



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

" Μελέτη και παραμετροποίηση του DVB-Πρωτοκόλλου ανάλογα με το περιεχόμενο "

ΠΑΠΑΔΑΚΣΗ ΚΩΝΣΤΑΝΤΙΝΟΣ(Α.Μ. 1071)

ΕΠΙΒΛΕΠΩΝ : Επ. Καθηγητής ΑΝΤΩΝΟΠΟΥΛΟΣ ΧΡΗΣΤΟΣ

ΠΑΤΡΑ, 2020

Contents

Πρόλογος	5
ΚΕΦΑΛΑΙΟ 1: ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΗΛΕΟΡΑΣΗΣ ΑΠΟ ΑΝΑΛΟΓΙΚΗ ΣΕ ΨΗΦΙΑΚΗ	6
1.1 Εισαγωγή στην Τηλεόραση	7
1.2 Η εφεύρεση της Τηλεόρασης	7
1.3 Οι πρώτες μεταδόσεις	10
1.4 Τα είδη της τηλεόρασης	10
1.4.1 Ασπρόμαυρη τηλεόραση	11
1.4.2 Η έγχρωμη τηλεόραση	11
1.4.3 Ψηφιακή τηλεόραση	13
1.4.4 Δορυφορική Τηλεόραση	14
1.4.5 Διαδικτυακή τηλεόραση	15
1.4.6 Αμφίδρομη Τηλεόραση	15
ΚΕΦΑΛΑΙΟ 2: ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ DVB-T ΠΡΩΤΟΚΟΛΛΟΥ	16
2.1 Εισαγωγή στο DVB-T πρωτόκολλο	16
2.2 Τεχνικά χαρακτηριστικά του DVB-T	16
2.2.1 Αλγόριθμος Reed-Solomon	23
2.2.2 Convolutional Interleaver	27
2.2.3 Convolutional Code	29
2.2.4 Viterbi decoder	36
2.2.5 Block Interleaver	37
2.2.6 Constellation Diagram	39
2.2.7 Orthogonal frequency – division multiplexing	43
2.4 Το πρωτόκολλο DVB-T2	44
2.4.1 Διαφορές του DVB-T με το DVB-T2	44
ΚΕΦΑΛΑΙΟ 3: ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΩΤΟΚΟΛΛΟΥ DVB-T ΜΕΣΩ ΤΟΥ MATLAB	45
3.1 Random-integer Generator	46
3.2 Reed-Solomon encoder και decoder	47
3.3 Convolutional interleaver και deinterleaver	49
3.4 Convolutional code	50
3.5 Inner Interleaver	51
3.6 Mapper	53

3.7 OFDM Modulator και Demodulator	55
3.8 AWGN.....	56
3.9 Spectrum Scope και Scatter Plot.....	58
3.10 Demapper	59
3.11 Viterbi Decoder	62
3.12 Error Rate Calculation	63
ΚΕΦΑΛΑΙΟ 4: ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΜΕΣΩ MATLAB.....	65
4.1 Κώδικας του DVB-T Πρωτοκόλλου	65
4.2 Αλλαγές στο κύκλωμα του DVB-T στο Matlab.....	71
4.3 Script για τα delay και για τις μετρήσεις	73
4.4 Μετρήσεις και συμπεράσματα.....	76
ΚΕΦΑΛΑΙΟ 5: ΣΥΝΟΨΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	87
Bibliography	89

Λίστα εικόνων

Figure 1 Nipkow Disk.....	8
Figure 2 Ασπρόμαυρη Τηλεόραση.....	11
Figure 3 Η πρώτη έγχρωμη τηλεόραση	12
Figure 4 Satellite TV	14
Figure 5 Internet TV	15
Figure 6 Σύστημα εκπομπής DVB-T	18
Figure 7 Reed-Solomon Codeword	25
Figure 8 Convolutional Interleaver	28
Figure 9 Deinterleaver	28
Figure 10 Rate 1/3, convolutional encoder with constraint length 3.....	31
Figure 11 Rate 1/2 8-state recursive systematic convolutional encoder.	32
Figure 12 Trellis diagram.....	33
Figure 13 Convolutional Code (7,[171 133])	34
Figure 14 Block Interleaver & Deinterleaver	38
Figure 15 Row & Column major order	39
Figure 16 8 – PSK.....	40
Figure 17 Modulation Schemes	42
Figure 18 DVB-T protocol.....	46
Figure 19 Random-Integer Generator	47
Figure 20 Reed-Solomon Encoder.....	48
Figure 21 Convolutional interleaver	49
Figure 22 Convolutional code	50
Figure 23 Inner Interleaver	51
Figure 24 Block interleaver	52
Figure 25 Symbol interleaver	53
Figure 26 Modulator	54
Figure 27 OFDM modulator	56
Figure 28 AWGN.....	57
Figure 29 Spectrum Scope	58
Figure 30 Scatter Plot.....	59
Figure 31 Demapper	60
Figure 32 Variance Calculator	60
Figure 33 Demodulator	61
Figure 34 Inner Decoder	62
Figure 35 Viterbi Decoder	63
Figure 36 Error Rate Calculation	64
Figure 37 Inner Decoder	72
Figure 38 Scatter Plot & Spectrum Scope for QPSK.....	77
Figure 39 Διάγραμμα BER vs SNR for QPSK	77
Figure 40 Scatter Plot & Spectrum Scope for 16-QAM.....	78

Figure 41 Διάγραμμα BER vs SNR for 16-QAM	78
Figure 42 Scatter Plot & Spectrum Scope for 64-QAM.....	79
Figure 43 Διάγραμμα BER vs SNR for 64-QAM	79
Figure 44 Spectrum Scope 2k mode	81
Figure 45 Διάγραμμα 2k mode	81
Figure 46 Spectrum Scope 8k mode	82
Figure 47 Διάγραμμα 8k mode	82
Figure 48 Code Rate 1/2	83
Figure 49 Code Rate 2/3	84
Figure 50 Code Rate 3/4	84
Figure 51 Code Rate 5/6	85
Figure 52 Code Rate 7/8	85

Πρόλογος

Το θέμα της πτυχιακής μου είναι:

ΜΕΛΕΤΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΟΥ DVB-T ΠΡΩΤΟΚΟΛΛΟΥ ΑΝΑΛΟΓΑ ΜΕ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ

Η επιλογή του συγκεκριμένου θέματος έγινε έχοντας ως στόχο να αποκτήσω περισσότερες γνώσεις πάνω στο θέμα της τηλεόρασης και πιο συγκεκριμένα της ψηφιακής τηλεόρασης η οποία έχει πρωταγωνιστικό ρόλο στις ζώες μας. Πιο συγκεκριμένα θα ασχοληθώ με το πρωτόκολλο DVB-T το οποίο χρησιμοποιείται στους αποκωδικοποιητές μας καθώς και με όλη τη διαδρομή που περνάει το σήμα από το πομπό ως και τον δέκτη.

Από την αρχή της δημιουργίας της η τηλεόραση αποτελεί, αν όχι το βασικότερο, ένα από τα βασικότερα μέσα ενημέρωσης και ψυχαγωγίας, καθώς υπάρχει στα σπίτια όλων και την χειρίζονται όλες οι ηλικίες, από μικρά παιδιά μέχρι και ηλικιωμένους ανθρώπους. Για τον λόγο αυτό λοιπόν αποφάσισα να ασχοληθώ με την τηλεόραση.

Σύμφωνα λοιπόν με όσα ανέφερα παραπάνω στη πτυχιακή μου θα αναφερθώ στην ιστορία της τηλεόρασης και όλες τις τεχνολογίες που εφευρέθηκαν ανά καιρούς όπως για παράδειγμα την έγχρωμη τηλεόραση κτλ. Στη συνέχεια θα ασχοληθώ με το πρωτόκολλο DVB-T και με τους αλγόριθμους που χρησιμοποιεί για την κωδικοποίηση του σήματος. Και τέλος με τη βοήθεια του εργαλείου MATLAB θα τρέξω κάποιες προσομοιώσεις του DVB-T αλλάζοντας τις μεταβλητές του προκειμένου να βγάλω κάποια συμπεράσματα για τη χρήση του σε διάφορες καιρικές συνθήκες.

ΚΕΦΑΛΑΙΟ 1: ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΗΛΕΟΡΑΣΗΣ ΑΠΟ ΑΝΑΛΟΓΙΚΗ ΣΕ ΨΗΦΙΑΚΗ

1.1 Εισαγωγή στην Τηλεόραση

Η τηλεόραση είναι ένα σύστημα τηλεπικοινωνίας το οποίο χρησιμοποιείται για τη μετάδοση και λήψη κινούμενων μονόχρωμων(μαύρο - άσπρο), έγχρωμων και τριών διαστάσεων εικόνων αλλά και ήχου εξ αποστάσεως. Αποτελεί το κυριότερο και δημοφιλέστερο Μέσο Μαζικής Επικοινωνίας και η χρήση της είναι ιδιαίτερα διαδεδομένη σε όλο τον κόσμο.

1.2 Η εφεύρεση της Τηλεόρασης

Η εφεύρεση της τηλεόρασης δεν ήταν αποτέλεσμα δουλειάς ενός αλλά πολλών ατόμων γύρω στα τέλη του 19^{ου} αιώνα. Άνθρωποι και εταιρείες από διάφορα μέρη του κόσμου συμμετείχαν στη δημιουργία μιας νέας συσκευής η οποία θα αντικαθιστούσε τη προηγούμενη τεχνολογία. Υπήρξαν πολλοί οι οποίοι θέλησαν να επωφεληθούν από αυτή την εφεύρεση, ενώ υπήρξαν και άλλοι οι οποίοι ήθελαν να αλλάξουν τον κόσμο μέσα από την τεχνολογία οπτικής επικοινωνίας. Για να φτάσουμε όμως στη τηλεόραση όπως τη ξέρουμε σήμερα χρειάστηκε να περάσουν αρκετά χρόνια. Αφού το πρώτο τηλεοπτικό σύστημα λειτούργησε τη δεκαετία του '40.

Η μηχανή από την οποία προήλθε η βασική ιδέα στην οποία στηρίχτηκε η δημιουργία της τηλεόρασης ήταν το τηλέγραφο του οποίου δημιουργός ήταν ο **Samuel Morse**¹ καθώς και το τηλέφωνο του **Alexander Graham Bell**². Το τηλέγραφο μπορούσε να μεταδώσει συνδυασμούς κωδικοποιημένων λέξεων και γραμμάτων δια μέσου των ηλεκτρονικών παλμών κατά μήκος των καλωδίων. Με την ίδια λογική, για να γίνει μετάδοση εικόνων θα μπορούσε το φως να μετατραπεί σε ηλεκτρικούς παλμούς, κάνοντας έτσι δυνατή τη μεταβίβαση των παλμών αυτών σε απόσταση και την επαναφορά τους σε φως. Για να επιτευχθεί αυτό βοήθησε η ανακάλυψη των φωτοηλεκτρικών ιδιοτήτων του σεληνίου το **1873**. Σε αυτό βοήθησε ο αμερικάνος **George R. Carey**³ ο οποίος πρότεινε την δημιουργία ενός τηλεοπτικού δικτύου που θα μπορούσε να

¹ Ο Samuel Morse (27 Απριλίου 1791 – 2 Απριλίου 1872) ήταν Αμερικάνος ζωγράφος και εφευρέτης ο οποίος εφεύρε το τηλέγραφο και βοήθησε στη ανάπτυξη του κώδικα Morse **Invalid source specified..**

² Ο Alexander Graham Bell (3 Μαρτίου 1847 – 2 Αυγούστου 1922) ήταν διαπρεπής Σκωτσέζος επιστήμονας, εφευρέτης και μηχανικός, ο οποίος θεωρείται ως εφευρέτης του πρώτου πρακτικού τηλεφώνου **Invalid source specified..**

³ Ο George R. Carey ήταν Αμερικάνος εφευρέτης ο οποίος ήταν από τους πρώτους που σχεδίασε το ηλεκτροσκόπιο χρησιμοποιώντας τις φωτοηλεκτρικές ιδιότητες του σεληνίου ως μέσο μετάδοσης εικόνων **Invalid source specified..**

μεταδώσει κινούμενες εικόνες σε απόσταση με τη μετατροπή της φωτεινής ροής που εκπέμπεται από τα διάφορα σημεία μίας εικόνας, σε ηλεκτρομαγνητικά σήματα. Σε αυτό το σύστημα τόσο ο πομπός όσο και ο δέκτης αποτελούνταν από 2500 φωτοηλεκτρικά κύτταρα σεληνίου και άλλες τόσες λυχνίες, ενώ η μηχανή λήψης και οθόνη συνδεόταν με 2500 καλώδια.

Αργότερα, το **1872**, ο Γάλλος **Constantin Senlecq** ήταν αυτός ο οποίος διατύπωσε τη θεμελιώδη αρχή της διαδοχικής μετάδοσης των στοιχείων της εικόνας, μέσα από μια συσκευή που είχε μόνο ένα καλώδιο. Σε αυτή τη συσκευή μεταδίδονταν διαδοχικά όλα τα τμήματα της εικόνας σε 0,1 δευτερόλεπτα κάνοντας τον τηλεθεατή να πιστεύει πως η προβολή της εικόνας ήταν συνεχής.

Η πρώτη όμως συσκευή για την ανάλυση μιας σκηνής ώστε να παραχθούν ηλεκτρικά σήματα κατάλληλα για μετάδοση ήταν ένα σύστημα ανίχνευσης που προτάθηκε και κατασκευάστηκε από τον **Paul Nipkow** το 1884 ο οποίος ήταν ένας 23χρονος Γερμανός φοιτητής. Αυτή η συσκευή ονομάστηκε **Nipkow disk**. Ο δίσκος Nipkow ήταν ένας περιστρεφόμενος δίσκος με διάφορες μικρές τρύπες σε σπειροειδή διάταξη, μπροστά από ένα φωτοηλεκτρικό κύτταρο.

