύ των δύο πηγών $\Delta\lambda = \lambda - \lambda_{LO}$ που μεταφράζεται σε μία διαφορά της κεντρικής συχνότητας $\Delta f = c/\lambda - c/\lambda_{LO}$ όπου λ_{LO} είναι το μήκος κύματος του τοπικού ταλαντωτή. Η διαφορά φάσης που δημιουργείται δίνεται από την σχέση:

$$\Delta\phi = (\Delta f)t + \Delta\theta \quad (13)$$

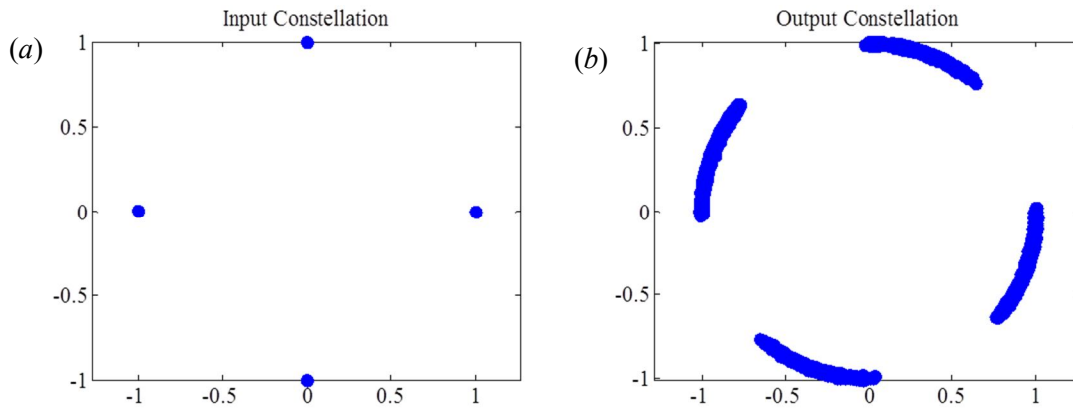
όπου $\Delta\theta$ είναι η διαφορά εξαιτίας του θορύβου φάσης. Στην πράξη τα τηλεπικοινωνιακά laser ημιαγωγού μπορεί να παρουσιάζουν μία διαφορά $\pm 2\text{GHz}$ από την ονομαστική τιμή συχνότητας του κατασκευαστή η οποία μάλιστα μπορεί να μεταβάλλεται με το χρόνο και την θερμοκρασία. Υπάρχουν αρκετοί αλγόριθμοι που επιτρέπουν την εκτίμηση της συχνότητας με μεγάλη ακρίβεια [12],[13] και για το λόγο αυτό θα θεωρήσουμε ότι η συχνοτική διαφορά μεταξύ του σήματος λήψης και του τοπικού ταλαντωτή είναι σχεδόν μηδενική, $\Delta f \approx 0$.

Όπως είδαμε και στα πλαίσια του Π1.2, τα ρεύματα στην έξοδο του δέκτη δίνονται από την σχέση:

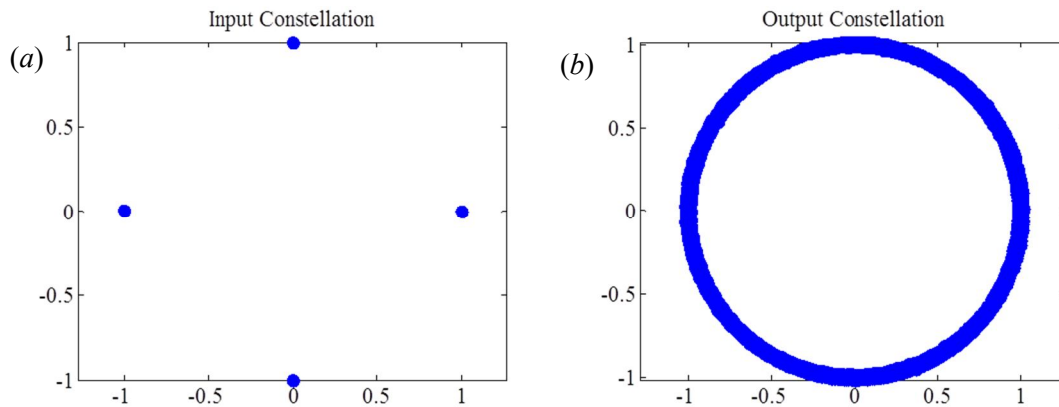
$$i_I = \frac{R\sqrt{P_{LO}P_R}}{2} \{e_I \cos(\Delta\phi) - e_Q \sin(\Delta\phi)\} + n_I \quad (14)$$

$$i_Q = \frac{R\sqrt{P_{LO}P_R}}{2} \{e_I \sin(\Delta\phi) + e_Q \cos(\Delta\phi)\} + n_Q \quad (15)$$

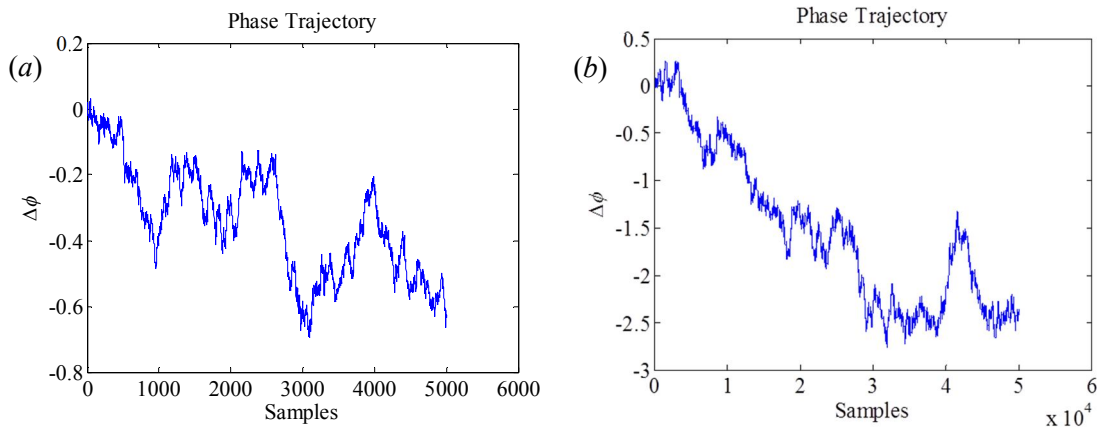
Παρατηρούμε επομένως ότι η μεταβολή της φάσης προκαλεί στροφή του αστερισμού των συμβόλων κατά μία γωνία $\Delta\phi$ και επομένως μπορεί να οδηγήσει σε εσφαλμένη αποκωδικοποίηση των συμβόλων.



Εικόνα 10: Αρχικός και τελικός αστερισμός συμβόλων για την περίπτωση ενός PSK σήματος θεωρώντας την μετάδοση 500 συμβόλων.



Εικόνα 11: Αρχικός και τελικός αστερισμός συμβόλων για την περίπτωση ενός PSK σήματος θεωρώντας την μετάδοση 5000 συμβόλων.



Εικόνα 12: Παράδειγμα εξέλιξης φάσης στην περίπτωση που μεταδίδουμε 5000 και 50000 δείγματα.

Η Εικόνα 10 και η Εικόνα 11 δείχνουν την επίδραση του θορύβου φάσης με την πάροδο του χρόνου στην περίπτωση όπου μεταδίδουμε 5×10^3 και 5×10^4 για το σύστημα που έχει παραμέτρους οι οποίες αναφέρει ο

Πίνακας 1. Έχουμε θεωρήσει μόνο την επίδραση του θορύβου φάσης και έχουμε αγνοήσει στην συνεισφορά του προσθετικού θορύβου ($n_I = n_Q = 0$). Παρατηρούμε ότι στην πρώτη περίπτωση (Εικόνα 10) υπάρχει μία μερική στροφή των συμβόλων στο δέκτη εξαιτίας του θορύβου φάσης ενώ στην περίπτωση όπου έχουμε μεταδώσει περισσότερα σύμβολα (Εικόνα 11), η στροφή είναι μεγαλύτερη και πλέον δεν μπορούν να ξεχωρίσουν οι περιοχές που αντιστοιχούν στα αρχικά σύμβολα. Αυτή η συμπεριφορά οφείλεται στο ότι η αβεβαιότητα φάσης αυξάνεται με την πάροδο του χρόνου. Όπως είδαμε στο Π1.2, έχουμε

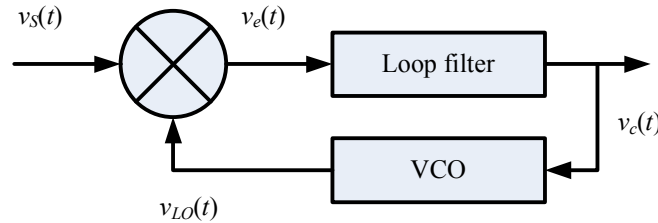
$$\langle \Delta\phi^2 \rangle = 2\pi(\Delta\nu_T + \Delta\nu_{LO})\Delta t \quad (16)$$

όπου Δt είναι το χρονικό διάστημα στο οποίο θεωρούμε τις μεταβολές της φάσης που στα συγκεκριμένα παραδείγματα θεωρείται ίσο με $N_s T_s$ όπου N_s είναι το πλήθος των υπό μετάδοση συμβόλων και $T_s=1/R_s$ είναι η διάρκεια του κάθε συμβόλου. Στην Εικόνα 12 δείχνουμε παραδείγματα της εξέλιξη της φάσης που αντιστοιχούν στην μετάδοση 5×10^3 και 5×10^4 . Στην δεύτερη περίπτωση η ολίσθηση φάσης κατά απόλυτη τιμή ξεπερνάει κατά πολύ το $\pi/2 \approx 1.57$ και επομένως για αυτό το λόγο οι περιοχές που αντιστοιχούν στα σύμβολα εξόδου δεν είναι διακριτές και επικαλύπτονται.

6.2 Μέθοδοι εκτίμησης φάσης

Για την εκτίμηση της ολίσθησης φάσης του σήματος $\Delta\phi$ μπορούμε να ακολουθήσουμε δύο στρατηγικές: α) την εκτίμηση της στο φυσικό επίπεδο με αναλογικό τρόπο και β) η εκτίμησης της στο ψηφιακό επίπεδο. Στην πρώτη κατηγορία ανήκουν τεχνικές που βασίζονται σε οπτικά κυκλώματα βρόγχων κλειδώματος φάσης (phase locked loop – PLL) ενώ στην δεύτερη ο υπολογισμός της φάσης γίνεται μέσω με ψηφιακή επεξεργασία σήματος (digital signal processing – DSP).

6.3 Εκτίμηση φάσης με PLL.



Εικόνα 13: Η αρχή λειτουργίας του PLL.

Στην Εικόνα 13 δείχνουμε την αρχή λειτουργίας ενός ηλεκτρονικού PLL που αποτελεί και την βάση για την κατανόηση της αρχής λειτουργίας του οπτικού PLL. Στην συγκεκριμένη υλοποίηση η τάση του σήματος $v_s(t)$ πολλαπλασιάζεται με την τάση του τοπικού ταλαντωτή $v_{LO}(t)$ και περνάει μέσα από ένα ηλεκτρονικό φίλτρο η έξοδος του οποίου τροφοδοτεί έναν ταλαντωτή τάσης ελεγχόμενο από τάση (voltage control oscillator – VCO). Το φίλτρο έχει στόχο την μείωση της ισχύος του θορύβου ενώ παράλληλα φιλτράρει τις υψηλές συχνότητες που σχηματίζονται στην έξοδο του πολλαπλασιαστή. Τα σήματα στην είσοδο γράφονται συνήθως ως εξής:

$$v_s(t) = V_s \sin(2\pi f_s t + \phi_s) \quad (17)$$

$$v_{LO}(t) = V_{LO} \cos(2\pi f_{LO} t + \phi_{LO}) \quad (18)$$

Στην περίπτωση όπου έχουμε μικρές διαφορές στην συχνότητα, μπορούμε να δείξουμε ότι το σήμα στην έξοδο του πολλαπλασιαστή δίνεται από την σχέση

$$v_e(t) \cong \frac{K_{PD} V_s V_{LO}}{2} \sin(\phi_s - \phi_{LO}) \quad (19)$$

όπου K_{PD} είναι μία σταθερά η οποία εξαρτάται από την υλοποίηση του πολλαπλασιαστή και έχουμε αφαιρέσει τις υψηλές συχνότητες. Η φάση εξόδου του VCO δίνεται από την σχέση,

$$\phi_{LO}(t) = G_{VCO} \int_{-\infty}^t v_c(\tau) d\tau \quad (20)$$

όπου η έξοδος του φίλτρου ανάδρασης καθορίζεται από την σχέση

$$u_c(t) = \int_{-\infty}^{+\infty} h(t-\tau)u_e(\tau)d\tau \quad (21)$$

όπου $h(t)$ είναι η κρουστική απόκριση του φίλτρου.

Υπάρχουν αρκετές προτάσεις για το πώς το PLL μπορεί να υλοποιηθεί στις σύμφωνες οπτικές επικοινωνίες και ο ενδιαφερόμενος αναγνώστης παραπέμπεται στην βιβλιογραφία για το σκοπό αυτό [5].