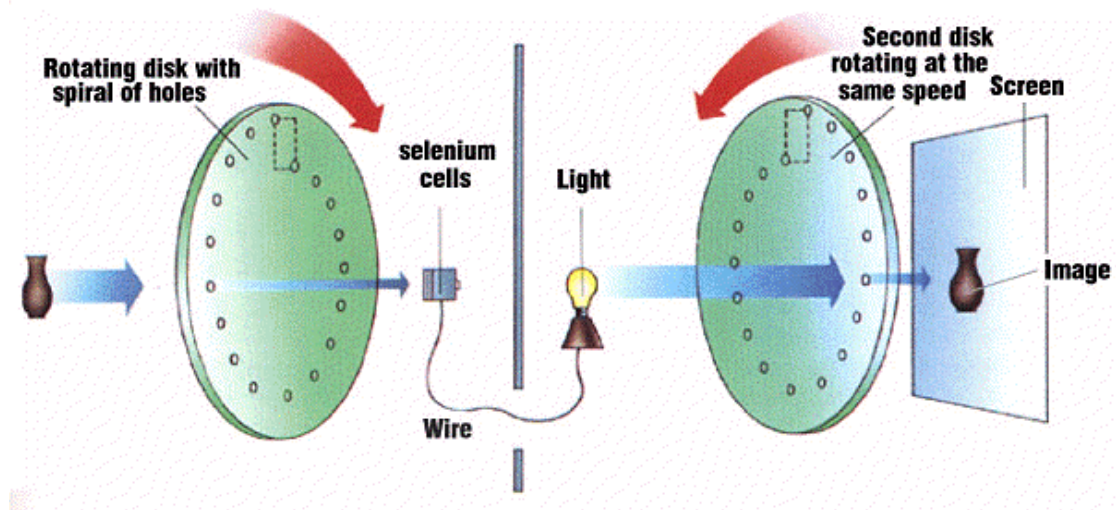


Figure 1 Nipkow Disk

Ο δίσκος Nipkow χρησιμοποιήθηκε από το 1923 έως το 1925 σε πειραματικά συστήματα που αναπτύχθηκαν στις ΗΠΑ από τον **Charles F. Jenkins**⁴ και στην Μεγάλη Βρετανία από τον **John Logie Baird**⁵. Παρά τις βελτιώσεις που κατάφεραν στο δίσκο Nipkow οι παραπάνω, οι σοβαροί περιορισμοί της μηχανικής προσέγγισης, η ανεπάρκεια οπτικού συστήματος και έλλειψη μεγάλης έντασης πηγής φωτός ικανής να διαμορφωθεί από ένα ηλεκτρικό σήμα στις υψηλότερες συχνότητες που απαιτούνται, είχε αποτέλεσμα την διακοπή οποιασδήποτε πρακτικής εφαρμογής του δίσκου Nipkow.

Το 1929 ο Charles Jenkins δημιούργησε ένα τηλεοπτικό σύστημα που μπορούσε να δείξει είδωλα σε μια οθόνη ενός δέκτη. Αυτό το τηλεοπτικό σύστημα χρησιμοποιήθηκε για να εκπέμψει τηλεοπτικές εικόνες από την Ουάσιγκτον στην Φιλαδέλφεια. Στην Αγγλία χρησιμοποιείται το ίδιο σύστημα από τον John Baird ενώ η British Broadcasting Company(**BBC**) μεταδίδει προγράμματα με τη χρήση του συγκεκριμένου συστήματος.

Αργότερα την ίδια χρονιά ο **Phil Farnsworth** ένας Αμερικάνος μηχανικός παρουσίασε μια τηλεοπτική κάμερα που ονόμασε εικονοτόμο (image dissector). Αν και κατασκευάστηκε πριν την εποχή του, καινοτομούσε στο ότι υπήρξε ένα εξ' ολοκλήρου ηλεκτρονικό σύστημα που μετέτρεπε κάθε στοιχείο της εικόνας σε ηλεκτρικό σήμα (Wikipedia, Wikipedia, 2015).

Η πραγματική όμως επανάσταση στην εξέλιξη της τηλεόρασης με το εικονοτόμο ήρθε με τη χρήση καθοδικών ακτινών σε γυάλινες λυχνίες κενού(cathode ray tube ή CRT). Το **1907** και το **1908** ο **Boris Rosing**⁶ και ο **Alan Archibald Campbell-Swinton**⁷ αντίστοιχα είχαν προτείνει, ανεξάρτητα ο ένας από τον άλλον, την χρήση καθοδικού σωλήνα για την αναπαραγωγή της τηλεοπτικής εικόνας. Η ανάπτυξη του CRT για τη χρήση του στη τηλεόραση έγινε από τον **Allen Dumont** τη δεκαετία του '30 στις ΗΠΑ.Ο πρώτος τηλεοπτικός δέκτης κατασκευάστηκε από την General Electric(GE) το 1928. Η πρώτη γενιά τηλεοράσεων δεν ήταν

⁴ Ο Charles Francis Jenkins(22 Αυγούστου 1867-6 Ιουνίου 1934) ήταν Αμερικάνος και ένας από τους εφευρέτες της τηλεόρασης αν και χρησιμοποίησε περισσότερο μηχανικές παρά ηλεκτρονικές τεχνολογίες **Invalid source specified..**

⁵ Ο John Logie Baird (14 Αυγούστου 1888 – 1946) ήταν Σκωτσέζος μηχανικός και ήταν ο πρώτος στην Ιστορία που κατάφερε να αναμεταδώσει εικόνες από ένα σημείο στο άλλο, όπως κάνει σήμερα η τηλεόραση, της οποίας θεωρείται ο εφευρέτης **Invalid source specified..**

⁶ Ο Boris Rosing(23 Απριλίου 1869 – 20 Απριλίου 1933) ήταν Ρώσος επιστήμονας και εφευρέτης πάνω στο πεδίο της τηλεόρασης **Invalid source specified..**

⁷ Ο Alan Archibald Campbell-Swinton(18 Οκτωβρίου 1863 – 19 Φεβρουαρίου 1930) ήταν Σκωτσέζος ηλεκτρολόγος μηχανικός, ο οποίος παρείχε τη θεωρητική βάση της ηλεκτρονικής τηλεόρασης δύο δεκαετίες πριν την ύπαρξη της τεχνολογίας για την εφαρμογή της **Invalid source specified..**

πλήρως ηλεκτρονική αφού στην οθόνη υπήρχε ένα μικρό μοτέρ με ένα περιστρεφόμενο δίσκο και μια λάμπα νέον. Αυτά τα δύο δούλευαν μαζί δίνοντας μια θολή εικόνα.

1.3 Οι πρώτες μεταδόσεις.

Η πρώτη δοκιμαστική μετάδοση πραγματοποιήθηκε στην ευρύτερη περιοχή της Νέας Υόρκης μέσω του σταθμού WGY. Αυτό πραγματοποιήθηκε στις 10 Μαΐου το 1928. Η πρώτη κανονική δημόσια εκπομπή τηλεοπτικού προγράμματος πραγματοποιήθηκε στο Λονδίνο το 1936. Δύο ήταν οι εταιρείες οι οποίες πραγματοποίησαν αυτές τις εκπομπές οι οποίες ήταν η Marconi-EMI και η Baird-Television. Η πρώτη μετάδοση εξωτερικού γεγονότος έγινε στις 20 Απριλίου του 1939 και τότε ήταν που καλύφθηκε η Διεθνής Έκθεση στην Νέα Υόρκη (World Fair).

Τον Ιούλιο του 1941 οι σταθμοί NBC και CBS πήραν από την FCC (Federal Communication Commission) την άδεια ως πρώτοι εμπορικοί σταθμοί στις ΗΠΑ. Το 1945 το FCC κατανέμει το φάσμα των συχνοτήτων σε 13 τηλεοπτικά κανάλια VHF. Από εκεί και πέρα η εξέλιξη ήταν ραγδαία. Το FCC το 1952 επεκτείνει τις τηλεοπτικές εκπομπές με τη κατανομή του FCC φάσματος σε άλλα 70 νέα τηλεοπτικά κανάλια.

1.4 Τα είδη της τηλεόρασης

Το ευρύ κοινό από τα πρώτα χρόνια εμπορευματοποίησης της τηλεόρασης την αποδέχτηκε με μεγάλη ανταπόκριση αφού αποτελούσε πρωταρχική πηγή ψυχαγωγίας και ενημέρωσης. Πολλοί χαρακτηρίζουν την τηλεόραση μέσω καθορισμού της θέσης του καναπέ στα σαλόνια των θεατών. Η εξέλιξη των τεχνολογιών και φυσικά της τηλεόρασης έδωσε τη δυνατότητα στους θεατές να έχουν διάφορες μορφές τηλεόρασης που η κάθε μία προσφέρει τις δικές τις δυνατότητες. Παρακάτω θα τα δούμε τα είδη της τηλεόρασης που έχουν βγει μέχρι και σήμερα.

1.4.1 Ασπρόμαυρη τηλεόραση



Figure 2 Ασπρόμαυρη Τηλεόραση

Όταν πρωτοεμφανίστηκε η ασπρόμαυρη τηλεόραση είχε πολύ περιορισμένες δυνατότητες. Σαν συσκευή ήταν ογκώδης και είχε οκτώ κουμπιά από τα οποία μόνο τα δύο λειτουργούσαν. Ακόμα και ο χρόνος απόκρισης είτε για να κάνεις εναλλαγή σε κάποια κανάλια είτε για να την ανοίξεις ήταν πολύ μεγάλος.

Η ασπρόμαυρη τηλεόραση δείχνει να αποτελείται από ένα μεγάλο αριθμό στίγματα, κουκίδες κανονισμένες σε ένα συνηθισμένο γεωμετρικό υπόδειγμα. Τα στίγματα αυτά είναι πολύ μικρά στις άσπρες περιοχές της εικόνας και πολύ μεγάλα, αφού σχεδόν έρχονται σε επαφή μεταξύ τους, στις πιο σκοτεινές περιοχές. Τα στίγματα με μεσαία μεγέθη δίνουν διάφορες ποικιλίες γκριζου χρώματος, τα οποία βρίσκονται σε ανάμεσα στις φωτεινές και σκοτεινές περιοχές. Αυτή η τεχνική της διαίρεσης της εικόνας σε πολύ μικρά στοιχεία χρησιμοποιείται στη τηλεόραση. Πληροφορίες για το βαθμό του γκριζου δηλαδή για το βαθμό φωτεινότητας της εικόνας εκπέμπονται στο άκρο λήψεως που χρησιμοποιούνται για να ανοικοδομήσουν την εικόνα της αρχικής σκηνής. Αυτές οι σκηνές απεικονίζουν την εικόνα σε 30 γραμμές και αποτελούνται από κινητές εικόνες.

1.4.2 Η έγχρωμη τηλεόραση



Figure 3 Η πρώτη έγχρωμη τηλεόραση

Στα πρώτα χρόνια γινόταν χρήση της ασπρόμαυρης τηλεόρασης και λίγο αργότερα έκανε την εμφάνισή της η έγχρωμη. Η καθυστέρηση στη χρήση της έγχρωμης τηλεόρασης δεν οφειλόταν μόνο στις πιο πολύπλοκες αρχές λειτουργίας της σε σχέση με την ασπρόμαυρη αλλά και στην έλλειψη συμβατότητας με τα συστήματα της ασπρόμαυρης τηλεόρασης.

Το 1904 έγινε αντιληπτό ότι για την παραγωγή της έγχρωμης τηλεόρασης έπρεπε να γίνει χρήση τριών βασικών χρωμάτων του φωτός, του κόκκινου, του μπλε και του πράσινου. Ο Baird το 1928 παρουσίασε την έγχρωμη τηλεόραση κάνοντας χρήση του δίσκου του Nipkow. Το 1940 στη Νέα Υόρκη ο Peter Goldmark παρουσίασε ένα καλύτερης ποιότητας σύστημα έγχρωμης τηλεόρασης. Αυτό το νέο σύστημα χρησιμοποιήθηκε το 1951 για δημόσια εκπομπή, το οποίο όμως εγκαταλείφθηκε λόγω της ασυμβατότητάς του με την ασπρόμαυρη τηλεόραση. Το 1953 ήταν τότε που τελειοποιήθηκε η συμβατή έγχρωμη τηλεόραση και ένα χρόνο μετά εφαρμόστηκε με τη χρήση του συστήματος NTSC, το οποίο χρησιμοποιείται μέχρι και σήμερα στην Αμερική. Το σύστημα αυτό πήρε το όνομα του από τα αρχικά του οργανισμού National Television System Committee. Αυτός ήταν ο οργανισμός ο οποίος ερευνήσε σε έκταση το θέμα και δημιούργησε το σύστημα που έγινε αποδεκτό από την βιομηχανία και το κοινό. Το NTSC έχει καθιερωθεί και χρησιμοποιείται στην Αμερική, στον Καναδά και στην Ιαπωνία. Επίσης αποτελεί την βάση για τα συστήματα που ακολούθησαν.

Στην Ευρώπη χρησιμοποιούνται άλλα συστήματα. Εκεί έχουν καθιερωθεί και έχουν γίνει αποδεκτά τα συστήματα PAL και SECAM. Το PAL (Phase Alteration Line) προτάθηκε στη Γερμανία από τον Dr Bruch και στην ουσία είναι μια παραλλαγή του συστήματος NTSC. Αυτό είναι το σύστημα το οποίο χρησιμοποιείται από το 1967 σε πολλές χώρες της Ευρώπης. Αντίθετα το SECAM προτάθηκε στη Γαλλία το 1958 από τον Henri de France. Τα αρχικά σημαίνουν “Sequentiel Couleur a Memoire”. Αυτό το σύστημα χρησιμοποιείται στη Γαλλία, στη Σοβιετική Ένωση και σε άλλες ανατολικές χώρες της Ευρώπης από το 1967. Στην Ελλάδα ήταν το 1980 τότε που καθιερώθηκε το σύστημα SECAM.

Έγινε προσπάθεια για καθιέρωση ενός κοινού συστήματος που θα ήταν αποδεκτό σε όλο τον κόσμο. Έτσι το 1950 στη διεθνή σύσκεψη του IRCC που έγινε στο Λονδίνο διαμορφώθηκαν 4 βασικά συστήματα. Αυτά τα συστήματα ήταν το Ευρωπαϊκό με 625 οριζόντιες γραμμές ανά εικόνα, το Αγγλικό με 405, το Αμερικάνικο με 525 και το Γαλλικό με 819. Από αυτά τα συστήματα, το Γαλλικό και το Αγγλικό καταργήθηκαν το 1984, με αποτέλεσμα να έχουμε σήμερα σε όλο τον κόσμο δύο βασικά συστήματα, το Αμερικάνικο των 525 γραμμών και 60Hz και το Ευρωπαϊκό των 625 γραμμών και 50Hz, τα οποία έχουν μεταξύ τους κάποια κοινά χαρακτηριστικά.

Στη συνέχεια η τηλεόραση αναπτύχθηκε με μεγάλους ρυθμούς και συνεχίζει να αναπτύσσεται μέχρι τις μέρες μας. Νέες τεχνολογίες εφαρμόζονται συνεχώς με αποκορύφωμα τη ψηφιακή τηλεόραση η οποία εισβάλλει στη ζωή μας με γρήγορους ρυθμούς (Σαραντής, 2014).

1.4.3 Ψηφιακή τηλεόραση

Η επίγεια ψηφιακή τηλεόραση πρωτοεμφανίστηκε το 1972, ενώ ξεκίνησε δεκαετίες πριν για επαγγελματικούς και στρατιωτικούς σκοπούς. Η ψηφιοποίηση του τηλεοπτικού σήματος γίνεται μέσα από τη μετατροπή της εικόνας και του ήχου σε δυαδικούς αριθμούς 0 και 1. Αυτό έχει ως αποτέλεσμα την τεχνολογική πρόοδο και την εμφάνιση νέων δυνατοτήτων και υπηρεσιών προσφέροντας πλεονεκτήματα τόσο στη πλευρά του τηλεθεατή όσο και στον παρόχου. Ο όρος επίγεια ψηφιακή τηλεόραση βγαίνει από την μετάδοση του ψηφιακού σήματος μέσω ασυρμάτου επίγειου δικτύου ευρυεκπομπής.

Αυτή η τεχνολογία αναπτύσσεται και εξελίσσεται με γρήγορους ρυθμούς και κυριαρχεί σε όλους τους τομείς της σημερινής τεχνολογίας συμβάλλοντας στη δημιουργία ενός ψηφιακού κόσμου στον οποίο οι υπηρεσίες διαφορετικών τεχνολογιών θα συγκλίνουν. Η καθυστέρηση στην ψηφιακή ευρυεκπομπή των τηλεοπτικών προγραμμάτων οφείλεται στην δυσκολία ψηφιοποίησης των τηλεοπτικών σημάτων και στις δυσκολίες που προκύπτουν από την συμβατότητα της υπάρχουσας τεχνολογίας.

Επίσης η ανάπτυξη προτύπων για τη μετάδοση ψηφιακών τηλεοπτικών σημάτων σε παγκόσμιο επίπεδο αποτελούσε δυσκολία και ανασταλτικό παράγοντα. Γι αυτό το λόγο η εμπορευματοποίηση της ψηφιακής τηλεόρασης βρίσκεται σε πρώιμο στάδιο. Αλλά η ψηφιακή τηλεόραση έχει αρχίσει να αναπτύσσεται με γρήγορους ρυθμούς (Wikipedia, Wikipedia, 2015).

1.4.4 Δορυφορική Τηλεόραση



Figure 4 Satellite TV

Με τον όρο δορυφορική τηλεόραση εννοούμε την εκπομπή και τη λήψη τηλεοπτικού σήματος από επίγειους σταθμούς μέσω δορυφόρου. Η λειτουργία του δορυφόρου είναι να λαμβάνει το τηλεοπτικό σήμα από τη γη και να αναμεταδίδει πίσω. Αρχικά η δορυφορική

τηλεόραση ήταν αναλογική αλλά πλέον είναι ψηφιακή. Οι δορυφόροι που χρησιμοποιούνται στη ψηφιακή τηλεόραση υπάγονται στην κατηγορία των γεωστατικών δορυφόρων.

1.4.5 Διαδικτυακή τηλεόραση



Figure 5 Internet TV

Η διαδικτυακή τηλεόραση μπορεί να περιγραφεί ως ένα σύστημα κατά το οποίο ένα ψηφιακό τηλεοπτικό σήμα εκπέμπεται σε συνδρομητές του internet με τη βοήθεια του IP (**Internet Protocol**) και μιας ευρυζωνικής (**broadband**) σύνδεσης. Η υπηρεσία αυτή τις περισσότερες φορές παρέχεται με μία άλλη υπηρεσία που ονομάζεται ‘Βίντεο κατ’ απαίτηση’ (**Video On Demand**) και ταυτόχρονα μπορεί να περιλαμβάνει και άλλες υπηρεσίες. Το τηλεοπτικό σήμα είναι συνήθως κωδικοποιημένο σε μορφή αναγνωρίσιμη από τον υπολογιστή ή άλλα ψηφιακά μέσα και διανέμεται μέσω της **IP Multicast**. Αυτή η μέθοδος δίνει την δυνατότητα να αποστείλεις πληροφορία ταυτόχρονα σε πολλούς αποδέκτες – υπολογιστές.

1.4.6 Αμφίδρομη Τηλεόραση

Η αμφίδρομη τηλεόραση με σταθερά βήματα αναπτύσσεται. Με την αμφίδρομη τηλεόραση μας δίνεται η δυνατότητα να επιλέγουμε προγράμματα της αρεσκείας μας όποτε επιθυμούμε, να πλοηγούμαστε στο διαδίκτυο και να απολαμβάνουμε μια σειρά από υπηρεσίες.

Με αυτό τον τρόπο ο τηλεθεατής μπορεί να παρακολουθήσει ότι θέλει και όποτε το θέλει χωρίς να εξαρτάται από το πρόγραμμα του κάθε καναλιού.

ΚΕΦΑΛΑΙΟ 2: ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ DVB-T ΠΡΩΤΟΚΟΛΛΟΥ

2.1 Εισαγωγή στο DVB-T πρωτόκολλο

Το DVB-T ή αλλιώς Digital Video Broadcast – Terrestrial είναι το πιο διαδεδομένο πρότυπο ψηφιακής τηλεόρασης που χρησιμοποιείται στον κόσμο για τις επίγειες τηλεοπτικές εκπομπές. Παρέχει κάποιες ανέσεις και επιτρέπει μία πολύ πιο αποτελεσματική χρήση του διαθέσιμου φάσματος ραδιοσυχνοτήτων από τις προηγούμενες αναλογικές μεταδόσεις. Το πρότυπο αυτό συμφωνήθηκε το 1997 αργώντας χρονικά σε σχέση με το DVB-S. Αυτό συνέβη λόγω της πολυπλοκότητας του προτύπου. Η πολυπλοκότητα αυτή οφείλεται στο ότι πρέπει να ανταποκρίνεται σε διαφορετικά περιβάλλοντα θορύβου, εύρους ζώνης και πολυδιαδρομικής διάδοσης, κάτι που αυξάνει τη δυσκολία σχεδιάσεως στο επίγειο περιβάλλον. Επίσης υπάρχει έλλειψη ελεύθερου διαθέσιμου φάσματος.

Οι πρώτες μεταδόσεις του DVB-T ξεκίνησαν στη Σουηδία και την Αγγλία το 1998. Αργότερα το 2002 ξεκίνησαν και στην Γερμανία ενώ το 2003 έκλεισε ο πρώτος αναλογικός τηλεοπτικός σταθμός στο Βερολίνο. Το 2008, το DVB-T ήταν το πρότυπο που εγκρίθηκε σε περισσότερες από 35 χώρες και πάνω από 60 εκατομμύρια δέκτες άρχισαν να το χρησιμοποιούν (radio-electronics).

2.2 Τεχνικά χαρακτηριστικά του DVB-T

Το DVB-T πρόκειται για ένα σύστημα το οποίο θεωρείται ευέλικτο αφού έχουμε την δυνατότητα επιλογής των παραμέτρων του ανάλογα με την περίπτωση. Διαθέτει τρεις διαφορετικές διαμορφώσεις την QPSK, την 16 QAM και την 64 QAM. Η επιλογή της καθεμίας και ανάλογα το περιβάλλον στο οποίο τη χρησιμοποιούμε, παρουσιάζουν κάποια πλεονεκτήματα και μειονεκτήματα.

Οπότε έχουμε την QPSK η οποία παρουσιάζει μεγάλη ανθεκτικότητα στον θόρυβο του καναλιού και στις επικείμενες παρεμβολές αλλά έχει ως αντίκτυπο τις μικρές τιμές ρυθμών μετάδοσης που κυμαίνονται από 5Mbps – 10Mbps.

Η 16 QAM προσφέρει μία μέση λύση, εξισορροπώντας το επίπεδο ανεκτικότητας στον θόρυβο και απόδοσης εύρους ζώνης. Δηλαδή παρουσιάζει ανεκτικότητα στις παρεμβολές με ικανοποιητικές τιμές απόδοσης εύρους ζώνης. Οι ρυθμοί μετάδοσης που επιτυγχάνονται κυμαίνονται από 10Mbps – 21Mbps.

Η 64 QAM προσφέρει τη μεγαλύτερη τιμή απόδοσης εύρους ζώνης με τιμές που κυμαίνονται από 14Mbps – 31Mbps. Όμως είναι πολύ ευάλωτη στο θόρυβο και στις παρεμβολές. Αυτό έχει ως αποτέλεσμα την απαίτηση μεγαλύτερης εκπεμπόμενης ισχύος έτσι ώστε ο λόγος επιθυμητού σήματος προς παρεμβολή να αυξηθεί για το δεδομένο εύρος ζώνης (C/I – σηματοθορυβικός λόγος).

Το σύστημα DVB-T αντιμετωπίζει την ανοχή του στο θόρυβο με τη χρήση της FEC (Forward Error Correction) κωδικοποίησης η οποία έπεται της κωδικοποίησης Reed Solomon και ενός εξωτερικού interleaver. Στο σήμα πληροφορίας προστίθενται κατά την μετάδοση παραπάνω δυαδικά ψηφία bit. Με την παρατήρηση της αλλοίωσης αυτών των bit στον δέκτη μπορεί αρχικά να διαπιστωθεί αν υπέστη το σήμα αλλοίωση λόγω θορύβου. Στη συνέχεια μπορεί να γίνει μια είδους πρόβλεψη διόρθωσης σφάλματος για το σήμα πληροφορίας μέσω της σύγκρισης των πλεοναζόντων bit πληροφορίας στον δέκτη και στον πομπό. Ο λόγος του αριθμού των bit του αρχικού σήματος προς το ολικό(αρχικό και πλεονάζων) ορίζει τον ρυθμό κωδικοποίησης, που για την περίπτωση του DVB-T υποστηρίζονται πέντε ρυθμοί εσωτερικής κωδικοποίησης οι οποίοι είναι: 1/2, 2/3, 3/4, 5/6 και 7/8. Η κωδικοποίηση του DVB-T γίνεται με τεχνική OFDM και επειδή διαθέτει τεχνικές διόρθωσης σφαλμάτων (FEC), είναι κατά βάση COFDM (coded).

Τα τμήματα από τα οποία αποτελείται ένα σύστημα εκπομπής DVB-T μπορούν να φανούν στο παρακάτω σχήμα.