6.4 Ψηφιακή εκτίμηση φάσης

Ίσως ο πιο απλός και πρακτικός τρόπος να γίνει εκτίμηση της φάσης είναι στο ψηφιακό επίπεδο χρησιμοποιώντας απλά κυκλώματα επεξεργασίας σήματος. Για να γίνει αυτό θα πρέπει καταρχήν να διακρίνουμε την ολίσθηση φάσης που προκαλείται από τον θόρυβο φάσης ή την συχνοτική διαφορά μεταξύ τοπικού ταλαντωτή και οπτικής πηγής από τις μεταβολές της φάσης που οφείλονται στην διαμόρφωση. Για τον σκοπό αυτό αρχικά κατασκευάζουμε το μιγαδικό σήμα λήψης

$$z = z_r + jz_i = i_l + ji_Q \quad (22)$$

Αν στις **Error! Reference source not found.**(15) αγνοήσουμε τον προσθετικό θόρυβο τότε εύκολα βλέπουμε ότι:

$$z_r = \frac{R\sqrt{P_{LO}P_R}}{2} \{e_l \cos(\Delta\phi) - e_Q \sin(\Delta\phi)\} = \frac{R\sqrt{P_{LO}P_R}}{2} \operatorname{Re}\{Ee^{j\Delta\phi}\} \quad (23)$$

$$z_i = \frac{R\sqrt{P_{LO}P_R}}{2} \{e_l \sin(\Delta\phi) + e_Q \cos(\Delta\phi)\} = \frac{R\sqrt{P_{LO}P_R}}{2} \operatorname{Im}\{Ee^{j\Delta\phi}\} \quad (24)$$

όπου με E συμβολίζουμε το μιγαδικό σύμβολο,

$$E = e_Q + je_l \quad (25)$$

Επομένως θα έχουμε:

$$z = \frac{R\sqrt{P_{LO}P_R}}{2} Ee^{j\Delta\phi} \quad (26)$$

Στην περίπτωση όπου χρησιμοποιούμε PSK τάξεως M θα έχουμε,

$$E = \exp\left(j\frac{2\pi}{M}m\right) \quad (27)$$

όπου m είναι ακέραιος οπότε εύκολα καταλαβαίνουμε ότι,

$$E^M = \exp(j2\pi) = 1 \quad (28)$$

δηλαδή η ύψωση του μιγαδικού συμβόλου στην M -οστή αφαιρεί την μεταβολή της φάσης που οφείλεται στην διαμόρφωση. Επομένως αν υψώσουμε το z στην M -οστή η φάση που θα απομείνει θα οφείλεται στον θόρυβο φάσης ή στην συχνοτική διαφορά των δύο laser,

$$z^M = \left[\frac{R\sqrt{P_{LO}P_R}}{2}\right]^M e^{jM\Delta\phi} \quad (29)$$

οπότε η φάση $\arg\{z^M\}$ του z^M καθορίζεται μονάχα¹ από το $M\Delta\phi$.

Σε ένα υψύρυστο σύστημα αναμένουμε ότι ο ρυθμός μετάδοσης συμβόλων R_S να είναι πολύ μεγαλύτερος από το άθροισμα των ευρών γραμμής $\Delta\nu_I + \Delta\nu_{LO}$ οπότε θα έχουμε $(\Delta\nu_I + \Delta\nu_{LO})T_S \ll 1$ και επομένως,

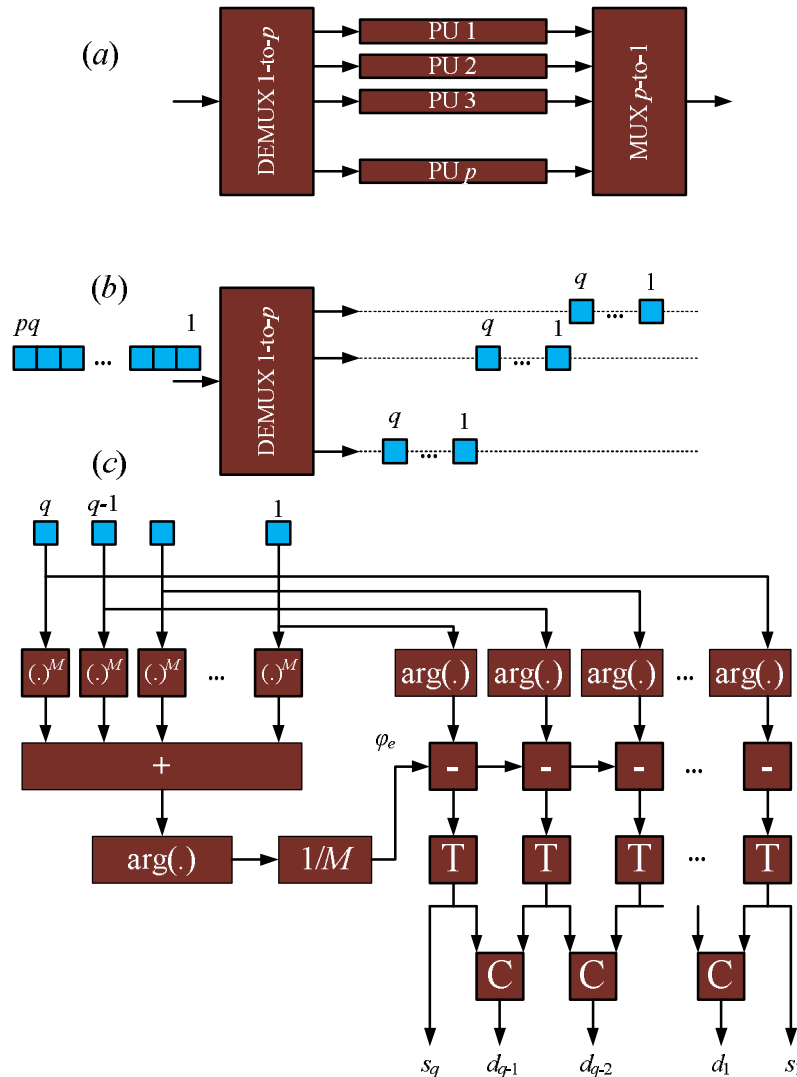
¹ Στο σημείο αυτό θα πρέπει να σημειώσουμε ότι δεν μπορούμε να γράψουμε απευθείας ότι $\arg\{z^M\} = M\Delta\phi$ επειδή εξ ορισμού $-\pi \leq \arg\{Z\} < \pi$ οπότε η πιο σωστή έκφραση θα ήταν $\arg\{z^M\} = 2\pi\kappa + M\Delta\phi$ όπου κ ένας ακέραιος.

$$\sigma_\phi^2 = \langle \Delta\phi^2 \rangle = \langle [\phi(t+T_s) - \phi(t)]^2 \rangle = 2\pi(\Delta\nu_T + \Delta\nu_{LO})T_s \ll 1 \quad (30)$$

Για παράδειγμα για τις παραμέτρους τις οποίες παραθέτει ο

Πίνακας 1, έχουμε $T_s=1\text{ns}$ και $\Delta\nu_T+\Delta\nu_R=700\text{kHz}$ οπότε το μέσο τετραγωνικό σφάλμα φάσης είναι $\sigma_\phi \approx \pi/50$ που υποδεικνύει ότι η μεταβολή της φάσης είναι πολύ μικρή στη διάρκεια μερικών συμβόλων και μπορεί να θεωρηθεί σταθερή. Επομένως παρουσία και του προσθετικού θορύβου, μπορούμε να εκτιμήσουμε τη φάση υπολογίζοντας τη φάση ϕ_e του αθροίσματος Z του z^M από q διαδοχικά σύμβολα διαιρεμένο με τον τάξη της διαμόρφωσης M ,

$$Z = \frac{z_{i-q+1}^M + \dots + z_i^M}{M} \quad (31)$$



Εικόνα 14: Εκτίμηση της φάσης με ψηφιακή λογική.

Στην πράξη, η εκτίμηση φάσης μπορεί να γίνει με τον τρόπο που δείχνουμε στην Εικόνα 14 [6]. Αρχικά τα pq πλήθος σύμβολα εισέρχονται σε έναν αποτολυπλέκτη 1-σε- p και ομαδοποιούνται σε p ομάδες από q σύμβολα η κάθε μία. Στη συνέχεια κάθε ομάδα συμβόλων αποστέλλεται σε μία μονάδα επεξεργασίας (processing units – PUs). Κάθε PU υπολογίζει το άθροισμα των q συμβόλων και την φάση του αθροίσματος αυτού την οποία και αφαιρεί από την φάση των επιμέρους δειγμάτων. Στη συνέχεια η φάση των συμβόλων περνάει από μία διάταξη καταφλίου (που συμβολίζεται με T στην Εικόνα 14) στην οποία αποφασίζουμε σε ποιο PSK σύμβολο αντιστοιχεί.

Θα πρέπει να σημειωθεί ότι για να διευκολυνθεί η ανίχνευση της φάσης, συχνά στην διαμόρφωση PSK πραγματοποιείται διαφορικός τρόπος προ-κωδικοποίησης. Δηλαδή τα διαδοχικά σύμβολα αποτυπώνονται στις διαφορές της φάσης και όχι απευθείας στην φάση αυτή κάθε αυτή. Επομένως η φάση του σήματος γράφεται την χρονική στιγμή $t=kT_s$

$$\phi_s(kT_s) = \Phi_k = \sum_{i=-\infty}^k \phi_i \quad (32)$$

όπου ϕ_i είναι η φάση του i -οστού PSK συμβόλου δ_i προς μετάδοση και η οποία είναι ακέραιο πολλαπλάσιο του $2\pi/M$. Επομένως απουσία τυχαίων μεταβολών της φάσης και προσθετικού θορύβου, το σύμβολο που φτάνει στον δέκτη είναι το $s_k = \exp(j\Phi_k)$ και εύκολα μπορούμε να δούμε ότι,

$$\arg\{s_{k+1}\} - \arg\{s_k\} = \Phi_{k+1} - \Phi_k + 2\pi(i_{k+1} - i_k) = \phi_{k+1} + 2\pi(i_{k+1} - i_k) \quad (33)$$

όπου i_{k+1} και i_k είναι ακέραιοι τέτοιοι ώστε οι γωνίες $\Phi_{k+1} - 2\pi i_{k+1}$ και $\Phi_k - 2\pi i_k$ να ανήκουν στο διάστημα στο $(-\pi, \pi]$. Εφόσον όμως $-\pi < \phi_{k+1} \leq \pi$ έπεται ότι

$$\arg\{s_{k+1}\} - \arg\{s_k\} = \phi_{k+1} \quad (34)$$

Επομένως, μία στρατηγική για την ανάδειξη της φάσης των διαφορικών συμβόλων είναι απλά να αφαιρούμε το όρισμα των συμβόλων λήψης σε δύο διαδοχικά σύμβολα και με τον τρόπο αυτό να υπολογίζουμε το όρισμα του διαφορικού συμβόλου σύμφωνα με την (34). Αυτή η μέθοδος αντιστοιχεί στο DPSK και επιβαρύνεται από το ότι η εκτίμηση της διαφορικής φάσης επιβαρύνεται από τα ενδεχόμενα σφάλματα φάσης $\delta\phi_{k+1}$ και $\delta\phi_k$ λόγω του προσθετικού θορύβου τόσο στο s_{k+1} όσο και στο s_k . Πράγματι παρουσία των σφαλμάτων αυτών θα έχουμε,

$$\arg\{s_{k+1}\} - \arg\{s_k\} = \phi_{k+1} + \delta\phi_{k+1} - \delta\phi_k \quad (35)$$

οπότε αν θεωρήσουμε ότι τα σφάλματα είναι ανεξάρτητα με μηδενική μέση τιμή, τότε η διακύμανση του διαφορικού σφάλματος $\delta\phi_{k+1} - \delta\phi_k$ θα είναι

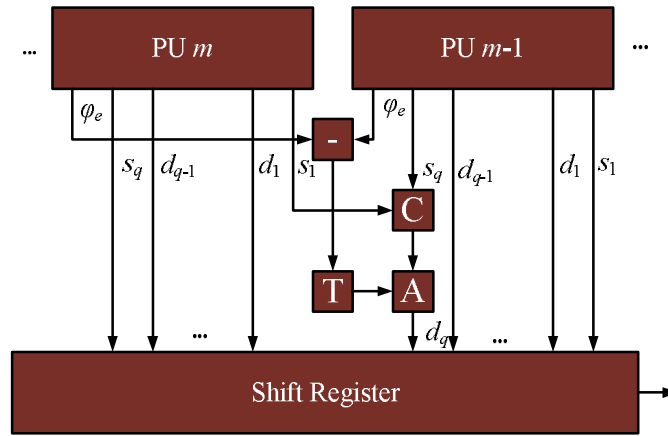
$$\langle (\delta\phi_{k+1} - \delta\phi_k)^2 \rangle = \langle \delta\phi_{k+1}^2 \rangle + \langle \delta\phi_k^2 \rangle \quad (36)$$

και επομένως μεγαλύτερη από τις επιμέρους διακυμάνσεις $\langle \delta\phi_{k+1}^2 \rangle$ και $\langle \delta\phi_k^2 \rangle$ υπονοώντας μία ποινή ισχύος (power penalty). Μία εναλλακτική προσέγγιση είναι να προσπαθήσουμε να εκτιμήσουμε πρώτα τα σύμβολα PSK \bar{s}_k που λαμβάνουμε στο δέκτη με μία διάταξη κατωφλίου και στην συνέχεια να υπολογίσουμε τα διαφορικά σύμβολα. Στην αθόρυβη περίπτωση θα έχουμε $\bar{s}_k = s_k$ οπότε