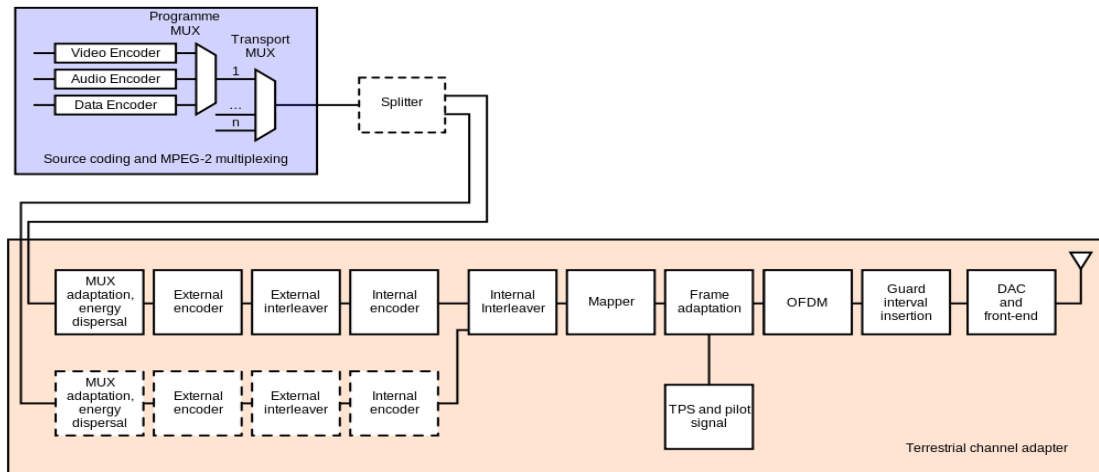


Figure 6 Σύστημα εκπομπής DVB-T

Αναλυτικά τα τμήματα αυτά είναι:

- Source coding and MPEG-2 multiplexing:** Τα συμπιεσμένα βίντεο, ο συμπιεσμένος ήχος και οι ροές δεδομένων πολυπλέκονται σε ροές προγραμμάτων MPEG (MPEG – program streams). Το MPEG – program stream είναι μια μορφή κοντέινερ για πολυπλεξία ψηφιακού ήχου, βίντεο και άλλων. Οι ροές δεδομένων (Program Streams) χρησιμοποιούνται σε δίσκους DVD – Video και σε δίσκους HD DVD Video, αλλά με ορισμένους περιορισμούς και επεκτάσεις. Ένα ή περισσότερα MPEG – PS ενώνονται σε MPEG transport stream. Τα MPEG – Transport Streams ή απλά τα transport streams είναι μια μορφή ψηφιακού κοντέινερ για τη μετάδοση και την αποθήκευση δεδομένων ήχου, βίντεο και δεδομένων Program and System Information Protocol (PSIP). Χρησιμοποιείται σε συστήματα μετάδοσης όπως τα DVB, ATSC και IPTV. Αυτή είναι η βασική ροή η οποία μεταδίδεται και λαμβάνεται από τα Set Top Boxes (STB). Τα επιτρεπτά bitrates για τα μεταφερόμενα δεδομένα εξαρτώνται από έναν αριθμό παραμέτρων κωδικοποίησης και διαμόρφωσης. Αυτά κυμαίνονται από περίπου 5 έως 32 Mbits/s.
- Splitter:** Δύο διαφορετικά MPEG – TS μπορούν να μεταδοθούν την ίδια στιγμή, χρησιμοποιώντας μία τεχνική που ονομάζεται ιεραρχική (Hierarchical). Μπορεί να χρησιμοποιηθεί για τη μετάδοση, για παράδειγμα ενός σήματος SDTV και ενός άλλου

HDTV υψηλής ευκρίνειας, στον ίδιο φορέα. Γενικά το σήμα SDTV είναι πιο στιβαρό από το HDTV. Στον δέκτη, ανάλογα της ποιότητας του λαμβανόμενου σήματος, το STB μπορεί να είναι σε θέση να αποκωδικοποιήσει το σήμα HDTV ή, αν η ισχύς του σήματος είναι χαμηλή, μπορεί να αλλάξει σε SDTV. Με αυτό τον τρόπο, όλοι οι δέκτες που βρίσκονται κοντά στη τοποθεσία του πομπού μπορούν να κλειδώσουν σε HDTV σήμα, ενώ όλοι οι άλλοι που βρίσκονται πιο μακριά μπορεί να είναι ακόμα σε θέση να λαμβάνουν και να αποκωδικοποιούν ένα SDTV σήμα.

- **MUX adaptation and energy dispersal:** Το MPEG – Transport Stream αναγνωρίζεται ως ακολουθία πακέτων δεδομένων, ενός σταθερού μήκους (πχ 188 bytes). Με μια τεχνική που ονομάζεται ενεργειακή διασπορά ή energy dispersal (scrambler), η ακολουθία byte αποχρωματίζεται. Ένας scrambler στις τηλεπικοινωνίες είναι μια συσκευή που μεταφέρει ή αναστρέφει σήματα ή αλλιώς κωδικοποιεί ένα μήνυμα στη μεριά του πομπού έτσι ώστε να το κάνει ακατανόητο στον δέκτη αν αυτός δεν διαθέτει μια κατάλληλα ρυθμισμένη descrambling συσκευή.
- **External Encoder** (Εξωτερικός κωδικοποιητής): Σαν πρώτο επίπεδο διόρθωσης λαθών (error correction) στα μεταδιδόμενα δεδομένα, χρησιμοποιείται ο κωδικοποιητής Reed-Solomon (204, 188). Σύμφωνα με αυτόν τον κωδικοποιητή στα 188 bytes μεταδιδόμενης πληροφορίας προστίθενται άλλα 16 bytes διόρθωσης με αποτέλεσμα τα bytes της μεταδιδόμενης πληροφορίας να είναι 204. Μπορούν να διορθωθούν έως και 8 bytes ανά πακέτο δεδομένων.
- **External Interleaver** (Εξωτερικός διεμπλοκέας): Ο εξωτερικός διεμπλοκέας αναδιατάσσει τα bytes του μεταδιδόμενου σήματος έτσι ώστε να διευκολύνει τη διόρθωση των σφαλμάτων και να αποτρέψει μεγάλες ακολουθίες λαθών. Αν έχουμε λανθασμένα bytes τότε υπάρχει η πιθανότητα τα bytes αυτά να είναι περισσότερα από τα bytes που θα μπορεί να διορθώσει ο κώδικας διόρθωσης λαθών Reed-Solomon(204,188). Αν όμως μπορέσουμε και αναδιατάξουμε τα bytes έτσι ώστε τα γειτονικά bytes να ανήκουν, κατά τη μετάδοση, σε διαφορετικά πακέτα, τότε να λανθασμένα bytes θα

κατανεμηθούν και θα είναι πιο εύκολη η διόρθωση τους. Αυτή η διαδικασία γίνεται με τον εξής τρόπο. Στο ρεύμα εισόδου συνδέονται κυκλικά 12 κλάδοι με ένα μεταγωγέα. Κάθε ένας από αυτούς τους κλάδους είναι και ένας καταχωρητής ολίσθησης FIFO (First In, First Out) με βάθος $17*j$ κύτταρα όπου j ο αριθμός του κλάδου. Κάθε κύτταρο περιέχει ένα byte και οι μεταγωγείς εισόδου και εξόδου είναι συγχρονισμένοι. Για λόγους συγχρονισμού πάντα τα bytes συγχρονισμού οδηγούνται στον κλάδο 0. Αυτή η διαδικασία δεν αλλάζει το ρυθμό μετάδοσης των δεδομένων απλά βοηθάει τον χρήστη να κατανείμει ομοιόμορφα τα λάθη που προκύπτουν σε μία μετάδοση.

- **Internal Encoder** (Εσωτερικός κωδικοποιητής): Μετά από την εξωτερική κωδικοποίηση ακολουθεί η συνελικτική κωδικοποίηση, που στόχος της είναι η αύξηση της απόδοσης της κωδικοποίησης Reed-Solomon. Η αιτία που απαιτείται αποδοτικότερη προστασία για τα σφάλματα είναι ότι στα κανάλια μετάδοσης, η ποιότητα των σημάτων μεταβάλλεται και υπάρχει περίπτωση ένας μεγαλύτερος αριθμός δυαδικών ψηφίων, από εκείνον που μπορεί να διορθώσει το Reed-Solomon, να υποστεί αλλοίωση. Ο λόγος της αλλοίωσης μπορεί να προέρχεται από ένα πιθανό χτύπημα κεραυνού ή από παρεμβολές ηλεκτρικών συσκευών. Όμως η προστασία κάθε πακέτου μετάδοσης από τέτοιου είδους αλλοιώσεις δεν είναι πολύ οικονομική και αυτό γιατί δεν συμβαίνουν αρκετά συχνά. Σε τέτοιες περιπτώσεις εφαρμόζεται η συνελικτική κωδικοποίηση. Στη συνελικτική κωδικοποίηση τα δεδομένα αφού κωδικοποιηθούν με τη μέθοδο FEC (Forward Error Correction), τροφοδοτούνται σε μία RAM και μεταδίδονται αναδιαταγμένα. Με τη χρήση μιας ακόμα RAM κατά τη λήψη, τα δεδομένα επιστρέφουν στην αρχική τους δομή. Το αποτέλεσμα αυτής της κωδικοποίησης είναι το σύνολο των δυαδικών ψηφίων που έχουν υποστεί σφάλμα να μετατρέπεται σε έναν ενιαίο αριθμό εσφαλμένων συμβόλων, τα οποία διορθώνονται εύκολα. Στο συρμό δεδομένων εφαρμόζεται ο συνελικτικός κώδικας με ρυθμό $1/2$ και μήκος 7. Ο συνελικτικός κώδικας μπορεί να έχει τους ρυθμούς $1/2$, $2/3$, $3/4$, $5/6$ και $7/8$.
- **Internal Interleaver** (Εσωτερικός διεμπλοκέας): Σε αυτό στο κομμάτι ο εσωτερικός διεμπλοκέας αναδιατάσσει τις ακολουθίες δεδομένων με σκοπό την μείωση της επιρροής των λαθών. Ο ισχυρός πλεονασμός που εισάγεται επιτρέπει μία πολύ ισχυρή διόρθωση

λαθών. Για κάποια σήματα με πολύ χαμηλό λόγο σήματος προς θόρυβο (SNR, signal Signal to Noise Ratio) στην είσοδο του δέκτη, μπορεί να είναι αναγκαίο. Ωστόσο αυτό έχει ως αποτέλεσμα τον υποδιπλασιασμό της φασματικής απόδοσης του καναλιού. Όμως αυτού του τύπου η συνελικτική κωδικοποίηση επιτρέπει τη μείωση του πλεονασμού που εισάγεται μέσω της διάτρησης της εξόδου του συνελικτικού κωδικοποιητή. Αυτός μας επιτρέπει την μη λήψη όλων των διαδοχικών bits των ακολουθιών εξόδου, αλλά μόνο ένα από τα δύο ταυτόχρονα bits με ένα συγκεκριμένο λόγο διάτρησης. Η εσωτερική κωδικοποίηση των bytes χωρίζεται σε δύο μέρη. Αρχικά έχουμε την αναδιάταξη bit και στη συνέχεια την αναδιάταξη του συμβόλου.

- **Mapper** (Χαρτογραφητής): Σε αυτό το στάδιο σε πλαίσιο OFDM γίνεται η διαμόρφωση όλων των φερόντων. Η διαμόρφωση αυτή γίνεται είτε με το QPSK (Quadrature Phase Shift Keying) είτε με το QAM (Quadrature Amplitude Modulation). Τέλος γίνεται ο συμβολισμός των φερόντων αυτών σε αστερισμούς πολικών διαγραμμάτων.

- **Frame Adaptation** (Προσαρμογή σε πλαίσια): Από τους παραπάνω αστερισμούς μιγαδικών συμβόλων σε πολικό διάγραμμα δημιουργούνται μπλοκ σταθερού μήκους (1512, 3024 ή 6048 σύμβολα ανά μπλοκ). 68 μπλοκ οργανώνονται και αποτελούν ένα πλαίσιο(frame) και ένα υπερπλαίσιο αποτελείται από 4 πλαίσια.
- **TPS and Pilot Signal** (Πιλοτικά και TPS σήματα): Στα προηγούμενα μπλοκ εισάγονται κάποια επιπλέον σήματα πληροφορίας για τον δέκτη για κάποιες παραμέτρους της εκπομπής οι οποίες είναι παραμετροποιήσιμες και μεταβλητές. Αυτές οι παράμετροι είναι το είδος της διαμόρφωσης, η επιλογή ιεραρχικού ή μη τρόπου μετάδοσης, ο ρυθμός διαστήματος διαφύλαξης(guard interval), οι εσωτερικοί ρυθμοί μετάδοσης, ο τρόπος μετάδοσης και ο αριθμός του πλαισίου στο υπερπλαίσιο.
- **OFDM Modulation** (Διαμορφωτής OFDM): Σε αυτό το στάδιο η ακολουθία των blocks διαμορφώνονται σύμφωνα με την OFDM τεχνική, χρησιμοποιώντας 2048, 4046 ή 8192 φέροντα (2k, 4k and 8k).
- **Guard Interval Insertion** (Εισαγωγή Guard Interval): Σε αυτό το βήμα για να μειωθεί η πολυπλοκότητα του δέκτη όσον αφορά τον συγχρονισμό, κάθε OFDM μπλοκ επεκτείνεται αντιγράφοντας στην αρχή του το τέλος του. Το μήκος αυτών των Guard Interval μπορεί να είναι 1/4, 1/8, 1/16 ή 1/32 του αρχικού μήκους του μπλοκ.
- **DAC and Front End** (Μετατροπέας ψηφιακού σε αναλογικού): Στο τελευταίο αυτό στάδιο γίνεται η μετατροπή του ψηφιακού σήματος σε αναλογικό και στη συνέχεια γίνεται η μεταβίβαση σε υψηλότερη συχνότητα ή οποία εξαρτάται από την κεραία εκπομπής.

Στον δέκτη για την αποκωδικοποίηση του λαμβανόμενου σήματος πραγματοποιείται ακριβώς ίδια η παραπάνω διαδικασία αλλά με αντίστροφη σειρά (Wikipedia, Wikipedia, 2015)

Στη συνέχεια θα δούμε πιο αναλυτικά τα δομικά στοιχεία του DVB και θα δούμε και τους αλγόριθμους που χρησιμοποιούνται σε κάθε στοιχείο ξεχωριστά.

2.2.1 Αλγόριθμος Reed-Solomon

Στην αρχή του πρωτοκόλλου DVB-T, στη πλευρά του πομπού, γίνεται η κωδικοποίηση του σήματος. Αυτό το πετυχαίνουμε με τη χρήση του κώδικα Reed – Solomon, ο οποίος είναι ένας Forward Error Correction (FEC) κώδικας. Αντίστοιχα υπάρχει ακόμα ένας στη πλευρά του δέκτη ο οποίος κάνει την αποκωδικοποίηση του σήματος.

Οι αλγόριθμοι Reed-Solomon είναι μια ομάδα αλγορίθμων διόρθωσης σφαλμάτων. Οι αλγόριθμοι αυτοί επινοήθηκαν από τους **Irving S. Reed** και **Gustave Solomon** το 1960. Έχουν πολλές εφαρμογές, οι σημαντικότερες από τις οποίες περιλαμβάνουν καταναλωτικές τεχνολογίες όπως CD, DVD, δίσκους Blu-ray, κωδικούς QR, τεχνολογίες μετάδοσης δεδομένων όπως DSL και WiMAX, συστήματα εκπομπής όπως δορυφορικές επικοινωνίες, DVB και ATSC και συστήματα αποθήκευσης όπως το RAID 6.

Οι αλγόριθμοι Reed-Solomon λειτουργούν σε ένα μπλοκ δεδομένων το οποίο αντιμετωπίζεται ως ένα σύνολο πεπερασμένων στοιχείων πεδίου που ονομάζονται σύμβολα. Οι αλγόριθμοι Reed-Solomon είναι σε θέση να ανιχνεύσουν και να διορθώσουν πολλά σφάλματα συμβόλων. Με τη προσθήκη των συμβόλων ελέγχου t στα δεδομένα, ένας κώδικας μπορεί να ανιχνεύσει (αλλά δεν μπορεί να διορθώσει) οποιονδήποτε συνδυασμό **έως και t** εσφαλμένων συμβόλων και να διορθώσει $t/2$ εσφαλμένα σύμβολα. Ως αλγόριθμος διαγραφής μπορεί να διορθώσει μέχρι και t διαγραφές σε θέσεις που είναι γνωστές και παρέχονται στον αλγόριθμο, ή μπορεί να ανιχνεύσει και να διορθώσει συνδυασμούς σφαλμάτων και διαγραφών. Οι αλγόριθμοι Reed-Solomon είναι επίσης κατάλληλοι ως κωδικοί διόρθωσης πολλαπλών σφαλμάτων, καθώς μια ακολουθία $b+1$ διαδοχικών bit σφαλμάτων μπορεί να επηρεάσει το πολύ δύο σύμβολα μεγέθους b . Η επιλογή του t εξαρτάται από τον σχεδιαστή του κώδικα και μπορεί να επιλέγει εντός ευρέων ορίων.

2.2.1.1 Η δομή του Reed-Solomon

Όπως είπαμε και προηγουμένως ο αλγόριθμος Reed-Solomon στη πραγματικότητα πρόκειται για μια οικογένεια από κώδικες, όπου κάθε κώδικας χαρακτηρίζεται από τρεις παραμέτρους:

- **q** που είναι το μέγεθος αλφαβήτου
- **n** που είναι το μήκος του block
- **k** που είναι το μήκος του μηνύματος

Στις παραπάνω παραμέτρους ισχύει το $k < n < q$. Το σύνολο των **συμβόλων αλφαβήτου** ερμηνεύεται ως πεπερασμένο πεδίο τάξης q , και έτσι το q πρέπει να είναι πρωταρχική δύναμη. Στις πιο χρήσιμες παραμετροποιήσεις του αλγόριθμου Reed-Solomon, το **μήκος του μπλοκ** είναι συνήθως ένα σταθερό πολλαπλάσιο του **μήκους του μηνύματος**, δηλαδή ο **ρυθμός $R=k/n$** είναι μερικός σταθερός. Επιπλέον το μήκος του μπλοκ είναι ίσο ή λίγο μικρότερο(μείον 1) του μεγέθους του αλφαβήτου δηλαδή $n=q$ ή $n=q-1$.

2.2.1.2 Οι ιδιότητες του Reed-Solomon

Οι κώδικες Reed-Solomon είναι ένα υποσύνολο των κωδικών BCH και είναι γραμμικοί. Ένας κώδικας Reed-Solomon καθορίζεται ως **RS (n,k)** με **s-bit** σύμβολα. Αυτό σημαίνει ότι ο κωδικοποιητής παίρνει **k** σύμβολα δεδομένων των **s bits** το καθένα και προσθέτει κάποια σύμβολα ισοτιμίας για να δημιουργήσει ένα codeword **n** συμβόλων. Τα σύμβολα ισοτιμίας είναι **n-k** και κάθε σύμβολο είναι των **s bits**. Ένας αποκωδικοποιητής Reed-Solomon μπορεί να διορθώσει μέχρι και **t** σύμβολα ανά codeword, τα οποία περιέχουν σφάλματα, όπου $2t = n-k$.

Στη παρακάτω φωτογραφία απεικονίζεται ένα τυπικό Reed-Solomon codeword. Αυτός είναι γνωστός και ως συστηματικός κώδικας μιας και τα δεδομένα παραμένουν αμετάβλητα και τα σύμβολα ισοτιμίας προσαρτώνται.

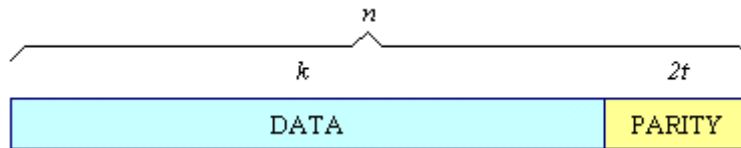


Figure 7 Reed-Solomon Codeword

Ένας δημοφιλής κώδικας Reed-Solomon είναι οι **RS(255,223)** με **8-bit** σύμβολα. Κάθε codeword περιέχει **255** codeword bytes, από τα οποία τα **223** bytes είναι δεδομένα και **32** (255-223) bytes είναι τα σύμβολα ισοτιμίας. Οπότε για το συγκεκριμένο παράδειγμα έχουμε:

$$n = 255, k = 223, s = 8$$

Τα σφάλματα που μπορεί να διορθώσει ο συγκεκριμένος Reed-Solomon είναι:

$$2t = 32, t = 16$$

Επομένως ο συγκεκριμένος αποκωδικοποιητής μπορεί να διορθώσει 16 σφάλματα στο κάθε codeword. Δηλαδή μπορούν να διορθωθούν έως και 16 bytes σε κάθε codeword.

Δεδομένου του μεγέθους συμβόλου s (8-bit), το μέγιστο μήκος του codeword (n) για τον κώδικα Reed-Solomon είναι $n = 2^s - 1$.

2.2.1.3 Οι τεχνικές shortening και puncturing

Οι σχεδιαστές του Reed-Solomon δεν είναι απαραίτητο να χρησιμοποιούν τα “φυσικά” μεγέθη των μπλοκ του αλγόριθμου Reed-Solomon. Μια τεχνική γνωστή και ως **shortening** μπορεί να παράγει έναν μικρότερο κώδικα οποιουδήποτε επιθυμητού μεγέθους από έναν μεγαλύτερο κώδικα. Παράλληλα με το **shortening** υπάρχει άλλη μία τεχνική που ονομάζεται **puncturing** η οποία επιτρέπει να παραλειφθούν ορισμένα από τα κωδικοποιημένα σύμβολα ισοτιμίας.

Έτσι οι κώδικες Reed-Solomon μπορούν περάσουν μια διαδικασία που ονομάζεται **shortening**, όπως είπαμε και παραπάνω. Σε αυτή τη διαδικασία ο κώδικας κάνει έναν αριθμό

συμβόλων δεδομένων μηδέν στον κωδικοποιητή, χωρίς να κάνει μετάδοση των συγκεκριμένων, και στη συνέχεια τα ξανά τοποθετεί στον αποκωδικοποιητή.

Ένα παράδειγμα αποτελεί ο RS(255,223) ο οποίος μπορεί να γίνει shortened σε RS(200,168). Ο κωδικοποιητής παίρνει ένα μπλοκ των 168 bytes δεδομένων, προσθέτει 55 μηδενικά bytes, δημιουργεί ένα codeword (255,223) και μεταδίδει μόνο τα 168 bytes δεδομένων και τα 32 bytes ισοτιμίας. Το ποσό επεξεργαστικής ‘‘ισχύος’’ που απαιτείται για την κωδικοποίηση και την αποκωδικοποίηση των κωδικών Reed-Solomon σχετίζεται με τον αριθμό συμβόλων ισοτιμίας του κάθε codeword. Μια μεγάλη τιμή του t σημαίνει πως ένας μεγάλος αριθμός σφαλμάτων μπορεί να διορθωθεί αλλά αυτό απαιτεί μεγαλύτερη υπολογιστική ισχύ σε σχέση με εκείνη που απαιτείται σε κάποια χαμηλότερη τιμή του t .

2.2.1.4 Σφάλματα συμβόλου

Ένα σφάλμα συμβόλου εμφανίζεται όταν το 1 bit σε ένα σύμβολο είναι λάθος ή όταν όλα τα bit σε ένα σύμβολο είναι λάθος. Για παράδειγμα ο Reed-Solomon (255,223) μπορεί να διορθώσει 16 σφάλματα συμβόλων. Στη χειρότερη περίπτωση, μπορεί να προκύψουν 16 εσφαλμένα bit, το καθένα σε διαφορετικό σύμβολο (byte), έτσι ώστε ο αποκωδικοποιητής να διορθώνει 16 εσφαλμένα bit. Στη καλύτερη περίπτωση, προκύπτουν 16 εσφαλμένα πλήρη byte, έτσι ώστε ο αποκωδικοποιητής να μπορεί να διορθώσει $16 * 8$ εσφαλμένα bit.

Οι κώδικες Reed-Solomon είναι ιδιαίτερα κατάλληλοι για τη διόρθωση ριπής σφαλμάτων (όταν δηλαδή μια σειρά από bits σε ένα codeword λαμβάνονται κατά λάθος).

2.2.1.5 Αποκωδικοποίηση του Reed-Solomon

Οι διαδικασίες αλγεβρικής αποκωδικοποίησης Reed-Solomon μπορούν να διορθώσουν λάθη και διαγραφές. Μια διαγραφή προκύπτει όταν η θέση ενός εσφαλμένου συμβόλου είναι γνωστή. Ένας αποκωδικοποιητής μπορεί να διορθώσει μέχρι και t σφάλματα ή έως και $2t$ διαγραφές. Οι πληροφορίες των διαγραφών συχνά παρέχονται από τον αποδιαμορφωτή

(demodulator) σε ένα σύστημα ψηφιακής επικοινωνίας. Δηλαδή ο αποδιαμορφωτής “σημαδεύει” τα ληφθέντα σύμβολα τα οποία μπορεί να περιέχουν σφάλματα.

Όταν ένα codeword αποκωδικοποιείται, υπάρχουν τρία διαφορετικά πιθανά αποτελέσματα:

1. Αν $2s + r < 2t$ (s είναι τα σφάλματα, r είναι οι διαγραφές) τότε το αρχικό μεταδιδόμενο codeword θα ανακτάται πάντα.
2. Διαφορετικά ο αποκωδικοποιητής θα εντοπίσει ότι δεν μπορεί να ανακτήσει το αρχικό codeword και θα δείξει αυτό το γεγονός.
3. Ή ο αποκωδικοποιητής θα αποκωδικοποιήσει, με λάθος τρόπο, και θα ανακτήσει ένα λάθος codeword χωρίς καμία ένδειξη.

Η πιθανότητα καθεμίας από τις τρεις πιθανότητες εξαρτάται από τον κώδικα Reed-Solomon και από τον αριθμό και την κατανομή των σφαλμάτων.

Το πλεονέκτημα της χρήσης των κωδικών Reed-Solomon είναι ότι η πιθανότητα ενός σφάλματος να παραμείνει στα αποκωδικοποιημένα δεδομένα είναι πολύ μικρότερη από την πιθανότητα όταν δεν χρησιμοποιείται. Αυτό συνήθως περιγράφεται ως **coding gain**. Για παράδειγμα, ένα σύστημα ψηφιακής επικοινωνίας έχει σχεδιαστεί για να λειτουργεί με αναλογία εσφαλμένων bit (**Bit Error Rate – BER**) της τάξης του 10^{-9} , δηλαδή όχι περισσότερα από 1 στα 10^9 bits λαμβάνονται κατά λάθος. Αυτό μπορεί να επιτευχθεί με την ενίσχυση της ισχύος του πομπού ή με την προσθήκη του Reed-Solomon (ή κάποιου διαφορετικού τύπου διόρθωσης σφαλμάτων **Forward Error Correction – FEC**). Ο κώδικας Reed – Solomon επιτρέπει στο σύστημα να επιτύχει αυτό τον στόχο BER με χαμηλότερη ισχύ εξόδου του πομπού. Η “εξοικονόμηση” ισχύος που παρέχεται από τον Reed-Solomon (σε ντεσιμπέλ) είναι το coding gain. (cs.cmu.edu, 1998)

2.2.2 Convolutional Interleaver

Στη συνέχεια ακολουθεί η συνελκτική διεμπλοκή (**Convolutional Interleaver**). Σκοπός της συνελκτικής διεμπλοκής είναι η αύξηση της απόδοσης του κώδικα Reed-Solomon.

Ένας συνελκτικός διεμπλοκείας αποτελείται από ένα σύνολο **shift registers**, καθένας από τους οποίους έχει μια σταθερή καθυστέρηση (delay). Σε έναν τυπικό convolutional interleaver, τα delays είναι μη αρνητικά ακέραια πολλαπλάσια ενός σταθερού ακεραίου, παρόλο που ένας γενικός multiplex interleaver επιτρέπει απεριόριστες τιμές delay. Κάθε νέο σύμβολο από ένα διάνυσμα εισόδου τροφοδοτεί τον επόμενο shift register και το παλαιότερο σύμβολο σε αυτόν τον register γίνεται μέρος του διανύσματος εξόδου. Ένας convolutional interleaver έχει μνήμη, δηλαδή η λειτουργία του δεν εξαρτάται μόνο από τα τρέχοντα σύμβολα αλλά και από τα προηγούμενα.