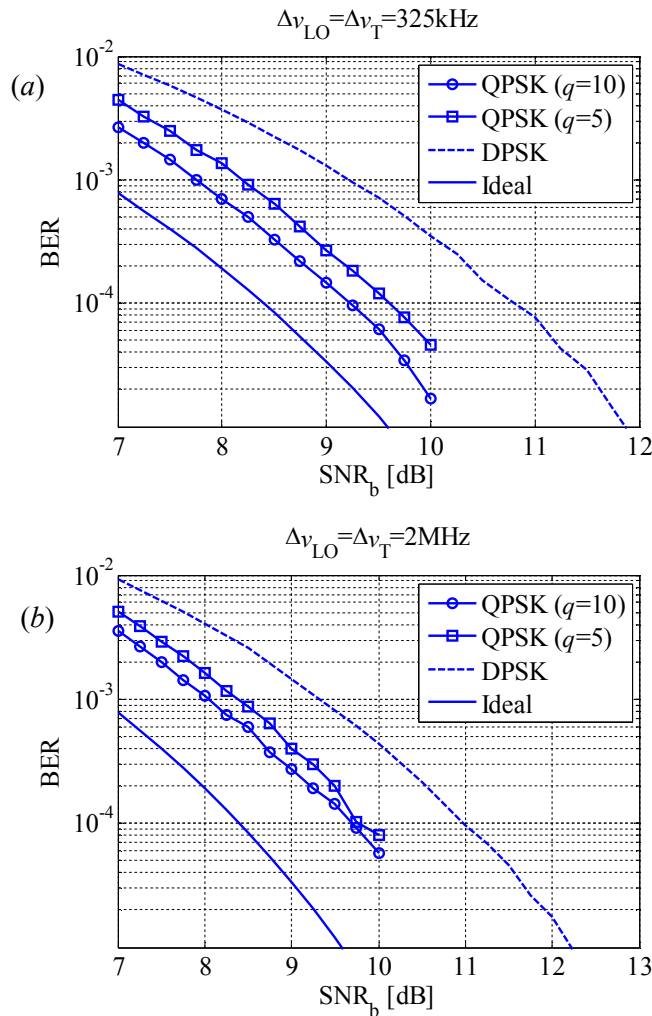
$$\arg\{\bar{s}_{k+1}\} - \arg\{\bar{s}_k\} = \phi_{k+1} \quad (37)$$

που συμπίπτει με την (34) οπότε η διαδικασία είναι πανομοιότυπη, ωστόσο στην ενθόρυβη περίπτωση η συμπεριφορά του θορύβου είναι διαφορετική. Ωστόσο στην περίπτωση αυτή, για να γίνει σωστά η αποκωδικοποίηση των συμβόλων θα πρέπει να προσθέσουμε και την ολίσθηση φάσης η οποία εκτιμάται από αφαιρώντας το όρισμα ϕ_e του Z από τα ορίσματα του s_k και s_{k+1} όπως δείχνει και η Εικόνα 14 σε μία προσπάθεια να απομακρύνουμε τον θόρυβο φάσης.

Όπως φαίνεται και στην Εικόνα 14, σε κάθε μονάδα επεξεργασίας κάνουμε διαφορική αποκωδικοποίηση των συμβόλων \bar{s}_k και υπολογίζουμε τα διαφορικά σύμβολα d_k στο δέκτη. Κάθε μονάδα έχει $(q+1)$ εξόδους οι δύο εξωτερικές εκ των οποίων δεν περιέχουν τα διαφορικά σύμβολα αλλά τα μη διαφορικά σύμβολα \bar{s}_q και \bar{s}_1 . Στην Εικόνα 15 δείχνουμε τον τρόπο που γίνεται η πολυπλεξία των συμβόλων. Τα ακριανά σύμβολα μεταξύ δύο μονάδων επεξεργασίας συγκρίνονται και αποκωδικοποιούνται διαφορικά. Ωστόσο στην περίπτωση όπου η διαφορά στις εκτιμήσεις της φάσης ϕ_e είναι μεγαλύτερη από π/M τότε το διαφορικό σύμβολο που αποκωδικοποιείται είναι το προηγούμενο σε σειρά στον αστερισμό από αυτό που προκύπτει από την διαφορική αποκωδικοποίηση ενώ το αντίθετο συμβαίνει στην περίπτωση όπου η διαφορά είναι μικρότερη από $-\pi/M$. Αυτό γίνεται ώστε η εκτίμηση της φάσης να παρακολουθεί την σωστή πορεία της όπως εξηγείται αναλυτικά στην [7].



Εικόνα 15: Πολυπλεξία των συμβόλων.



Εικόνα 16: Επιδόσεις QPSK χρησιμοποιώντας εκτίμηση φάσης για δύο διαφορετικά εύρη γραμμής πομπού και τοπικού ταλαντωτή.

Στην Εικόνα 16 δείχνουμε τις επιδόσεις του συστήματος στην περίπτωση όπου χρησιμοποιούμε εκτίμηση φάσης για δύο διαφορετικά εύρη γραμμής πομπού και τοπικού ταλαντωτή. Εξετάσαμε επίσης τις περιπτώσεις όπου $q=5$ και $q=10$ καθώς και την ιδανική περίπτωση όπου υπάρχει μόνο προσθετικός θόρυβος και όχι θόρυβος φάσης. Παρατηρούμε ότι σε κάθε περίπτωση οι μέθοδοι εκτίμησης φάσης επιφέρουν μία σημαντική βελτίωση σε σχέση με το DPSK. Επίσης η χρήση υπολογιστικών μονάδων με μεγαλύτερο q επιφέρει σημαντικότερη βελτίωση όταν το εύρος γραμμής είναι μικρό και επομένως η φάση μπορεί να θεωρηθεί σταθερή στη διάρκεια ενός συμβόλου. Όταν $\Delta\nu_T = \Delta\nu_R = 2\text{MHz}$ υπάρχει μικρότερη βελτίωση όταν

χρησιμοποιούμε $q=10$ σε σχέση με το σύστημα που χρησιμοποιεί $q=5$. Σημειώνουμε ότι στα αποτελέσματα αυτά έχουμε αυξήσει τον ρυθμό συμβόλων σε 10GS/s ώστε τα αποτελέσματα μας να είναι συγκρίσιμα με την [6].

7. Αποτελέσματα

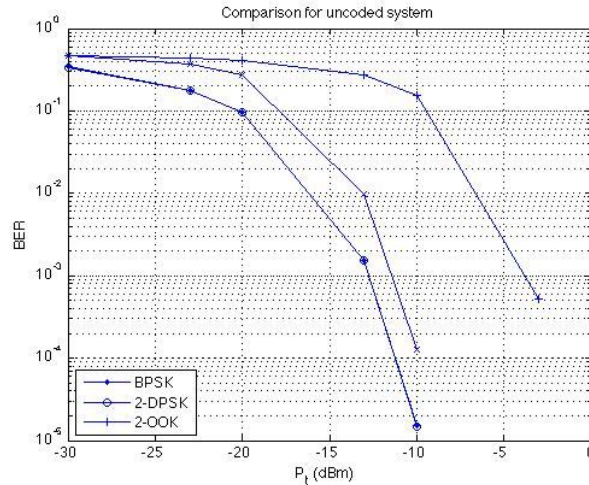
Με βάση τα παραπάνω αναπτύξαμε κώδικα προκειμένου να προσομοιώσουμε την συμπεριφορά του συστήματος στο Octave/Matlab. Στα πλαίσια της παρούσας μελέτης εκτελέσαμε 3 ανεξάρτητα σύνολα πειραμάτων. Ο Πίνακας 4 συνοψίζει τα αποτελέσματα όπως αυτά παρουσιάζονται στα αντίστοιχα σχήματα. παρουσιάζει την τάξη διαμόρφωσης και το επικρατέστερο σχήμα διαμόρφωσης κατά την μελέτη προσομοίωσης. Θα πρέπει να αναφερθεί ότι ενώ σε ένα ρεαλιστικό σύστημα θα έπρεπε να υπολογιστεί ένας θόρυβος φάσης και να υπάρχει μία διαδικασία εκτίμησης φάσης, κάτι το οποίο υποστηρίζεται από το μοντέλο μας, όλα τα αποτελέσματα που παραθέτουμε εδώ είναι χωρίς εκτίμηση φάσης. Αυτό έγινε για να υπάρξει μία πρώτη αξιολόγηση των αποτελεσμάτων αλλά και του μοντέλου στην πιο απλή περίπτωση όπου είναι εύκολο να γίνει σύγκριση.

Πίνακας 4: Παράμετροι διαμόρφωσης

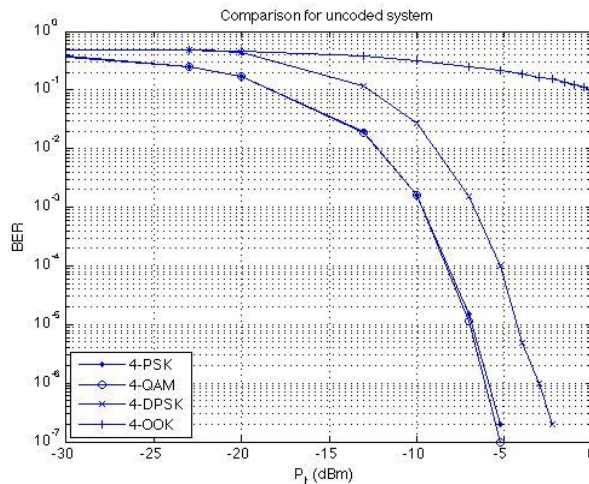
A/A	Τάξη Διαμόρφωσης	Κωδικοποίηση	Νικητής	Σχήμα
1	2	-	BPSK	Εικόνα 17
2	4	-	4-PSK/4-QAM	Εικόνα 18
3	8	-	8-PSK	Εικόνα 19
4	16	-	16-QAM	Εικόνα 20
5	32	-	32-QAM	Εικόνα 21
6	64	-	64-QAM	Εικόνα 22
7	8	CC 1/2	8-PSK	Εικόνα 23
8	16	CC 1/2	16-PSK	Εικόνα 24
9	32	CC 1/2	32-PSK	Εικόνα 25
10	64	CC 1/2	64-QAM	Εικόνα 26
11	8	CC 2/3	8-PSK	Εικόνα 27
12	16	CC 2/3	16-PSK	Εικόνα 28
13	32	CC 2/3	32-PSK	Εικόνα 29
14	64	CC 2/3	64-QAM	Εικόνα 30
15	8	LDPC $\frac{3}{4}$	8PSK	Εικόνα 31
16	16	LDPC $\frac{3}{4}$	16PSK	Εικόνα 32
17	32	LDPC $\frac{3}{4}$	32PSK	Εικόνα 33
18	64	LDPC $\frac{3}{4}$	64QAM	Εικόνα 34

7.1 Επιδόσεις Συστήματος Χωρίς Κωδικοποίηση

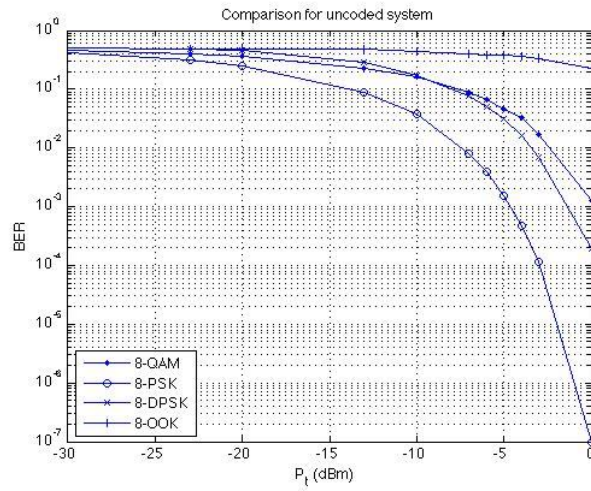
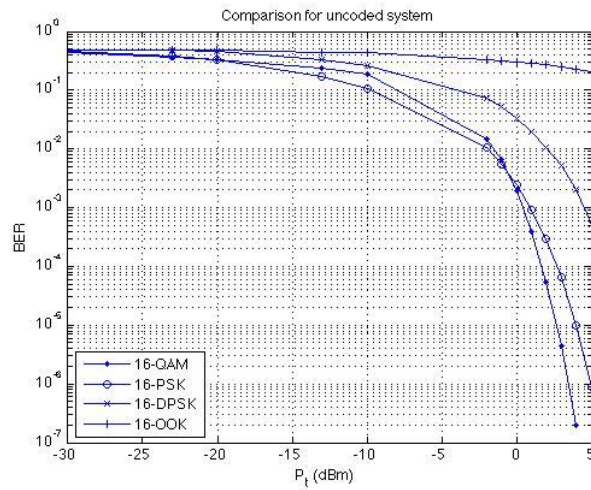
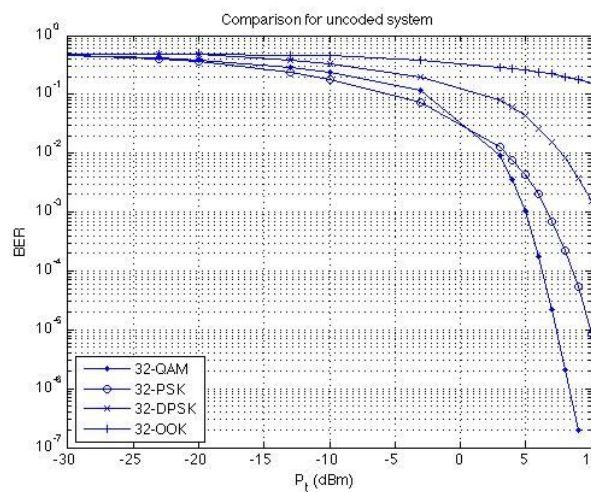
Το πρώτο σύνολο από πειράματα αφορά την εκτίμηση της επίδοσης διαφορετικών συστημάτων διαμόρφωσης για κανάλι το οποίο δεν υφίσταται κωδικοποίηση. Όπως προαναφέραμε συγκρίναμε τα ακόλουθα συστήματα διαμόρφωσης QAM, PSK, DPSK και OOK για διάφορες τάξεις αλφαβήτου $M=2, 4, 8, 16, 32$ και 64 . Από το πρώτο σύνολο πειραμάτων προκύπτει ότι τα επικρατέστερα συστήματα διαμόρφωσης ανάλογα με την τάξη του αλφαβήτου M που επιλέγεται για την επικοινωνία απουσία κωδίκων διόρθωσης σφαλμάτων είναι τα BPSK για τάξη $M=2$, τα 4-QAM/4-PSK για τάξη $M=4$, το 8-PSK για τάξη $M=8$, το 16-QAM για $M=16$, 32-QAM για το $M=32$, και τέλος το 64-QAM για το $M=64$.