Παρακάτω μπορούμε να δούμε έναν **Convolutional Interleaver** καθώς και έναν **Deinterleaver** (ο οποίος βρίσκεται στη πλευρά του δέκτη).

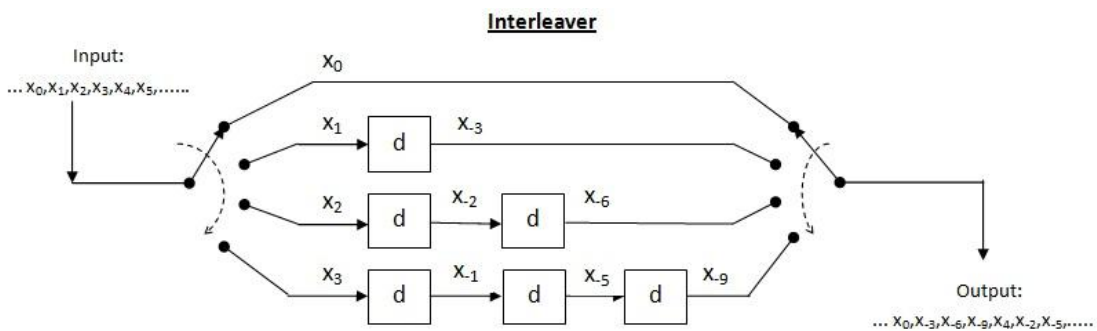


Figure 8 Convolutional Interleaver

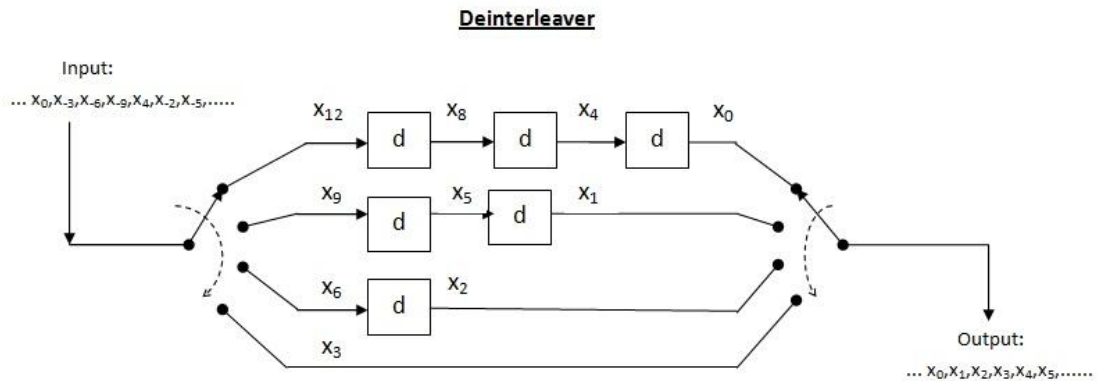


Figure 9 Deinterleaver

Αυτό που γίνεται στον Deinterleaver στην ουσία πρόκειται για την αντίστροφη διαδικασία που γίνεται στον Interleaver. (wikipedia, wikipedia, 2019)

2.2.3 Convolutional Code

Στις τηλεπικοινωνίες ένας συνελκτικός κώδικας (convolutional code) είναι ένας τύπος κώδικα διόρθωσης σφαλμάτων που δημιουργεί σύμβολα ισοτιμίας μέσω της «συρόμενης» εφαρμογής μίας δυαδικής πολυωνυμικής λειτουργίας σε μια ροή δεδομένων. Η «συρόμενη» εφαρμογή αντιπροσωπεύει την «συνέλιξη» του κωδικοποιητή πάνω στα δεδομένα, η οποία δημιουργεί τον όρο «συνελκτική κωδικοποίηση». Η «συρόμενη» φύση του συνελκτικών κωδικών διευκολύνει την αποκωδικοποίηση trellis χρησιμοποιώντας ένα αμετάβλητο trellis. Η αμετάβλητη αποκωδικοποίηση trellis επιτρέπει στους συνελκτικούς κώδικες να αποκωδικοποιούνται με λογική πολυπλοκότητα και με soft-decision.

Η ικανότητα εκτέλεσης μίας αποκωδικοποίησης soft-decision και χαμηλού κόστους είναι ένα από τα κύρια οφέλη των συνελκτικών κωδικών. Αυτό έρχεται σε αντίθεση με τους κλασικούς μπλοκ κώδικες, οι οποίοι γενικά αντιπροσωπεύονται από μεταβλητά trellis και επομένως αποκωδικοποιούνται με hard-decision. Οι συνελκτικοί κώδικες συχνά χαρακτηρίζονται από τον βασικό ρυθμό (code rate) και το βάθος (ή την μνήμη) του κωδικοποιητή $[n, k, K]$. Ο βασικός ρυθμός συνήθως δίνεται ως n/k , όπου n είναι ο ρυθμός δεδομένων εισόδου και k είναι ο ρυθμός συμβόλων εξόδου. Το βάθος συχνά ονομάζεται “μήκος περιορισμού” (constraint length) K , όπου η έξοδος είναι συνάρτηση της τρέχουσας εισόδου καθώς και των προηγούμενων $K-1$ εισόδων.

Οι συνελκτικοί κώδικες συχνά περιγράφονται ως συνεχείς. Ωστόσο, μπορεί επίσης να ειπωθεί ότι οι συνελκτικοί κώδικες έχουν τυχαίο μήκος μπλοκ, αντί να είναι συνεχείς, δεδομένου ότι οι περισσότερες συνελκτικές κωδικοποιήσεις πραγματοποιούνται σε μπλοκ δεδομένων. Οι συνελκτικά κωδικοποιημένοι μπλοκ κώδικες τυπικά χρησιμοποιούν τερματισμό. Το τυχαίο μήκος των μπλοκ των συνελκτικών κωδικών μπορεί επίσης να συγκριθεί με τους κλασικούς μπλοκ κώδικες, οι οποίοι έχουν γενικά σταθερό μήκος μπλοκ που καθορίζεται από αλγεβρικές ιδιότητες.

Ο ρυθμός (code rate) ενός συνελκτικού κώδικα συνήθως τροποποιείται μέσω του puncturing. Για παράδειγμα, ένας συνελκτικός κώδικας με code rate $n/k=1/2$ μπορεί να υποστεί puncture σε μεγαλύτερο ρυθμό, όπως για παράδειγμα $7/8$, απλά με τη μη μετάδοση συμβόλων κώδικα. Η απόδοση ενός punctured συνελκτικού κώδικα γενικά κλιμακώνεται καλά με το ποσό συμβόλων ισοτιμίας που μεταδίδεται. Η ικανότητα εκτέλεσης οικονομικών και soft-decision αποκωδικοποιήσεων σε συνελκτικούς κώδικες, καθώς και η ευελιξία του μήκους των μπλοκ και του code rate των συνελκτικών κωδικών, τους καθιστά ιδιαίτερα δημοφιλείς για τις ψηφιακές επικοινωνίες.

2.2.3.1 Χρήση των convolutional codes

Οι συνελκτικοί κώδικες χρησιμοποιούνται εκτενώς για την επίτευξη αξιόπιστης μεταφοράς δεδομένων σε πολλές εφαρμογές, όπως στο ψηφιακό βίντεο, ράδιο, κινητές επικοινωνίες αλλά και σε δορυφορικές επικοινωνίες. Αυτοί οι κώδικες συχνά χρησιμοποιούνται σε συνδυασμό με έναν hard-decision κώδικα, όπως για παράδειγμα τον Reed-Solomon.

2.2.3.2 Συνελκτική κωδικοποίηση (Convolutional encoding)

Μια συνελκτική κωδικοποίηση, ξεκινάει με k memory registers, καθένας από τους οποίους περιέχει ένα bit εισόδου. Εκτός και αν ορίζεται διαφορετικά, όλοι οι memory registers ξεκινάνε με μηδενική τιμή. Επίσης οι κωδικοποιητές έχουν n modulo-2 adders (ένας modulo-2 adder μπορεί να υλοποιηθεί με ένα απλό Boolean XOR, στο οποίο η λογική είναι: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=0$), και n δημιουργούς πολυωνύμων (**generator polynomials**), ένα για κάθε αθροιστή. Ένα bit εισαγωγής m_1 τροφοδοτείται στον αριστερότερο register. Χρησιμοποιώντας τους generator polynomials και τις υπάρχουσες τιμές στους υπόλοιπους registers, ο κωδικοποιητής εξάγει n σύμβολα. Αυτά τα σύμβολα μπορούν είτε να μεταδοθούν είτε να υποστούν puncturing ανάλογα με το επιθυμητό ρυθμό μετάδοσης. Στη συνέχεια όλα τα bit μεταφέρονται μια θέση δεξιά (το m_1 μεταφέρεται στο m_0 , το m_0 μεταφέρεται στο $m-1$) και ένα καινούργιο bit μπαίνει στο αριστερότερο register. Εάν δεν υπάρχουν άλλα bit εισόδου, ο

κωδικοποιητής συνεχίζει να αλλάζει bit μέχρι όλοι οι registers να επιστρέψουν στη μηδενική κατάσταση.

Για παράδειγμα, στη παρακάτω εικόνα ο ρυθμός μετάδοσης του κωδικοποιητή είναι 1/3 με constraint length (k) 3. Τα generator polynomials είναι $G1(1,1,1)$, $G2(0,1,1)$ και $G3(1,0,1)$. Οπότε, τα bit εξαγωγής υπολογίζονται ως εξής:

- $n1 = m1 + m0 + m-1$
- $n2 = m0 + m-1$
- $n3 = m1 + m-1$

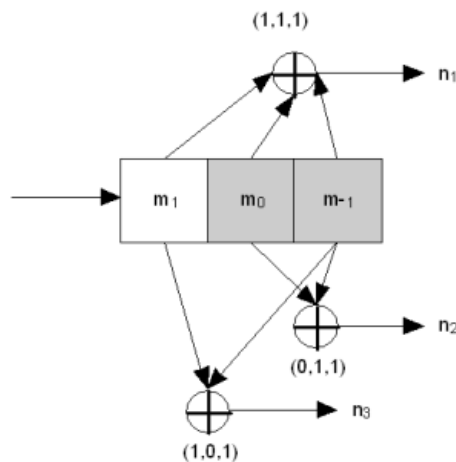


Figure 10 Rate 1/3, convolutional encoder with constraint length 3

Οι convolutional codes μπορούν να είναι συστηματικοί (**systematic**) και μη-συστηματικοί (**non-systematic**):

- Οι συστηματικοί επαναλαμβάνουν τη δομή του μηνύματος πριν το κωδικοποιήσουν
- Οι μη-συστηματικοί αλλάζουν την αρχική δομή

Οι non-systematic convolutional codes είναι πιο δημοφιλείς λόγω της μεγαλύτερης αντοχής στο θόρυβο. Ο κωδικοποιητής στη παραπάνω εικόνα πρόκειται για έναν **non-systematic** και **non-recursive** κωδικοποιητή. Στη συνέχεια θα δούμε τι είναι ένας **recursive** κωδικοποιητής.

2.2.3.3 Recursive and non-recursive codes

Όπως είπαμε και προηγουμένως ο κωδικοποιητής στο Figure 10 πρόκειται για έναν non-recursive. Στη παρακάτω εικόνα έχουμε ένα παράδειγμα ενός **recursive** κωδικοποιητή.

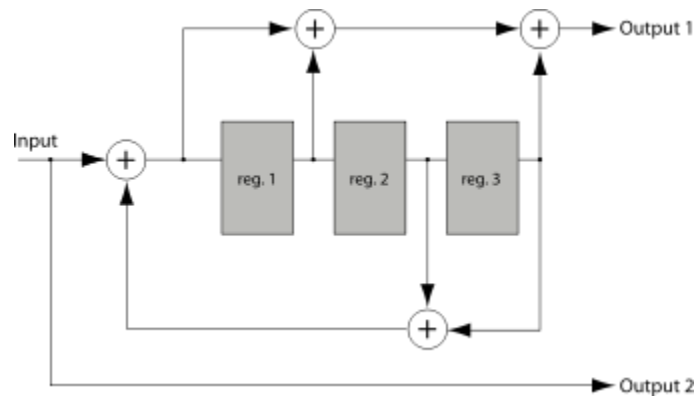


Figure 11 Rate 1/2 8-state recursive systematic convolutional encoder.

Ο κωδικοποιητής στο παράδειγμα πρόκειται για έναν systematic επειδή τα δεδομένα εισόδου χρησιμοποιούνται επίσης και στα σύμβολα εξόδου (output 2). Οι κώδικες που τα σύμβολα εξόδου δεν περιλαμβάνουν τα δεδομένα εισόδου ονομάζονται **non-systematic**.

Οι recursive codes είναι συνήθως systematic και οι non-recursive είναι συνήθως non-systematic. Αυτό δεν αποτελεί έναν αυστηρό κανόνα, αλλά μια κοινή πρακτική.

Οι recursive systematic κώδικες (RSC) έχουν γίνει πιο δημοφιλείς λόγω της χρήσης τους σε Turbo κώδικες. Οι recursive systematic κώδικες επίσης είναι γνωστοί και ως pseudo-systematic.

2.2.3.4 Διαγράμματα Trellis

Ένας convolutional encoder είναι μια μηχανή πεπερασμένων καταστάσεων. Ένας κωδικοποιητής με n binary cells θα έχουν 2^n καταστάσεις.