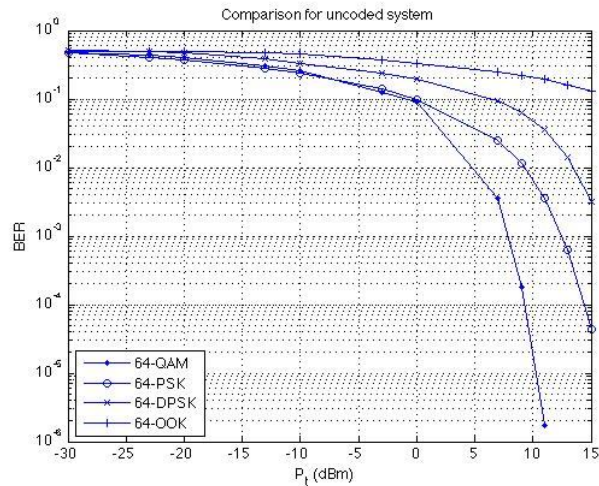


Εικόνα 17: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=2$.



Εικόνα 18: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=4$.

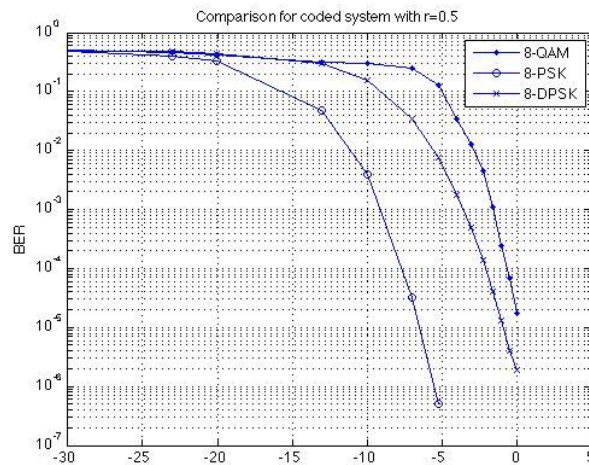
Εικόνα 19: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=8$.Εικόνα 20: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=16$.Εικόνα 21: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=32$.



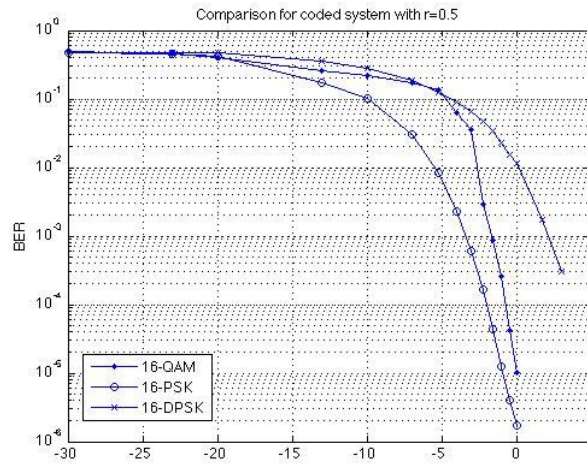
Εικόνα 22: Επιδόσεις για σύστημα χωρίς κωδικοποίηση, $M=64$.

7.2 Συνελικτικοί κώδικες (CC)

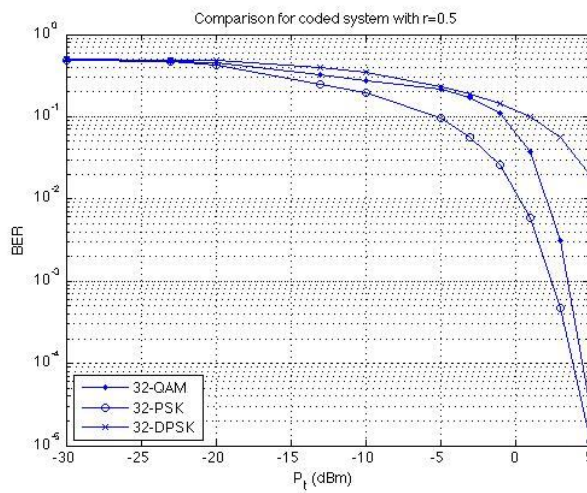
Ακολούθως εκτελέστηκε ένα σύνολο συγκριτικών πειραμάτων όπου χρησιμοποιήθηκαν συνελκτικοί κώδικες (CC) με δύο διαφορετικούς ρυθμούς κωδικοποίησης $1/2$ αλλά και $2/3$. Δοκιμάστηκαν μόνο τα συστήματα διαμόρφωσης PSK/DPSK και QAM καθώς το OOK φάνηκε πως υστερεί πολύ από τις επιδόσεις του συστήματος χωρίς κωδικοποίηση όπου δεν χρησιμοποιούνται κώδικες διόρθωσης σφαλμάτων. Η τάξη διαμόρφωσης ορίστηκε $M > 8$ καθώς για την περίπτωση $M \leq 4$ δεν υπάρχει ανταγωνισμός. Ακολουθούν τα σχήματα με τα αποτελέσματα από τα οποία γίνεται φανερό ότι ανεξάρτητα από τον ρυθμό κωδικοποίησης για τις τάξεις $M=8, 16$ και 32 επικρατεί το PSK ενώ για $M=64$ το QAM.



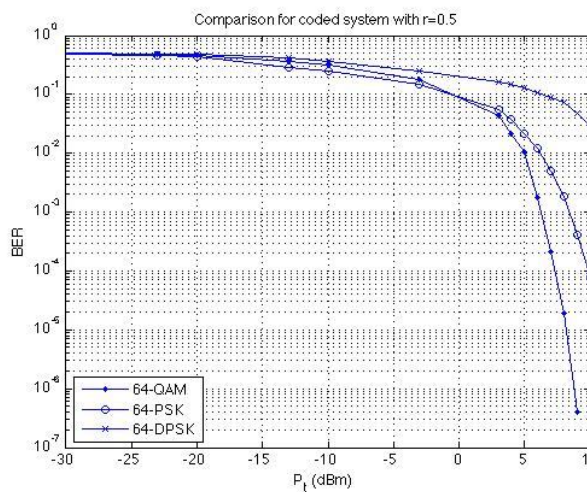
Εικόνα 23: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό $1/2$, $M=8$.



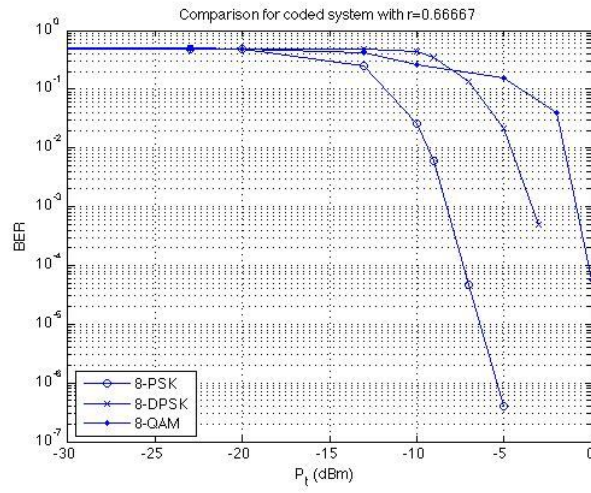
Εικόνα 24: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 1/2, $M=16$.



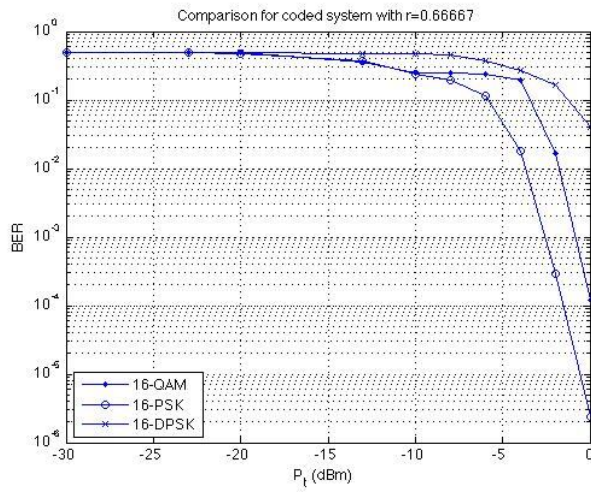
Εικόνα 25: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 1/2, $M=32$.



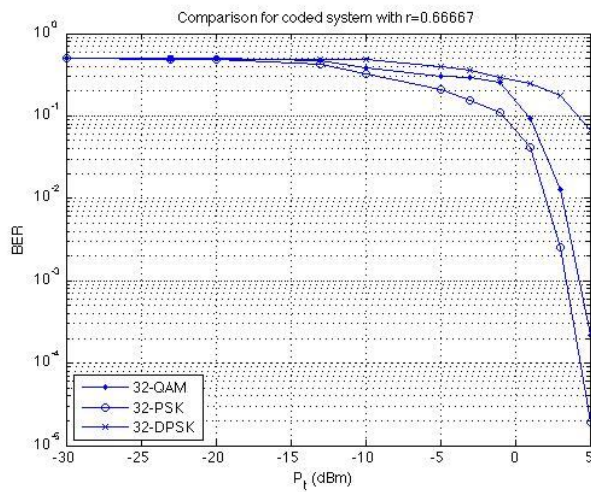
Εικόνα 26: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 1/2, $M=64$.



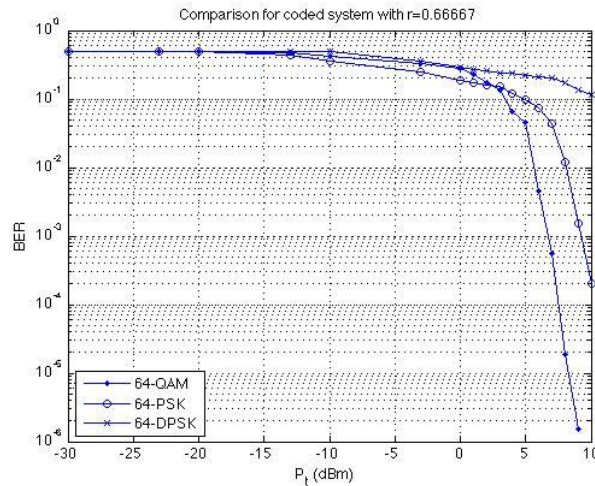
Εικόνα 27: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=8$.



Εικόνα 28: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=16$.



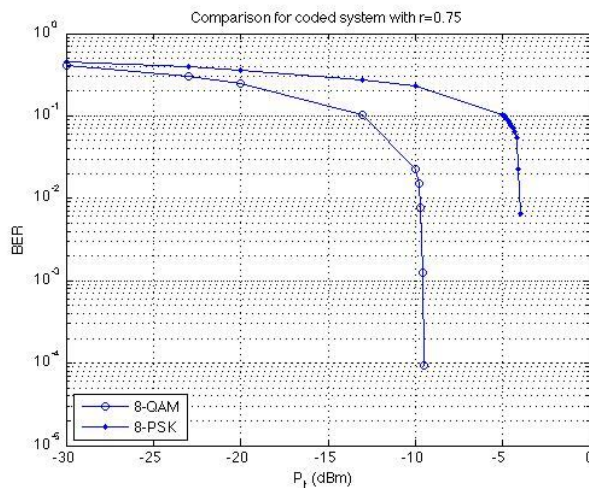
Εικόνα 29: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=32$.



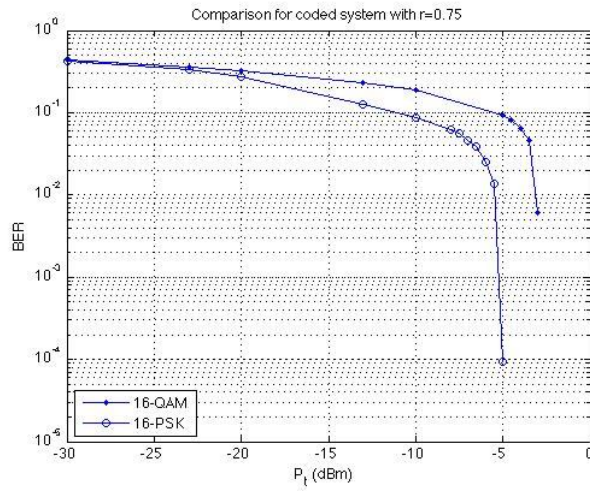
Εικόνα 30: Επιδόσεις για σύστημα για σύστημα CC με ρυθμό 2/3, $M=64$.

7.3 Υβριδικοί κώδικες (LDPC+BCH)

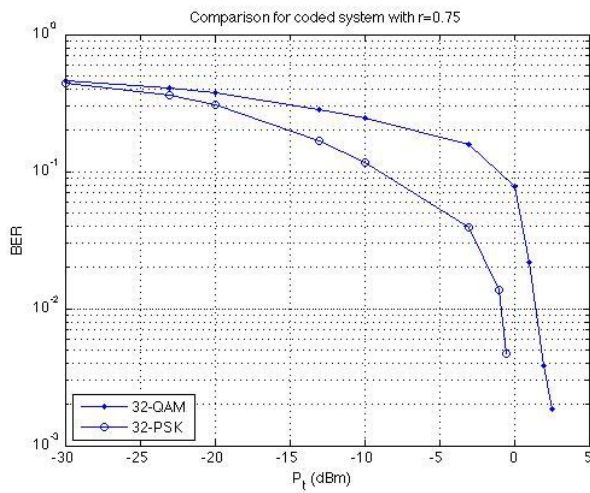
Ακολούθως εκτελέστηκε ένα σύνολο συγκριτικών πειραμάτων όπου χρησιμοποιήθηκαν LDPC κώδικες σε συνδιασμό με BCH κωδικοποίηση όπως ακριβώς και στο πρότυπο DVB-S2 αλλά για τα σχήματα διαμόρφωσης PSK και QAM και τάξη διαμόρφωσης $M > 8$ καθώς όπως προκύπτει από τα προηγούμενα αποτελέσματα αυτά είναι τα 2 επικρατέστερα. Από τα αποτελέσματα φαίνεται ότι διατηρείται το προβάδισμα για τις τάξεις $M=8, 16$ και 32 του PSK ενώ για $M=64$ του QAM που εμφανίζεται παρουσία συνελκτικών κωδικών παρά το ότι άλλαξε το σχήμα κωδικοποίησης. Ενδιαφέρον όμως έχει το ότι η μεταδιδόμενη ισχύς έχει μειωθεί κατά πολύ. Είναι μάλιστα εμφανές, από το τελευταίο σχήμα, ότι εμφανίζεται το αναμενόμενο φαινόμενο του καταράχτη κατά την εφαρμογή του LDPC κώδικα βελτιώνοντας κατά 6dBm περίπου την ποιότητα της επικοινωνίας με χρήση κώδικα.



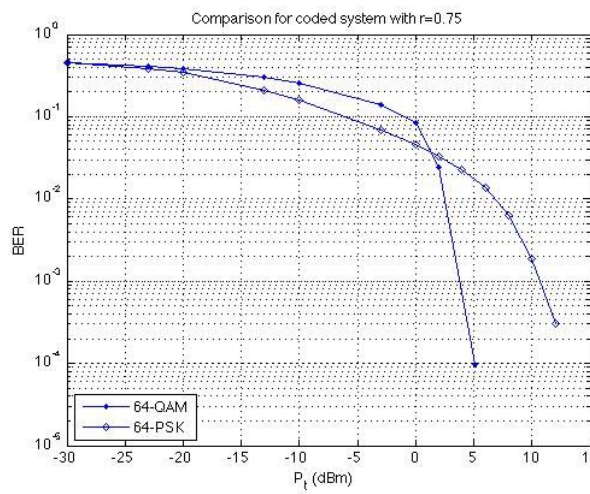
Εικόνα 31: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=8$.



Εικόνα 32: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=16$.



Εικόνα 33: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=32$.



Εικόνα 34: Επιδόσεις για σύστημα για σύστημα LDPC με ρυθμό 3/4, $M=64$.

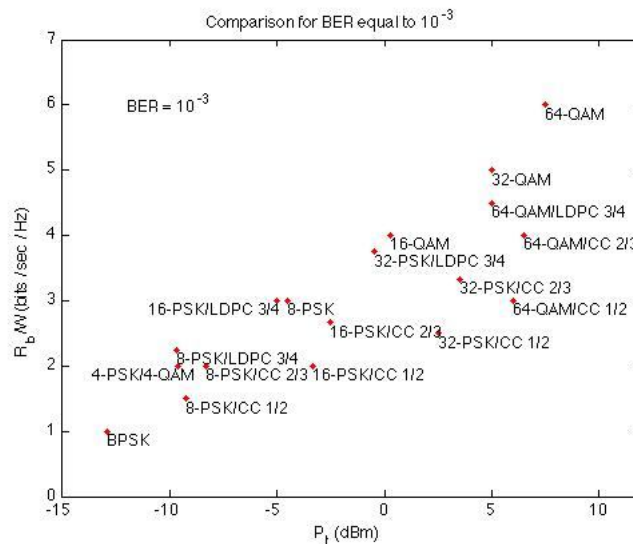
8. Συμπεράσματα

Ο Πίνακας 4 όπως έχουμε αναφέρει παρουσιάζει το σχήμα διαμόρφωσης που επικρατεί ανάλογα με την τάξη M του αλφαβήτου και την κατηγορία που μελετήθηκε δηλ. χωρίς κώδικες διόρθωσης σφάλματος, με συνελκτικούς κώδικες ή με υβριδικούς κώδικες. Προκειμένου όμως να είναι σαφές πια είναι η προτεινόμενη διαμόρφωση παραθέτουμε και τον ακόλουθο πίνακα.

Πίνακας 5: Κυρίαρχα σχήματα διαμόρφωσης

M	Uncoded		CC (1/2)		CC (2/3)		LDPC (3/4)	
	Νικητής	P_t (dBm)	Νικητής	P_t (dBm)	Νικητής	P_t (dBm)	Νικητής	P_t (dBm)
2	BPSK	-12.9	-	-	-	-	-	-
4	PSK/QAM	-9.6	-	-	-	-	-	-
8	PSK	-4.5	PSK	-9.23	PSK	-8.33	PSK	-9.7
16	QAM	0.3	PSK	-3.33	PSK	-2.5	PSK	-5.0
32	QAM	5.0	PSK	2.5	PSK	3.5	PSK	-0.5
64	QAM	7.5	QAM	6.0	QAM	6.5	QAM	5.0

Στην Εικόνα 35 απεικονίζεται μία σύγκριση των διαφόρων μεθόδων διαμόρφωσης με βάση τα αποτελέσματα του τελευταίου πίνακα και σε σχέση με το R_b/W που εκφράζει την απόδοση του εύρους ζώνης ως προς την ενέργεια του laser που χρειάστηκε για την μετάδοση με $BER=10^{-3}$.



Εικόνα 35: Σύγκριση διαφόρων μεθόδων διαμόρφωσης για πιθανότητα σφάλματος 10^{-3} .

Με βάση τα αποτελέσματα προκύπτει ότι:

- Η PSK διαμόρφωση παρουσιάζεται επικρατέστερη για τάξεις αλφαβήτου μικρότερες ή ίσες του 32.
- Στην περίπτωση που το αλφάβητο είναι τάξης μεγαλύτερης από 32 φαίνεται ότι πρέπει να χρησιμοποιηθεί QAM διαμόρφωση.
- Αν χρησιμοποιηθούν CC τότε η βελτίωση που επιτυγχάνεται στο BER για σταθερή ισχύ P_t πέφτει καθώς αυξάνεται ο ρυθμός κωδικοποίησης R .

- Επικρατέστερη μορφή κωδικοποίησης είναι αυτή με χρήση υβριδικών κωδίκων LDPC+ BCH του προτύπου DVB-S2 όπως αναμενόταν με την οποία μπορεί να έχει κανείς κέρδος στην ισχύ P_t από 5 dBm (για την περίπτωση του 8-PSK) μέχρι 2.5 dBm (για την περίπτωση του 64-QAM).

Από αυτά γίνεται ολοφάνερο ότι είναι εφικτή και στην περίπτωση των οπτικών επικοινωνιών η δημιουργία ενός σχήματος προσαρμοστικής επιλογής Διαμόρφωσης/Κωδικοποίησης (adaptive coding and modulation) με βάση το οποίο μπορεί να επιλέγεται ο κατάλληλος συνδιασμός Διαμόρφωσης/Κωδικοποίησης κάθε φορά ανάλογα με το διάυλο επικοινωνίας που παρεμβάλλεται ανάμεσα στον πομπό και στον δέκτη όπως και στις ράδιο-επικοινωνίες.

Το επόμενο βήμα είναι να συγκριθεί ή ποιότητα της επικοινωνίας στην περίπτωση που υπάρχει θόρυβος φάσης κατά την μετάδοση και διάφοροι άλλοι θόρυβοι λόγω του laser.

Αναφορές

- [1] G. E. Corazza *Digital Satellite Communications*, 2007 :Springer Science plus Business Media, LLC
- [2] John R. Barry “Wireless Infrared Communications”, Kluwer Academic Publishers, 1994
- [3] «Λεπτομέρειες μοντελοποίησης οπτικού σύμφωνου καναλιού με εργαλεία ανοιχτού λογισμικού», παραδοτέο Π1.1. του προγράμματος «Σύμφωνα Ασύρματα Οπτικά Συστήματα ως Δομικά Στοιχεία για την Υλοποίηση Δικτύων Οικίας και Επαγγελματικών Χώρων Επόμενης Γενιάς, (COWS)», διαθέσιμο ηλεκτρονικά στην διεύθυνση:
- [4] «Εργαλείο προσομοίωσης καναλιού και διαύλου», παραδοτέο Π1.2. του προγράμματος «Σύμφωνα Ασύρματα Οπτικά Συστήματα ως Δομικά Στοιχεία για την Υλοποίηση Δικτύων Οικίας και Επαγγελματικών Χώρων Επόμενης Γενιάς, (COWS)», διαθέσιμο ηλεκτρονικά στην διεύθυνση:
- [5] Herzog et al, “An optical phase locked loop for coherent space communications”, PhD thesis, Swiss Federal Institute of Technology, Zurich, diss. no. 16384 (2006).
- [6] Ly-Gagnon, D.; Tsukamoto, S.; Katoh, K.; Kikuchi, K., "Coherent detection of optical quadrature phase-shift keying signals with carrier phase estimation," *Lightwave Technology, Journal of*, vol.24, no.1, pp.12,21, Jan. 2006
- [7] R. Noe, “Phase noise tolerant synchronous QPSK receiver concept with digital I&Q baseband processing,” in *Proc. Opto-Electronics and Communications Conf. (OECC)*, Yokohama, Japan, Jul. 12–16, 2004, pp. 818–819.
- [8] ETSI. Digital Video Broadcasting (DVB): □Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications, DVB BlueBook A83-2, ETSI March 2014. URL <http://www.etsi.org>.
- [9] ETSI. Digital Video Broadcasting (DVB): Second Generation DVB Interactive Satellite System (DVB-RCS2) Part 2: Lower Layers for Satellite standard. Technical Report EN 101 545-2 v1.1.1, ETSI May 2012. URL <http://www.etsi.org>.
- [10] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001. □
- [11] S. Papaharalabos, D. Benmayor, P. T. Mathiopoulos, and P. Fan, “Performance comparisons and improvements of channel coding techniques for digital satellite broadcasting to mobile users,” *IEEE Trans. Broadcast.*, vol. 57, no. 1, pp. 94–102, Mar. 2011.
- [12] A. Leven, N. Kaneda, U-V Koc, and Y.-K. Chen “Frequency Estimation in Intradyne Reception,” *IEEE Photon. Technol. Lett.* 19, 366–368 (2007)
- [13] E. Ip and J. Kahn, "Feedforward Carrier Recovery for Coherent Optical Communications," *J. Lightwave Technol.* 25, 2675-2692 (2007).
- [14] Gitlin, Richard D., Jeremiah F. Hayes, and Stephen B. Weinstein, *Data Communications Principles*, New York, Plenum, 1992.
- [15] Proakis J. and Salehi M., (μετάφραση Καρούπαλος Κ., Ζέρβας Ε., Καραμπογιάνης Σ. και Σαγκριώτης Ε.), *Συστήματα Τηλεπικοινωνιών (μεταφρασμένο)*, Εκδόσεις Εθνικό & Καποδιστριακό Πανεπιστήμιο Αθηνών, 2002.

Παράρτημα

```

%% Main script file for COWS based simulation
% BCH and LDPC codes are implemented.
% Only normal FECFRAMES are considered (64800 bits).
% COWS_v001.m

% matlabpool('open',4);

global sp

%% Comment or uncomment the components you desire.
% Initialization
% Clear memory
clear all;
% Close all figures
close all;
% Enable pause
pause on;
% Start timer
datestr(now),tic

%% Plot Flags
% Flag controlling filter visualization tool
sp.flags.fvtool = 0;
% Flag controlling EYE plots. Not yet checked
sp.flags.eye = 0; % After receiver
% Flags controlling SCATTER plots
sp.flags.scatter = 0;
sp.flags.scHPAo = 1;
sp.flags.scHPAi = 0;
sp.flags.scNoise = 0;
sp.flags.scEst = 1;
sp.flags.scDem = 0;
sp.flags.sc_before_PAPR = 0;
sp.flags.sc_after_PAPR = 0;
% Flags controlling filtered signal plots
% Only for demonstration. Not fully implemented, it causes error floors.
sp.flags.fsignal = 0;
% Flags controlling SPECTRUM plots
sp.flags.spectrum = 0;
sp.flags.baseband = 0;
sp.flags.passband = 0; % TBF
sp.flags.afterSyn = 0;
sp.flags.spNoise = 0;
sp.flags.spHPAi = 1;
sp.flags.spHPAo = 1;
sp.flags.spOneI = 0;
sp.flags.spHPApI = 0;
% Flags for debugging
sp.flags.debug = 0;
sp.flags.debug_mod = 0;
sp.flags.debug_demod = 0;
% Flag controlling WAITBAR
sp.flags.waitbar = 0;
% Flag retaining scatter plots and spectrum plots
sp.flags.closeall = 0;

%% Plot strings to display different line styles
plt_point = ['. ' ; 'o ' ; 'x ' ; '+' ; 's ' ; 'd ' ; 'v '];
plt_color = ['b' ; 'r' ; 'k' ; 'g' ; 'm' ];
sp.plt = [strcat(plt_point, plt_color(1), '-');...
          strcat(plt_point, plt_color(2), '-');...
          strcat(plt_point, plt_color(3), '-');...
          strcat(plt_point, plt_color(4), '-)];

%% Packet oriented DVB-S2 simulation
sp.maxNumPackets = 10^7; % not implemented
%% Maximum number of bits during simulation
sp.maxNumBits = 10^6;
%% Maximum number of errors for statistical reasons
sp.maxNumErrs = 100;