Για παράδειγμα ο encoder που είδαμε παραπάνω στο Figure 10, έχει την τιμή '1' στον αριστερό register (m0) και '0' στον δεξιό (m-1). Ο m1 δεν είναι register αλλά πρόκειται για το input στον πρώτο register (m0). Θα ορίσουμε μια κατάσταση ως '10'. Σύμφωνα με ένα bit εισόδου, ο κωδικοποιητής στον επόμενο γύρο μπορεί να το μετατρέψει στη κατάσταση '01' ή στην '11'. Δεν μπορούν να γίνουν όλες οι μετατροπές. Δηλαδή ένας αποκωδικοποιητής δεν μπορεί να κάνει τη μετατροπή από τη κατάσταση '10' στη κατάσταση '00' ή ακόμα και να παραμείνει στη κατάσταση '10'.

Στην παρακάτω εικόνα μπορούν να φανούν όλες οι πιθανές μεταβάσεις.

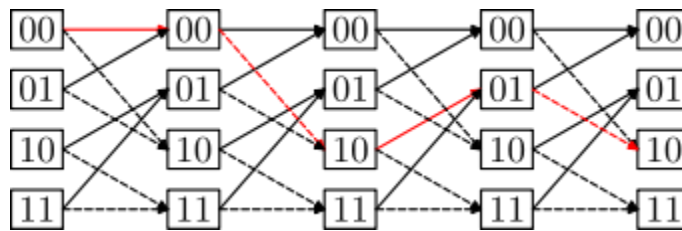


Figure 12 Trellis diagram

Μία κωδικοποιημένη ακολουθία μπορεί να αναπαρασταθεί ως μία διαδρομή σε αυτό το γράφημα. Για παράδειγμα μια έγκυρη διαδρομή εμφανίζεται με κόκκινο χρώμα.

Αυτό το διάγραμμα μας δίνει μια ιδέα σχετικά με την αποκωδικοποίηση. Αν μία ληφθείσα ακολουθία δεν ταιριάζει σε αυτό το διάγραμμα, τότε ελήφθη με σφάλματα, και πρέπει να διαλέξουμε τη κοντινότερη σωστή ακολουθία. Οι πραγματικοί αλγόριθμοι αποκωδικοποίησης εκμεταλλεύονται αυτήν την ιδέα.

2.2.3.5 Αποκωδικοποίηση συνελκτικών κωδικών

Υπάρχουν πολλοί αλγόριθμοι για την αποκωδικοποίηση των συνελκτικών κωδικών. Για σχετικά μικρές τιμές k , ο αλγόριθμος **Viterbi** χρησιμοποιείται καθολικά καθώς παρέχει απόδοση μέγιστης πιθανοφάνειας και είναι εξαιρετικά παραλληλισμένος. Οι αποκωδικοποιητές Viterbi

είναι επομένως εύκολο να εφαρμοστούν σε υλικό VLSI και σε λογισμικό CPU's με σετ εντολών SIMD.

Οι κώδικες με μεγαλύτερο **constraint length** αποκωδικοποιούνται με οποιοδήποτε από τους διαφόρους διαδοχικούς αλγορίθμους αποκωδικοποίησης, από τους οποίους ο αλγόριθμος Fano είναι ο πιο γνωστός. Σε αντίθεση με την αποκωδικοποίηση Viterbi, η διαδοχική αποκωδικοποίηση δεν έχει μέγιστη πιθανοφάνεια, αλλά η πολυπλοκότητά της αυξάνεται ελαφρώς με το constraint length, επιτρέποντας τη χρήση ισχυρών long constraint length codes.

Τόσο οι διαδοχικοί αλγόριθμοι αποκωδικοποίησης όσο και ο αλγόριθμος Viterbi επιστρέφουν hard decisions (binary – τα bits που σχηματίζουν το πιθανό codeword).

2.2.3.6 Γνωστοί convolutional codes

Στη πραγματικότητα οι convolutional codes που χρησιμοποιούνται στη βιομηχανία είναι κώδικες που φτιάχτηκαν κατά τη διάρκεια επιστημονικών ερευνών. Αυτό σχετίζεται με τη δυνατότητα επιλογής καταστροφικών convolutional codes οι οποίοι προκαλούν μεγάλο αριθμό σφαλμάτων.

Ένας ιδιαίτερα δημοφιλής convolutional code, χρησιμοποιείται στο πρόγραμμα Voyager, ο οποίος έχει constraint length=7 και rate=1/2. Αυτόν το convolutional code μπορούμε να τον δούμε στη παρακάτω εικόνα.

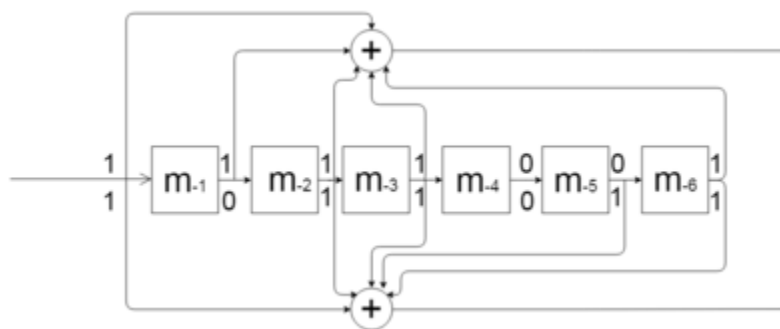


Figure 13 Convolutional Code (7,[171 133])

Ο συγκεκριμένος convolutional code έχει constraint length 7 (6 shift registers + 1input) και για polynomial generators έχει, στο οκταδικό, τους αριθμούς 171 και 133 (171=[1111001] και 133=[1011011]). Επίσης έχει code rate=1/2 γιατί έχουμε 1 είσοδο και 2 εξόδους.

2.2.3.7 Punctured convolutional codes

Οι convolutional codes με οποιοδήποτε ρυθμό μπορούν να σχεδιαστούν σύμφωνα με τη πολυωνυμική επιλογή. Ωστόσο, στη πράξη, συχνά χρησιμοποιείται μια διαδικασία puncturing προκειμένου να επιτύχουμε το επιθυμητό code rate. Το puncturing είναι μια τεχνική που χρησιμοποιείται για τη δημιουργία ενός code rate m/n από έναν βασικό κώδικα χαμηλού ρυθμού. Αυτό επιτυγχάνεται διαγράφοντας μερικά bits στην έξοδο του κωδικοποιητή. Τα bits διαγράφονται σύμφωνα με ένα προκαθορισμένο puncturing matrix. Στον επόμενο πίνακα μπορούμε να δούμε αυτά που χρησιμοποιούνται πιο συχνά.

Code rate	Puncturing matrix
1/2	1 1
2/3	1 0 1 1
3/4	1 0 1 1 1 0
5/6	1 0 1 0 1 1 1 0 1 0
7/8	1 0 0 0 1 0 1 1 1 1 1 0 1 0

Για παράδειγμα, εάν θέλουμε να έχουμε code **rate=2/3** χρησιμοποιώντας τον κατάλληλο puncturing από το παραπάνω πίνακα, θα πρέπει να πάρουμε μία βασική έξοδο κωδικοποιητή και να μεταδώσουμε το πρώτο bit από το από το πρώτη σειρά και όλα από τη δεύτερη σειρά. Η σειρά μετάδοσης καθορίζεται από το αντίστοιχο πρότυπο.

Οι punctured convolutional codes χρησιμοποιούνται ευρέως στις δορυφορικές επικοινωνίες καθώς και στη ψηφιακή μετάδοση βίντεο (Digital Video Broadcasting). (Wikipedia, Wikipedia, 2020)

2.2.4 Viterbi decoder

Ένας αποκωδικοποιητής Viterbi χρησιμοποιεί τον αλγόριθμο Viterbi για την αποκωδικοποίηση ενός codeword το οποίο κωδικοποιήθηκε με τη χρήση ενός convolutional code ή ενός trellis code.

Υπάρχουν και άλλοι αλγόριθμοι για την αποκωδικοποίηση ενός stream που κωδικοποιήθηκε από έναν convolutional code, όπως για παράδειγμα ο αλγόριθμος Fano. Ο αλγόριθμος Viterbi είναι ο πιο απαιτητικός όσο αναφορά τους πόρους σε σχέση με τους άλλους, αλλά κάνει αποκωδικοποίηση μέγιστης πιθανοφάνειας. Χρησιμοποιείται συχνότερα για την αποκωδικοποίηση των convolutional codes με constraint length 3 ή λιγότερα, αλλά στη πράξη χρησιμοποιούνται και με constraint length έως και 15. Η αποκωδικοποίηση Viterbi μπορεί να υλοποιηθεί τόσο σε υλικό όσο και σε λογισμικό.

Traceback

Η γενική προσέγγιση στο traceback είναι η συσσώρευση μετρήσεων διαδρομής έως και 5 φορές το constraint length ($5-(K-1)$), η εύρεση του κόμβου με το μεγαλύτερο συσσωρευμένο κόστος και η έναρξη του traceback από αυτόν τον κόμβο.

Ωστόσο ο υπολογισμός του κόμβου με το μεγαλύτερο συσσωρευμένο κόστος (είτε το μεγαλύτερο είτε το μικρότερο ολοκληρωμένο μονοπάτι) περιλαμβάνει την εύρεση ελάχιστων ή μέγιστων διαφορών αριθμών (συνήθως $2K-1$), η οποία μπορεί να είναι χρονοβόρα όταν υλοποιείται σε ενσωματωμένα συστήματα υλικού.

Τα περισσότερα συστήματα επικοινωνίας χρησιμοποιούν την αποκωδικοποίηση Viterbi που περιλαμβάνει πακέτα δεδομένων σταθερών μεγεθών, με σταθερό μοτίβο bit/byte είτε στη

αρχή ή/και στο τέλος του πακέτου δεδομένων. Χρησιμοποιώντας σαν αναφορά το γνωστό μοτίβο bit/byte, ο κόμβος έναρξης μπορεί να οριστεί σε μία σταθερή τιμή, αποκτώντας έτσι ένα μονοπάτι μέγιστης πιθανοφάνειας (Maximum Likelihood Path) κατά τη διάρκεια του traceback. (wikipedia, wikipedia, 2020)

2.2.5 Block Interleaver

Το interleaving χρησιμοποιείται για τη μετατροπή των συνελικτικών κωδικών (convolutional codes) από διορθωτές τυχαίων σφαλμάτων σε διορθωτές σφάλματος ριπής. Η βασική ιδέα πίσω από τη χρήση των interleaved κωδικών είναι να ανακατέψει τα σύμβολα στον δέκτη. Αυτό οδηγεί στη τυχαιοποίηση των ριπών των λαμβανόμενων σφαλμάτων που βρίσκονται πολύ κοντά. Έτσι, η κύρια λειτουργία ενός interleaver στο πομπό είναι να αλλάξει την ακολουθία των συμβόλων εισόδου. Στον δέκτη, ο deinterleaver θα αλλάξει τη ληφθείσα ακολουθία έτσι ώστε να πάρει την αρχική αναλλοίωτη ακολουθία που έστειλε ο πομπός.

Ένας interleaver είναι ο **block interleaver** τον οποίο μπορούμε να δούμε στη παρακάτω εικόνα:

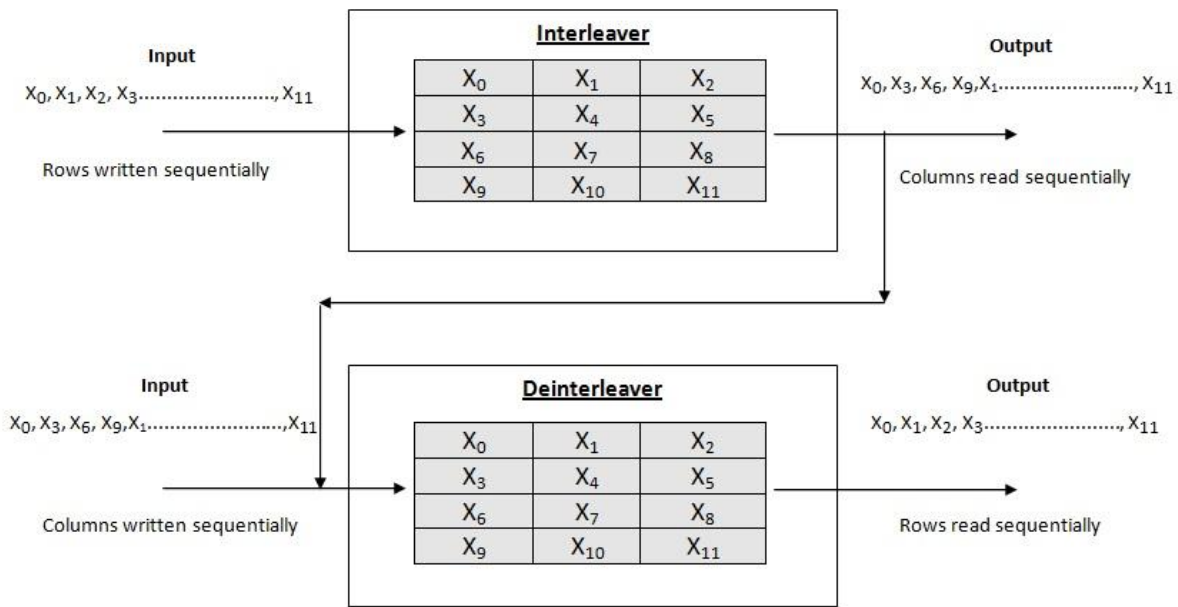


Fig. A.4 X 3 Interleaver and Deinterleaver

Figure 14 Block Interleaver & Deinterleaver

Στον παραπάνω block interleaver, τα σύμβολα εισόδου γράφονται διαδοχικά σε γραμμές και τα σύμβολα εξόδου λαμβάνονται διαβάζοντας διαδοχικά τις στήλες. Έτσι αυτό είναι της μορφής ενός πίνακα $\mathbf{M} \times \mathbf{N}$. Γενικά, το \mathbf{N} είναι το μήκος του codeword. Επίσης ο παραπάνω block interleaver πρόκειται για έναν που έχει Column – major order ενώ ο deinterleaver έχει Row – major order. Στον interleaver τα σύμβολα εισόδου μπαίνουν και αποθηκεύονται στις ανάλογες θέσεις και αναλόγως του interleaver βγαίνουν με μια προκαθορισμένη σειρά. Αυτό είναι και λογικό γιατί η δουλειά του deinterleaver είναι να γυρίσει την ληφθείσα ακολουθία στην αρχική της κατάσταση. Στη συνέχεια ακολουθεί μια εικόνα με 2 διαφορετικούς τύπους των block interleavers.