```

```

%% Number of symbols in a basic frame (loop iteration)
sp.base = 1024;
%% Transmission parameters
% Carrier frequency (Hz)
sp.Fc = 2*1e6;
% Symbol rate (symbols/second or bauds) = 1/Tsym, Tsym is the signal's symbol period
sp.Rsym = 1e9;
% Number of samples per symbol
sp.nSamps = 32;
% Calculate sampling frequency in Hz = 1/Tsamp, Tsamp is the signal's sampling period
sp.Fs = sp.Rsym * sp.nSamps;
%% Laser characteristics
% Flag to consider RF to Optical transmission
sp.flags.cows = 1;
% Laser Transmit Power Range
if sp.flags.cows
%   sp.laser.Type = 'Pt';
%   sp.laser.Pt_range = [0.001e-3 0.005e-3 0.01e-3 0.05e-3 0.06e-3 0.07e-3 0.08e-3 0.099e-3];
    sp.laser.Type = 'LW';
    sp.laser.BL_range = [1e5 5e5:5e5:40e6];
end

%% Flag for testing or production code
sp.TEST = 1;

%% Select a modulation type.
% sp.mod.Type = 'RHOMBIC'; % RHOMBIC modified settings
sp.mod.Type = 'DPSK'; % M-DPSK settings
% sp.mod.Type = 'QAM'; % M-QAM settings

%% Select constellation order and related parameters
% For CPM it is tested for values 2,4,8. Not working for the QAM if the order value equals 2.
sp.mod.Order = 2;
% Define bits/symbol
sp.mod.k = log2(sp.mod.Order);
% Maximum number of symbols during simulation
sp.maxNumSymbols = sp.maxNumBits / sp.mod.k;

%% Implement Gray coding or not
sp.flags.gray = 0;

%% Only for CPM like modulations
% sp.mod.Type = 'CPM'; % CPM settings
% sp.mod.Type = '2AMSK'; % 2AMSK settings
% sp.flags.flt2amsk = 1;
% sp.flags.simple = 1;
if (strcmpi(sp.mod.Type, 'cpm') || strcmpi(sp.mod.Type, '2amsk'))
% Modulation index
    sp.mod.idx = 1/2;
% Multi-amplitude CPM flag. Not sure if it is working any more
    sp.flags.MACPM = 0; % Receiver is not implemented TBF
    if sp.flags.MACPM
        sp.mod.Opr = '2-2'; % Type settings for order per ratio (Opr)
        sp.mod.Epr = '1-2'; % Type settings for energy per ratio
    end
    sp.mod.ph0 = 0; % Initial phase
    sp.mod.VD = 1; % Viterbi Traceback Depth
    sp.mod.PL = 1; % Pulse Length for the filter appearance of the CPM
end

%% Flag determining the use of codes. Select code parameters.
sp.flags.coding = 0;
% Code parameters
if sp.flags.coding
%   sp.code.Type = 'cc'; % Stands for convolutional codes
    sp.code.Type = 'ldpc'; % Stands for convolutional codes
    if ( strcmpi(sp.code.Type, 'cc') )
        sp.code.t = poly2trellis(3, [5 7]); % Trellis
        sp.code.R = 1/2; % Code rate
        sp.code.d = 48; % Viterbi depth
        sp.SPI = sp.base; % Number of symbols in a frame (loop iteration)
        sp.BPI = sp.SPI * 2 * sp.mod.k; % Number of bits per iteration (BPI)
        code = sp.code %#ok<*NOPTS> % Display code parameters in command window
    else if ( strcmpi(sp.code.Type, 'ldpc') )
        sp.flags.minterleavers = 0; % flag for multiple interleavers
        sp.flags.bch = 0;
        % Only LDPC codes have been implemented for normal FECFRAME of the DVBS2
    end
end

```

```

% A .ldpc was added to help future implementations of BCH codes or other
% types of codes
sp.code.ldpc.R = 3/4; % code rate
sp.code.R = sp.code.ldpc.R;
sp.code.ldpc.H = dvbs2ldpc(sp.code.ldpc.R);
if sp.flags.bch
    bch_objects;
end
sp.code.ldpc.enc = fec.ldpcenc(sp.code.ldpc.H);
sp.code.ldpc.dec = fec.ldpcdec(sp.code.ldpc.H);
sp.code.ldpc.dec.DecisionType = 'Hard decision';
sp.code.ldpc.dec.OutputFormat = 'Information part';
sp.code.ldpc.dec.NumIterations = 50;
sp.code.ldpc.dec.DoParityChecks = 'Yes';
if sp.flags.bch
    sp.BPI = kBCH; % Number of bits per iteration (frame) with bch codes
else
    sp.BPI = sp.code.ldpc.enc.NumInfoBits; % Number of bits per iteration (frame)
without bch codes
end
sp.SPI = sp.BPI / sp.mod.k; % Number of symbols in a frame
code = sp.code %#ok<*NOPTS> % Display code parameters in command window
end
end
else
    sp.SPI = sp.base; % Number of symbols in a frame (loop iteration)
    sp.BPI = sp.SPI * sp.mod.k; % Number of bits per iteration (BPI)
end

modulation = sp.mod % Display modulation parameters in command window

%% Flag determining OFDM multiple access
sp.flags.OFDM = 0;
if sp.flags.OFDM
    sp.MA.FFTL = 64; % Length of the FFT
    if (~sp.flags.coding)
        sp.SPI = sp.base * sp.MA.FFTL; % Number of symbols in a frame
        sp.BPI = sp.SPI * sp.mod.k;
    end
end

%% Flag determining multiple access
sp.flags.MA = 0;
% sp.MA.Type = 'OFDMA'; % OFDMA multiple access
% sp.MA.Type = 'IFDMA'; % IFDMA multiple access
sp.MA.Type = 'LFDMA'; % LFDMA multiple access
if sp.flags.MA
    if sp.flags.OFDM
        error('VD: You cannot have OFDM and MA techniques together! Check sp.flags.OFDM value!')
    else
        sp.MA.FFTL = 16; % Length of the FFT
        sp.MA.Mu = 128; % IFFT size for service link (user Uplink to satellite)
    end
    if (~sp.flags.coding)
        sp.SPI = sp.base * sp.MA.FFTL; % Number of symbols in a frame
        sp.BPI = sp.SPI * sp.mod.k;
    end
end

%% Flag determining prefix type
sp.flags.prefix = 0;
if sp.flags.prefix
    if ~(sp.flags.MA || sp.flags.OFDM)
        error('VD: You cannot have prefixing enabled without OFDM or MA technique!')
    else
        % Uncomment/comment next line to add/remove a CP with a CPL length and an extra window of
        (EW/2+EW/2) symbols
        % sp.prefix.Type = 'gp';
        % Uncomment/comment next line to add/remove a zero padding (ZP)
        % sp.prefix.Type = 'zp';
        % Uncomment/comment next line to add/remove a cyclic prefix (CP)
        sp.prefix.Type = 'cp';
        % Uncomment/comment next line to add/remove a CP with a CPL length and an extra window of
        (EW/2+EW/2) symbols
        if strcmpi( sp.prefix.Type, 'gp' )
            sp.prefix.EW = 8;
            sp.prefix.CPLength = 8;
        end
    end
end

```

```

        sp.prefix.Length = sp.prefix.CPLength + sp.prefix.EW;
    else
        sp.prefix.CPLength = 16;
        sp.prefix.Length = sp.prefix.CPLength;
    end
    % In case of gp comment previous line and uncomment next one
    if sp.flags.OFDM
        sp.prefix.rate = sp.MA.FFTL/(sp.prefix.Length+sp.MA.FFTL); %TBF
    elseif sp.flags.MA
        sp.prefix.rate = sp.MA.Mu/(sp.prefix.Length+sp.MA.Mu); %TBF
    end
end
end

%% Flag determining pulse shaping. If 1 then select a filter.
sp.flags.windowing = 0; %
if sp.flags.windowing
    % Select a rectangular window
    %     sp.wnd.Type = 'rect';
    % Select oversampling rate
    % sp.wnd.Type = 'normal' % not implemented!
    %     sp.wnd.Type = 'rc'; % not implemented!
    sp.wnd.Type = 'sqr';
    %     sp.wnd.Type = 'gaus';
    %     if ~(strcmpi(sp.mod.Type,'cpm') || strcmpi(sp.mod.Type,'2amsk'))
    if ~(strcmpi(sp.mod.Type,'cpm'))
        switch sp.wnd.Type
            case 'rc'
                sp.wnd.Ts = 1/sp.Rsym;
                sp.wnd.fR = 8; % filter rate or filter length in symbols fR+1
                sp.wnd.rolloff = .35; % rolloff factor
                sp.wnd.gdelay = 0; % Group delay
                Fs = sp.wnd.fR; % Length of the filter. Be careful it is not the sp.Fs
                R = sp.wnd.rolloff;
                filterSpec = fdesign.pulse shaping(sp.nSamps, 'Raised Cosine', ...
                    'Nsym,Beta', Fs, R);
                sp.wnd.hXmtFilt = design(filterSpec);
                % Make a copy of the transmitter filter and use it as matched filter.
                sp.wnd.hRcvFilt = copy(sp.wnd.hXmtFilt);
            case 'sqr'
                sp.wnd.Ts = 1/sp.Rsym;
                sp.wnd.fR = 8; % filter rate or filter length in symbols
                sp.wnd.rolloff = 0.35; % rolloff factor
                sp.wnd.gdelay = 0; % Group delay
                Fs = sp.wnd.fR; % Length of the filter. Be careful it is not the sp.Fs
                R = sp.wnd.rolloff;
                filterSpec = fdesign.pulse shaping(sp.nSamps, 'Square root raised cosine', ...
                    'Nsym,Beta', Fs, R);
                sp.wnd.hXmtFilt = design(filterSpec);
                % Make a copy of the transmitter filter and use it as matched filter.
                sp.wnd.hRcvFilt = copy(sp.wnd.hXmtFilt);
            case 'gaus'
                sp.wnd.Ts = 1/sp.Rsym;
                Ts = sp.wnd.Ts;
                sp.wnd.BT = 0.25; % 3-dB bandwidth-symbol time
                B = sp.wnd.BT * sp.Rsym;
                OF = sp.nSamps; % Oversampling factor
                % Number of symbol periods between beginning and peak of impulse response
                sp.wnd.fR = 5; % Symbol periods to the filters peak
                sp.wnd.L = 2 * OF * sp.wnd.fR + 1; % filter length in samples
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %     sp.wnd.hXmtFilt.Numerator = gaussfir(sp.wnd.BT, sp.wnd.fR, OF);
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                tf = linspace(-sp.wnd.fR*Ts, sp.wnd.fR*Ts, sp.wnd.L)';
                sp.wnd.hXmtFilt.Numerator = (1/(2*Ts)) * ( qfunc( 2*pi*B*(tf-0.5*Ts)/sqrt(log(2)) ) -
                    ...
                    qfunc( 2*pi*B*(tf+0.5*Ts)/sqrt(log(2)) ) );
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                sp.wnd.hRcvFilt = sp.wnd.hXmtFilt;
            end
        % Normalize the filter coefficients so that the filtered and unfiltered data matches when
        % overlaid.
        normFact = max(sp.wnd.hXmtFilt.Numerator);
        sp.wnd.hXmtFilt.Numerator = sp.wnd.hXmtFilt.Numerator / normFact;
        if ( strcmpi(sp.mod.Type,'2amsk') )
            if sp.flags.flt2amsk
                if sp.flags.simple

```

```

        sp.wnd.hRcvFlt.Numerator = sp.wnd.hRcvFlt.Numerator / sp.nSamps;
    end
end
else
    sp.wnd.hRcvFlt.Numerator = sp.wnd.hRcvFlt.Numerator * normFact * sp.nSamps;
end
else
    % Parameters for the filter of CPM modulation.
    % Not yet fully implemented. File dependency with my_s_mod.m

    %         sp.wnd.Type = 'Raised Cosine';
    %         sp.wnd.mode = 'LRC';
    %         sp.wnd.params = [];

    %         sp.wnd.Type = 'Tamed FM';
    %         sp.wnd.mode = 'TFM';
    %         sp.wnd.params = [];

    %         sp.wnd.Type = 'Spectral Raised Cosine';
    %         sp.wnd.mode = 'LSRC';
    %         sp.wnd.params.mainLobePulseLength = 1;
    %         RollOff = 0.3;
    %         sp.wnd.params.beta = RollOff;

    %         sp.wnd.Type = 'Gaussian';
    %         sp.wnd.mode = 'GMSK';
    %         sp.wnd.params.BT = 50; % 3-dB bandwidth-symbol time
    %         B = sp.wnd.params.BT * sp.Rsym;

    sp.wnd.Type = 'Rectangular';
    sp.wnd.mode = 'LREC';
    sp.wnd.params = [];

    sp.wnd.Ts = 1/sp.Rsym;
    Ts = sp.wnd.Ts;
    % Number of symbol periods between beginning and peak of impulse response
    sp.wnd.fR = sp.mod.PL; % Length of the filter in symbol periods. Be careful it is not the
sp.Fs
    sp.wnd.fL = sp.nSamps * sp.wnd.fR + 1; % filter length in samples
    % --- Compute the required sample times, widths and phase response functions
    [ecode, emsg, sfunParams] = cpmmmodparams_check(sp.wnd.mode, sp.wnd.params, sp.mod.PL,
sp.nSamps);
    params.g = [sfunParams.g;0]*sp.mod.idx*2*pi;
    % --- Reshuffle the filter into phase order for use by the polyphase filter
    [params.gMod, notUsed]= localUpFir2_check(params.g,sp.nSamps);
    hfilter = fvtool(params.g, 'phase');
    sp.wnd.hXmtFlt.Numerator = hfilter.Filters{1}.Numerator;
    close(hfilter);
end
window = sp.wnd % Display filtering parameters in command window
end

%% Flag determining PAPR reduction type
if sp.flags.OFDM
    sp.flags.PAPR = 0;
    if sp.flags.PAPR
        sp.PAPR.Type = 'DRM'; % Digital Reduction Method (DRM)
        sp.PAPR.Parameters = .25; % Parameter for DRM
        %         sp.PAPR.Type = 'ACM'; % Amplitude Clipping Method (ACM)
        %         sp.PAPR.Parameters = .1; % Parameter for ACM
        %         sp.PAPR.Type = 'TIM'; % Tone Injection Method (TIM)
    end
end

%% Flag determining non linearity type
sp.flags.NL = 0;
if sp.flags.NL
    % Uncomment/comment to use a NL model
    %         sp.NL.Type = 'Saleh';
    %         sp.NL.Type = 'ESA_linearized'1.0; % Intelsat
    %         sp.NL.Type = 'ESA';
    sp.NL.Type = 'Rapp';
    %         sp.NL.Type = 'HL' % Hard limiter not fully implemented
    sp.NL.IBO = 0; % Select Input Backoff level in dB
    %         sp.NL.DTH = pi/36; % Select delta th for the second SSPA in rads e.g. pi = 180 degrees
    NL = sp.NL % Display nonlinearity parameters in command window
    %% Flag for NL error recovery by scaling (not required if predistort)

```

```

    sp.flags.NLcorrect = 1;
    %% Flag for predistortion
    sp.flags.NLpred = 0;
    if sp.flags.NLpred
        sp.NL.pred.symbols = sp.mod.Order * 100; % W symbols
        sp.NL.pred.blocks = 200; % S blocks
    end
end

%% Flag for PLFRAME based on DVBS2 including pilots
sp.flags.PLframing = 0;
sp.dvbs2pilots = 396; % This value results from the standard

%% Flag for SMF involvement as RdG paper (No predistortion is required)
sp.flags.pilots = 0;
% sp.pilots = sp.mod.Order * 10; % 100 symbols per constellation point
sp.pilots = 100; % 100 symbols per constellation point

%% Flag determining noise addition
sp.flags.noise = 0;
% shows the results as a function of Es/No instead of Eb/No
sp.flags.EsNodB = 0; % TBF for CPM
if sp.flags.noise
    % Bit energy per No given in dB
    if sp.flags.coding
        EbNodB = [0:0.5:10]; % fliplr(8.5 - sqrt(logspace(-2,1,20)));
    else
        EbNodB = 0:14;
    end
end
else
    disp('VD: Noiseless mode!');
    if sp.flags.coding
        EbNodB = 10;
    else
        EbNodB = 100; % Tend to noise-less
    end
end
end
%% SIMULATION PARAMETERS
% Stop the simulation loop if there are estimated sufficient errors
sp.flags.fast = 1;

% BER estimation on/off
sp.flags.ber = 1;

% SER estimation on/off
if sp.flags.coding
    sp.flags.ser = 0;
else
    sp.flags.ser = 0;
end

% PER estimation on/off (for DVBS2 1 packet is 188bytes)
sp.flags.per = 0;

% Save figures on/off
sp.save_figures = 0;

% Theoretical error bound implemented only for 16-APSK on/off
sp.flags.analytic = 0;
if sp.flags.coding && sp.flags.analytic
    error('VD: sp.flags.coding should be set to 0!')
end

%% Execute an external script to create a FILENAME
my_file

%% SIMULATION
% The first time that script_loop is executed this run index is set to 1.
% script_loop increases this index in order to separate the results. This
% index is used for automatically setting the legends during the plot.
sp.irun = 1;
% profile on
sp.flags.aci = 0;
sp.ft = 2;
script_loop_04;
% [lut_obo] = predistortion_s(sp)
% profile viewer

```

```
%% Stop timer
toc

%% QAM
% sp.mod.Type = 'QAM'; % M-QAM settings
% sp.mod.Order = 16;
% sp.mod = rmfield(sp.mod,'ppr');
% sp.mod = rmfield(sp.mod,'gamma');
%
% script_loop_04
% dvbs2_td;

%% TBF
% EsNo should be fixed for CPM kind formats
% Current results may be misleading

clear sp.code
sp.laser.Pt_range = [0.001e-3 0.005e-3 0.01e-3 0.05e-3 0.06e-3 0.07e-3 0.075e-3 0.1e-03 0.5e-03 1e-
03 5e-03];
sp.code.Type = 'cc'; % Stands for convolutional codes
sp.code.t = poly2trellis(3,[5 7]); % Trellis
sp.code.R = 1/2; % Code rate
sp.code.d = 48; % Viterbi depth
sp.SPI = sp.base; % Number of symbols in a frame (loop iteration)
sp.BPI = sp.SPI * 2 * sp.mod.k; % Number of bits per iteration (BPI)
code = sp.code %#ok<*NOPTS> % Display code parameters in command window
script_loop_04;

clear sp.code
sp.flags.coding = 0;
script_loop_04;
```



```
% (c) 2014 V. Dalakas
% MY_FILE.m

if strcmp(sp.mod.Type,'2AMSK')
    filetitle = strcat('RL_',sp.mod.Type);
elseif strcmp(sp.mod.Type,'TEST2')
    filetitle = strcat('RL_',sp.mod.ppr);
elseif strcmp(sp.mod.Type,'2QPSK')
    filetitle = strcat('RL_',sp.mod.Type);
else
    filetitle = strcat('RL_',num2str(sp.mod.Order),'_',sp.mod.Type);
end
if (sp.flags.OFDM)
    filetitle = strcat(filetitle,'_N_',num2str(sp.MA.FFTL));
end
if (sp.flags.MA)
    filetitle = strcat(filetitle,'_N_',num2str(sp.MA.FFTL));
    filetitle = strcat(filetitle,'_M_',num2str(sp.MA.Mu));
end
if (sp.flags.windowing)
    filetitle = strcat(filetitle,'_',sp.wnd.Type);
    if strcmp(sp.wnd.Type,'sqrc')
        filetitle = strcat(filetitle,'_wnd_rate_',num2str(sp.wnd.rolloff));
    end
end
if (sp.flags.NL)
    filetitle = strcat(filetitle,'_',sp.NL.Type,'_IBO_',num2str(sp.NL.IBO));
end
if (sp.flags.coding)
    if strcmp(sp.code.Type,'cc')
        filetitle = strcat(filetitle,'_R_',num2str(sp.code.R));
    else
        filetitle = strcat(filetitle,'_R_',num2str(sp.code.ldpc.R));
    end
end
if isfield(sp.mod,'gamma1')
    filetitle = strcat(filetitle,'_g1_',num2str(sp.mod.gamma1));
end
if isfield(sp.mod,'gamma2')
    filetitle = strcat(filetitle,'_g2_',num2str(sp.mod.gamma2));
end

sp.filename = filetitle;
filetitle
```

```

%%
% script_loop_04.m bind with cows_v001.m and RL20.m
%
% This script executes the loop for ber Monte Carlo simulation and plots
% the results if it is desired. It supposes prior execution of the
% script_TD.m file
%
% (c) 2014 V. Dalakas
%
% variable output arguments are stored in a structure called exodos
% that has the results of the RL function

clear exodos
% initialize outputs
nout = 0;
% initialize parameters for the outputs
x = []; % vector for the x-axis in BER, SER or PER figures
ber = []; % BER values
ser = []; % SER values
per = []; % PER values

if sp.flags.ber
    % output index is increased
    nout = nout + 1;
    % figure index takes the same index
    fber = nout;
end

if sp.flags.ser
    % output index is increased
    nout = nout + 1;
    % figure index takes the same index
    fser = nout;
end

if sp.flags.per
    % output index is increased
    nout = nout + 1;
    % figure index takes the same index
    fper = nout;
end

% the experiment considers nonlinearity
if isfield(sp,'NL')
    % output index is increased
    nout = nout + 1;
    % execute static predistortion only on the first simulation run
    if sp.flags.NLpred
        sp.flags.NLpredon = 1;
    end
end

% the experiment considers noise with EsNo in dB instead of EbNo
if sp.flags.noise && sp.flags.EsNodB
    nout = nout + 1;
end

%% Simulation
if sp.flags.noise
    sp.EbNodB = [];
    XX = repmat(sp,1,length(EbNodB));
    % the experiment will be executed so many times as the EbNodB vector
    for cnt = 1:length(EbNodB)
        %% statistical credibility check. comment next if for parfor
        %         if (cnt > 1) %&& (sp.flags.fast)
        %             % if the last estimated value does not obtain the desired
        %             % accuracy level the experiment ends and the value is deleted
        %             if min(ber(cnt-1)) < (sp.maxNumErrs/sp.maxNumBits)
        %                 % message
        %                 disp('VD: Simulation ended!')
        %                 % delete last value
        %                 x(end) = [];
        %                 if sp.flags.ber
        %                     ber(end) = [];
        %                 end
        %                 if sp.flags.ser

```

```

%             ser(end) = [];
%             end
%             if sp.flags.per
%                 per(end) = [];
%             end
%             break;
%         end
%     end

%% set the EbNodB for the simulation run. The loop depends from this.
XX(cnt).EbNodB = EbNodB(cnt);
% this is maybe obsolete since the code is still changing
if XX(cnt).TEST
    exodos(cnt) = RL20(XX(cnt));
else
    exodos(cnt) = DVBS2(XX(cnt)); % TBF
end
nout = 1;
if XX(cnt).flags.ber
    ber(cnt) = exodos(cnt).BER;
    nout = nout + 1;
end
if XX(cnt).flags.ser
    ser(cnt) = exodos(cnt).SER;
    nout = nout + 1;
end
if XX(cnt).flags.per
    per(cnt) = exodos(cnt).PER;
    nout = nout + 1;
end
if isfield(XX(cnt), 'NL')
    OBO(cnt) = exodos(cnt).OBO;
    nout = nout + 1;
end
if XX(cnt).flags.EsNodB
    EsNodB = exodos(cnt).EsNodB;
    nout = nout + 1;
    % x-vector contains EsNo in dB
    x = [x EsNodB];
else
    % x-vector contains EbNo in dB
    x = [x XX(cnt).EbNodB];
end
end
else
if sp.flags.cows
    switch sp.laser.Type
        case 'Pt'
            sp.laser.BL = 40e6;
            sp.EbNodB = EbNodB;
            sp.laser.Pt = [];
            XX = repmat(sp,1,length(sp.laser.Pt_range));
            for cnt = 1:length(sp.laser.Pt_range)
                XX(cnt).laser.Pt = sp.laser.Pt_range(cnt);
                exodos(cnt) = RL20(XX(cnt));
                nout = 1;
                if XX(cnt).flags.ber
                    ber(cnt) = exodos(cnt).BER;
                    nout = nout + 1;
                end
                if XX(cnt).flags.ser
                    ser(cnt) = exodos(cnt).SER;
                    nout = nout + 1;
                end
                if XX(cnt).flags.per
                    per(cnt) = exodos(cnt).PER;
                    nout = nout + 1;
                end
                % x-vector contains Pt
                x = [x 10*log10(XX(cnt).laser.Pt * 1e3)];
            end
        case 'LW'
            sp.laser.Pt = 0.1e-3;
            sp.EbNodB = EbNodB;
            sp.laser.BL = [];
            XX = repmat(sp,1,length(sp.laser.BL_range));
            for cnt = 1:length(sp.laser.BL_range)

```

```

        XX(cnt).laser.BL = sp.laser.BL_range(cnt);
        exodos(cnt) = RL20(XX(cnt));
        nout = 1;
        if XX(cnt).flags.ber
            ber(cnt) = exodos(cnt).BER;
            nout = nout + 1;
        end
        if XX(cnt).flags.ser
            ser(cnt) = exodos(cnt).SER;
            nout = nout + 1;
        end
        if XX(cnt).flags.per
            per(cnt) = exodos(cnt).PER;
            nout = nout + 1;
        end
        % x-vector contains BL
        x = [x XX(cnt).laser.BL];
    end
end
else
    % without noise just for verification purposes during development
    % set the EbNodB for the simulation run, for proper execution the value
    % of EbNodB in cpm_v###.m should be large enough
    sp.EbNodB = EbNodB;
    % this is maybe obsolete since the code is still under development
    if sp.TEST
        exodos = RL20(sp);
    else
        exodos = DVBS2(sp); % TBF
    end
    % set output values
    nout = 1;
    if sp.flags.ber
        ber = exodos.BER;
        nout = nout + 1;
    end
    if sp.flags.ser
        ser = exodos.SER;
        nout = nout + 1;
    end
    if sp.flags.per
        per = exodos.PER;
        nout = nout + 1;
    end
    if isfield(sp,'NL')
        OBO = exodos.OBO;
        nout = nout + 1;
    end
end
end
end

%% This part handles the NL related plots
% init point for Total Degardation plots
if sp.irun==1
    ber_IBO = [];
end

% estimates the point for a predefined BER level with curve fitting
if isfield(sp.flags,'TD') || isfield(sp.flags,'RR')
    % different estimation procedure for steep or smooth BER curves
    if sp.flags.coding == 1
        predictedSNR = x(end);
        %%
        % [fo, gof] = fit(log10(ber'),x','cubicspline',fitoptions('cubicspline'));
        % predictedSNR = feval( fo, log10( sp.TD.level ) );
        % if predictedSNR < 0
        %     disp('Fit type changed to linear from polynomial')
        %     f = fittype('a0+a1*x');
        %     [fo, gof] = fit(log10(ber'),x',f);
        %     predictedSNR = 100*feval( fo, log10( sp.TD.level ) )
        % end
        %%
    else
        f = fittype('a0+a1*x+a2*x^2+a3*x^3');
        [fo, gof] = fit(log10(ber'),x',f);
        predictedSNR = feval( fo, log10( sp.TD.level ) );
    end
end

```

```

ber_IBO = [ber_IBO predictedSNR]
end

%% This part designs BER, SER or PER plots
if sp.flags.noise
    if sp.flags.ber
        if sp.irun==1
            %% Plot the theoretical BER
            % [BERth,SERth]=berawgn(0:length(EbNodB)-1,'qam',my_par{1,2}(2))
            % semilogy([0:length(BERth)-1],BERth,'k')
            % Plot the experimental BER
            % hold on
            % fber = gcf + 1;
            %%
            figure(fber);
            semilogy(x, ber, sp.plt(sp.irun,:));
            % see my_legend.m
            my_legend;
        else
            figure(fber);
            hold on
            semilogy(x, ber, sp.plt(sp.irun,:))
            % see my_legend.m
            my_legend;
            hold off
        end
        %% PLOT TITLE AND LEGEND TBF
        hold on
        strttitle = strcat('BER_',filetitle);
        title(strttitle,'Interpreter','none')
        ylabel('BER')
        if (sp.flags.EsNodB)
            xlabel('E_s/N_0 (dB)')
        else
            xlabel('E_b/N_0 (dB)')
        end
        grid on
        hold off

        if sp.flags.coding
            if strcmp(sp.mod.Type,'2AMSK') || strcmp(sp.mod.Type,'2QPSK')
                ftitle = strcat('Comparison for coded system with r=', num2str(sp.code.R) );
                title(ftitle);
            else
                ftitle = strcat('Comparison for coded system with r=', num2str(sp.code.ldpc.R) );
                title(ftitle);
            end
        else
            title('Comparison for uncoded system');
        end
        legend(str_legend);
        %% SAVE jpg and fig file formats
        if sp.save_figures
            filetitle = strcat('BER_',filetitle,'.jpg')
            saveas(fber,filetitle);
            filetitle(end-3:end) = '.fig'
            saveas(fber,filetitle);
        end
    end
end
if sp.flags.ser
    if sp.irun==1
        % fser = gcf + 1;
        figure(fser);
        semilogy(x, ser, sp.plt(sp.irun,:));
        % see my_legend.m
        my_legend;
    else
        figure(fser);
        hold on
        semilogy(x, ser, sp.plt(sp.irun,:))
        % see my_legend.m
        my_legend;
        hold off
    end
    %% PLOT TITLE AND LEGEND TBF
    hold on
    if sp.flags.ber

```

```

        strtitle = strcat('SER_',filetitle(5:end));
    else
        strtitle = strcat('SER_',filetitle);
    end
    title(strtitle,'Interpreter','none')
    ylabel('SER')
    if (sp.flags.EsNodB)
        xlabel('E_s/N_0 (dB)')
    else
        xlabel('E_b/N_0 (dB)')
    end
    grid on
    hold off
    if sp.flags.coding
        ftitle = strcat('Comparison for coded system with r=', num2str(sp.code.ldpc.R) );
        title(ftitle);
    else
        title('Comparison for uncoded system');
    end
    legend(str_legend);
    %% SAVE jpg and fig file formats
    if sp.save_figures
        if sp.flags.ber
            filetitle = strcat('SER_',filetitle(5:end-4),'.jpg')
        else
            filetitle = strcat('SER_',filetitle, '.jpg')
        end
        saveas(fser,filetitle);
        filetitle(end-3:end) = '.fig'
        saveas(fser,filetitle)
    end
end
if sp.flags.per
    if sp.irun==1
        figure(fper);
        semilogy(x, per, sp.plt(sp.irun,:));
        % see my_legend.m
        my_legend;
    else
        figure(fper);
        hold on
        semilogy(x, per, sp.plt(sp.irun,:))
        % see my_legend.m
        my_legend;
        hold off
    end
    %% PLOT TITLE AND LEGEND TBF
    hold on
    strtitle = strcat('PER_',filetitle);
    title(strtitle,'Interpreter','none')
    ylabel('PER')
    if (sp.flags.EsNodB)
        xlabel('E_s/N_0 (dB)')
    else
        xlabel('E_b/N_0 (dB)')
    end
    grid on
    hold off
    if sp.flags.coding
        ftitle = strcat('Comparison for coded system with r=', num2str(sp.code.ldpc.R) );
        title(ftitle);
    else
        title('Comparison for uncoded system');
    end
    legend(str_legend);
    %% SAVE jpg and fig file formats
    if sp.save_figures
        filetitle = strcat('PER_',filetitle, '.jpg')
        saveas(fper,filetitle);
        filetitle(end-3:end) = '.fig'
        saveas(fper,filetitle);
    end
end
end
end

%% This part designs BER, SER or PER plots for optical transmission
if sp.flags.cows

```

```

if sp.flags.ber
    if sp.irun==1
        figure(fber);
        semilogy(x, ber, sp.plt(sp.irun,:));
        % see my_legend.m
        my_legend;
    else
        figure(fber);
        hold on
        semilogy(x, ber, sp.plt(sp.irun,:))
        % see my_legend.m
        my_legend;
        hold off
    end
    %% PLOT TITLE AND LEGEND TBF
    hold on
    strttitle = strcat('BER_',filetitle);
    title(strttitle,'Interpreter','none')
    ylabel('BER')
    switch sp.laser.Type
        case 'Bt'
            xlabel('P_t (dBm)')
        case 'LW'
            xlabel('Line Width (Hz)')
    end
    grid on
    hold off
    if sp.flags.coding
        if strcmp(sp.code.Type,'cc')
            fttitle = strcat('Comparison for coded system with r=', num2str(sp.code.R) );
            title(fttitle);
        else
            fttitle = strcat('Comparison for coded system with r=', num2str(sp.code.ldpc.R) );
            title(fttitle);
        end
    else
        title('Comparison for uncoded system');
    end
    legend(str_legend);
    %% SAVE jpg and fig file formats
    if sp.save_figures
        filetitle = strcat('BER_',filetitle,'.jpg')
        saveas(fber,filetitle);
        filetitle(end-3:end) = '.fig'
        saveas(fber,filetitle);
    end
end
if sp.flags.ser
    if sp.irun==1
        %           fser = gcf + 1;
        figure(fser);
        semilogy(x, ser, sp.plt(sp.irun,:));
        % see my_legend.m
        my_legend;
    else
        figure(fser);
        hold on
        semilogy(x, ser, sp.plt(sp.irun,:))
        % see my_legend.m
        my_legend;
        hold off
    end
    %% PLOT TITLE AND LEGEND TBF
    hold on
    if sp.flags.ber
        strttitle = strcat('SER_',filetitle(5:end));
    else
        strttitle = strcat('SER_',filetitle);
    end
    title(strttitle,'Interpreter','none')
    ylabel('SER')
    switch sp.laser.Type
        case 'Bt'
            xlabel('P_t (dBm)')
        case 'LW'
            xlabel('Line Width (Hz)')
    end
end

```

```

grid on
hold off
if sp.flags.coding
    ftitle = strcat('Comparison for coded system with r=', num2str(sp.code.ldpc.R) );
    title(ftitle);
else
    title('Comparison for uncoded system');
end
legend(str_legend);
%% SAVE jpg and fig file formats
if sp.save_figures
    if sp.flags.ber
        filetitle = strcat('SER_',filetitle(5:end-4),'.jpg')
    else
        filetitle = strcat('SER_',filetitle, '.jpg')
    end
    saveas(fser,filetitle);
    filetitle(end-3:end) = '.fig'
    saveas(fser,filetitle)
end
end
if sp.flags.per
    if sp.irun==1
        figure(fper);
        semilogy(x, per, sp.plt(sp.irun,:));
        % see my_legend.m
        my_legend;
    else
        figure(fper);
        hold on
        semilogy(x, per, sp.plt(sp.irun,:))
        % see my_legend.m
        my_legend;
        hold off
    end
    %% PLOT TITLE AND LEGEND TBF
    hold on
    strttitle = strcat('PER_',filetitle);
    title(strttitle,'Interpreter','none')
    ylabel('PER')
    switch sp.laser.Type
        case 'Bt'
            xlabel('P_t (dBm)')
        case 'LW'
            xlabel('Line Width (Hz)')
    end
    grid on
    hold off
    if sp.flags.coding
        ftitle = strcat('Comparison for coded system with r=', num2str(sp.code.ldpc.R) );
        title(ftitle);
    else
        title('Comparison for uncoded system');
    end
    legend(str_legend);
    %% SAVE jpg and fig file formats
    if sp.save_figures
        filetitle = strcat('PER_',filetitle, '.jpg')
        saveas(fper,filetitle);
        filetitle(end-3:end) = '.fig'
        saveas(fper,filetitle);
    end
end
end

% increase the run index in case another candidate is considered
sp.irun = sp.irun + 1;

% TBF
% A function for the plots and the legends depending from BER/SER/PER

```

```

function symbols = cows( s, sp )
% File for optical DPSK physical layer in case of coherent optical
% transmission
% COWS.m

fIF = 4 * sp.Rsym;
wIF = 2 * pi * fIF;
Ts = 1 / sp.Rsym;
Ns = length(s);
Tw = Ns * Ts;
samples = sp.nSamps;
dt = Ts / samples;
Nt = Ns * samples;
dw = 2 * pi / ( dt * Nt );
R = 1.0018; % Responsivity

%-----
q = 1.6e-19;
KB = 1.38054e-23;
T = 298.15;
RSH = 5e8; % shunt resistance
B = 2*sp.mod.k/Ts; % TBC

% -----
e0 = 8.854e-12;
FOVR = pi/3;
Adet = 1e-4;
nc = 1.85; % refractive index of the concentrator

% syntetagemenes tou transmitter rs
x1=10;
y1=10;
z1=3;

% syntetagemenes tou receiver rR
x2 = 11;
y2 = 11;
z2 = 0.8;

% syntetagemenes tou anysmatos rs-rR
x3 = x1-x2;
y3 = y1-y2;
z3 = z1-z2;

DTR = sqrt(x3^2+y3^2+z3^2);
Height = z1-z2;
r = Height*tan(FOVR);
Acov = pi*(r^2);
cosx = Height / DTR;
xdif = acos(cosx);
cosy = cosx;

if xdif <= FOVR %&& sqrt(x3(i)^2 + y3(j)^2)<=r
    Aeff = (Adet*nc^2*cosx) / ((sin(FOVR))^2);
else
    Aeff = 0;
end

A = Aeff / Acov;
% -----

BL = sp.laser.BL;
% Uncomment next line to remove phase noise
% BL = 0;

Pt = sp.laser.Pt;
PLO = 10*Pt;

% Extract phases from symbols transposed vector for consistency
phase_s = angle(real(s)).';

```

```

psymb = cumsum(phase_s);

sigma=2*pi*BL*dt;
m=sqrt(sigma)*randn(1,Nt);

variance_shot= sqrt(4*q*R*(A*Pt+PLO)*B);
variance_thermal=sqrt((4*KB*T*B)/RSH);
shot = variance_shot*randn(1,Nt);
thermal = variance_thermal*randn(1,Nt);

i=0:(Nt-1);
phi0=psymb(floor(i/samples)+1);
phin=cumsum(m);
phi=phi0+phin;

current = 2*R*sqrt(A*Pt*PLO)*cos(2*pi*fIF*dt*(1:Nt)+phi) + shot + thermal;
% [A R Pt]

% PQ_n2 = 2*R^2*A*Pt^2*10;

% PQ_n1 = norm(current).^2/length(current);

if sp.flags.debug
    scatterplot(current)
    disp('Catherine's code... current')
    keyboard
end

freq=fft(current);
A1 = fftshift(freq);

w2=((1:Nt)-1-Nt/2)*dw;
f1=w2/(2*pi);

W0 = 2*pi*fIF; Bw=10e9; % specify cutoff frequency and bandwidth
[z,p,k]=buttap(3); % specify the Butterworth filter
[num,den]=zp2tf(z,p,k); % specify filter in polynomial form
[num,den]=lp2bp(num,den,W0,Bw); % convert LPF to BPF
tf(num,den); % print the transfer function of desired filter
w=w2; % specify frequency vector
H=freqs(num,den,w); % compute the frequency response

Us = times(A1,H);
ifreq=ifftshift(Us);
us = ifft(ifreq);

vs = us(samples+1:Nt);
us = us (1:Nt-samples);

w1=((1:Nt-samples)-1-(Nt-samples)/2)*dw;
f=w1/(2*pi);

v = times(vs,us);

freq=fft(v);
V = fftshift(freq);

H1 = sqrt (1./(1+(2*w1/wIF).^10));

[z2,p2,k2]=buttap(5); % fourth-order Butterworth filter
[num2,den2]=zp2tf(z2,p2,k2); % convert to polynomial form
tf(num2,den2); % print the transfer function H(s)
wc2=2*pi*fIF; % specify cutoff frequency in rad/s
[num2,den2]= lp2lp(num2,den2,wc2); % desired filter
w3=w1; % define a frequency vector
H3=freqs(num2,den2,w3); % compute frequency response
mag3=abs(H3); % compute magnitude response

Rsymb=times(V,H3);

ifreq=ifftshift(Rsymb);
rsymb=ifft(ifreq);

rsymb=rsymb((1:(Ns-1))*samples);
% dsymb=(rsymb<=0).';

```

```
% symbols = -2 * dsymb + 1;
symbols = rsymb;

if sp.flags.coding
    switch sp.code.Type
        case 'cc' % Convolutional coding
            symbols = [symbols symbols(end)];
            symbols = symbols.';
        case 'ldpc' % LDPC coding
            symbols = [0 symbols];
            symbols = symbols.';
    end
end

if sp.flags.debug
    scatterplot(symbols)
    disp('Catherine's code...')
    keyboard
end
```