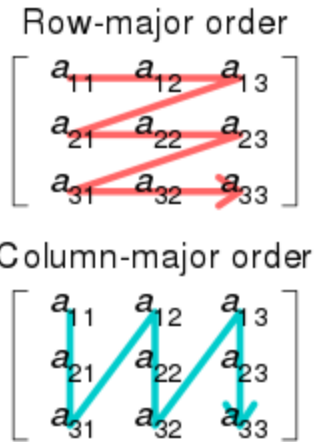


Figure 15 Row & Column major order

Όμως ο block interleaver έχει και κάποια μειονεκτήματα. Όπως μπορούμε να δούμε από την εικόνα (Figure 14) οι στήλες διαβάζονται διαδοχικά, ο δέκτης μπορεί να ερμηνεύσει μια σειρά μόνο όταν λάβει το ολοκληρωμένο μήνυμα και όχι πριν από αυτό. Επίσης, ο δέκτης απαιτεί ένα σημαντικό ποσό μνήμης έτσι ώστε να αποθηκεύσει τα ληφθέντα σύμβολα, και πρέπει να αποθηκεύσει ολόκληρο το μήνυμα. Έτσι αυτοί οι παράγοντες δημιουργούν δύο μειονεκτήματα, το ένα είναι το latency και το άλλο είναι ο αποθηκευτικός χώρος (αρκετά μεγάλη ποσότητα μνήμης). Αυτά τα μειονεκτήματα μπορούμε να τα αποφύγουμε με τη χρήση ενός convolutional interleaver για τον οποίο μιλήσαμε προηγουμένως. (wikipedia, wikipedia, 2019)

2.2.6 Constellation Diagram

Ένα constellation diagram (διάγραμμα αστερισμού) είναι μια αναπαράσταση ενός σήματος, το οποίο έχει διαμορφωθεί από ένα σχήμα ψηφιακής διαμόρφωσης όπως το **quadrature amplitude modulation** (QAM) ή το **phase – shift keying** (PSK). Εμφανίζει το σήμα ως ένα δισδιάστατο διάγραμμα διασποράς xy σε στιγμές δειγματοληψίας συμβόλων. Η γωνία ενός σημείου μετρούμενη αριστερόστροφα από τον οριζόντιο άξονα x , αντιπροσωπεύει τη μετατόπιση φάσης του κύματος φορέα από μια φάση αναφοράς. Η απόσταση ενός σημείου από

το σημείο 0 που ενώνονται οι άξονες x και y, αντιπροσωπεύει το πλάτος ή την ισχύ του σήματος.

Σε ένα σύστημα ψηφιακής διαμόρφωσης, οι πληροφορίες μεταδίδονται σαν μια σειρά δειγμάτων, καθένα από τα οποία πιάνουν ένα ομοιόμορφο time slot. Κατά τη διάρκεια κάθε δείγματος, το φέρον κύμα έχει ένα σταθερό πλάτος και φάση, τα οποία περιορίζονται σε έναν πεπερασμένο αριθμό τιμών. Έτσι κάθε δείγμα κωδικοποιεί ένα από τον πεπερασμένο αριθμό συμβόλων, τα οποία με τη σειρά τους αντιπροσωπεύουν ένα ή περισσότερα δυαδικά ψηφία (bits) πληροφορίας. Κάθε σύμβολο κωδικοποιείται ως έναν διαφορετικό συνδυασμό πλάτους και φάσης του φορέα, έτσι κάθε σύμβολο αντιπροσωπεύεται από ένα σημείο στο διάγραμμα αστερισμού (constellation diagram), το οποίο ονομάζεται σημείο αστερισμού. Το διάγραμμα αστερισμού δείχνει όλα τα πιθανά σύμβολα τα οποία μπορούν να μεταδοθούν από το σύστημα ως μια συλλογή σημείων. Σε ένα σήμα με διαμόρφωση φάσης ή συχνότητας, το πλάτος του σήματος είναι σταθερό, έτσι τα σημεία βρίσκονται σε ένα έναν κύκλο γύρω από το σημείο που ενώνεται ο άξονας x με τον y.

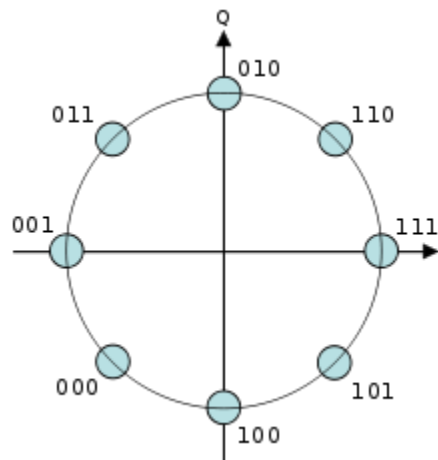


Figure 16 8 – PSK

Ο αριθμός των σημείων αστερισμού σε ένα διάγραμμα δίνει το μέγεθος του "αλφαβήτου" των συμβόλων των οποίων μπορούν να μεταδοθούν από κάθε δείγμα, και έτσι καθορίζει τον αριθμό των bits που μεταδίδονται ανά δείγμα. Συνήθως είναι δύναμη του 2. Για παράδειγμα, ένα διάγραμμα με τέσσερα σημεία αντιπροσωπεύει ένα σχήμα διαμόρφωσης που μπορεί να

κωδικοποιήσει χωριστά και τους 4 συνδυασμούς δύο δυαδικών ψηφίων: 00, 01, 10 και 11 και έτσι μπορεί να μεταδώσει δύο ψηφία ανά δείγμα.

Αφού περάσει από το κανάλι επικοινωνίας, το σήμα αποκωδικοποιείται από ένα αποδιαμορφωτή. Η λειτουργία του αποδιαμορφωτή είναι να ταξινομήσει κάθε δείγμα ως σύμβολο. Το σύνολο των τιμών δείγματος που ο αποδιαμορφωτής ταξινομεί ως δεδομένο σύμβολο μπορεί να αναπαρασταθεί από μια περιοχή στο επίπεδο που σχεδιάζεται γύρω από κάθε σημείο αστερισμού. Εάν ο θόρυβος προκαλέσει το σημείο, που αναπαριστά ένα δείγμα, να απομακρυνθεί προς την περιοχή που αναπαριστά άλλο σύμβολο, ο αποδιαμορφωτής θα αναγνωρίσει εσφαλμένα αυτό το δείγμα ως άλλο σύμβολο, με αποτέλεσμα να έχουμε ένα σφάλμα. Οι περισσότεροι αποδιαμορφωτές επιλέγουν, ως εκτίμηση του τι πράγματι μεταδόθηκε, το σημείο αστερισμού που είναι πιο κοντά με αυτό του ληφθέντος δείγματος. Στο διάγραμμα αστερισμού αυτές οι περιοχές ανίχνευσης μπορούν εύκολα να αναπαρασταθούν διαιρώντας το επίπεδο με γραμμές ίσες από κάθε γειτονικό ζεύγος σημείων.

Το μισό της απόστασης μεταξύ κάθε ζεύγους γειτονικών σημείων είναι το εύρος του πρόσθετου θορύβου ή παραμόρφωσης που απαιτείται για να προκαλέσει ένα από τα σημεία να αναγνωριστεί σαν κάποιο άλλο, και συνεπώς να προκαλέσει ένα σφάλμα. Επομένως, όσο πιο μακριά βρίσκονται τα σημεία μεταξύ τους τόσο πιο μεγάλη είναι η αντοχή του διαμορφωτή στο θόρυβο. Τα πρακτικά συστήματα διαμόρφωσης έχουν σχεδιαστεί για να μεγιστοποιούν τον ελάχιστο θόρυβο που απαιτείται για τη δημιουργία σφάλματος. Στο διάγραμμα αστερισμού αυτό σημαίνει ότι η απόσταση ανάμεσα σε κάθε ζεύγος γειτονικών σημείων είναι ίση.

2.2.6.1 Digital Quadrature Amplitude Modulation (QAM)

Όπως σε πολλά σχήματα ψηφιακής διαμόρφωσης, το διάγραμμα αστερισμού είναι χρήσιμο για το QAM. Στο QAM, τα σημεία αστερισμού είναι συνήθως διατεταγμένα σε ένα τετράγωνο πλέγμα με ίση κάθετη και οριζόντια απόσταση, αν και είναι δυνατές και άλλες διαμορφώσεις. Δεδομένου ότι στις ψηφιακές τηλεπικοινωνίες τα δεδομένα είναι συνήθως δυαδικά, ο αριθμός σημείων στο πλέγμα είναι συνήθως δύναμη του 2 (2, 4, 8, ...). Δεδομένου ότι το QAM είναι συνήθως τετράγωνο, οι πιο συνηθισμένες μορφές είναι οι 16-QAM, 64-QAM

και 256-QAM. Πηγαίνοντας σε ένα υψηλότερης τάξης αστερισμό, είναι δυνατή η μετάδοση περισσότερων bits ανά σύμβολο. Ωστόσο εάν η μέση ενέργεια του αστερισμού πρέπει να παραμένει η ίδια, τα σημεία πρέπει να είναι πιο κοντά μεταξύ τους και επομένως είναι πιο ευαίσθητα στον θόρυβο. Αυτό έχει ως αποτέλεσμα να έχουμε ένα μεγαλύτερο bit error rate (ber). Έτσι ένα QAM μεγαλύτερης τάξης μπορεί να παρέχει περισσότερα δεδομένα αλλά με λιγότερη αξιοπιστία σε σχέση με ένα QAM μικρότερης τάξης. Για τη χρήση QAM μεγαλύτερης τάξης χωρίς να γίνει αύξηση του BER απαιτείται ένα υψηλότερο signal – to – noise ratio (SNR) αυξάνοντας την ενέργεια του σήματος.

Οι διαμορφώσεις που χρησιμοποιούνται στην τηλεόραση είναι οι 16-QAM, 64-QAM και 256-QAM. Στη παρακάτω φαίνονται κάποια παραδείγματα:

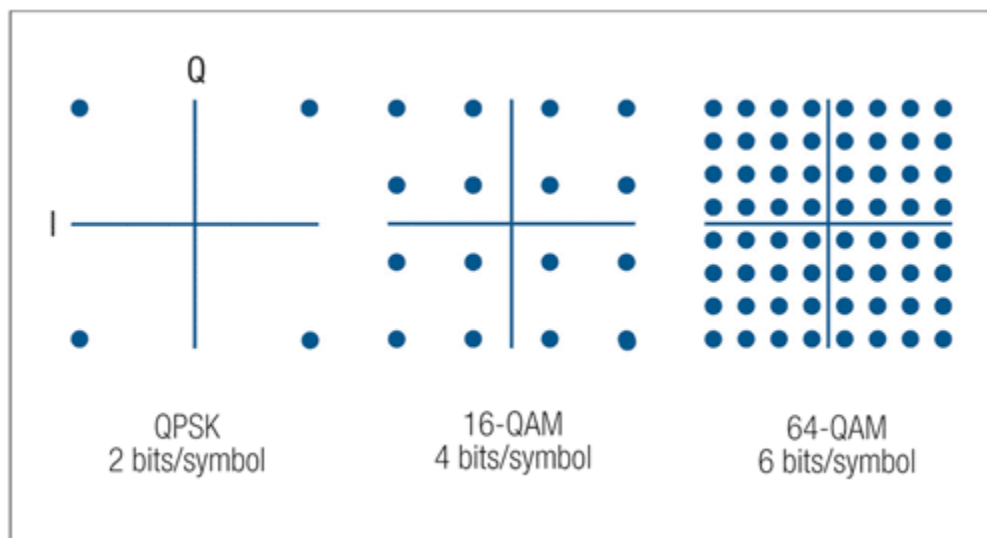


Figure 17 Modulation Schemes

2.2.6.2 Quadrature phase – shift keying (QPSK)

Το phase – shift keying (PSK) είναι μια διαδικασία ψηφιακής διαμόρφωσης που μεταφέρει δεδομένα αλλάζοντας (διαμορφώνοντας) τη φάση ενός σήματος αναφοράς με σταθερή συχνότητα. Το QPSK είναι γνωστό και ως τετραπλή φάση PSK, 4-PSK ή 4-QAM (παρόλο που οι βασικές έννοιες του QPSK και του 4-QAM είναι διαφορετικές, τα

διαμορφωμένα ραδιοκύματα που προκύπτουν είναι ακριβώς τα ίδια). Το QPSK χρησιμοποιεί 4 σημεία στο διάγραμμα αστερισμού, σε απόσταση γύρω από έναν κύκλο. Με τέσσερις φάσεις, το QPSK μπορεί να κωδικοποιήσει δυο bit ανά σύμβολο.

Η μαθηματική ανάλυση δείχνει ότι το QPSK μπορεί να χρησιμοποιηθεί για να διπλασιάσει τον ρυθμό δεδομένων σε σύγκριση με έναν BPSK (binary phase-shift keying), διατηρώντας ταυτόχρονα το ίδιο εύρος ζώνης του σήματος. Επίσης μπορεί να χρησιμοποιηθεί για να διατηρήσει τον ρυθμό δεδομένων του BPSK αλλά μειώνοντας στο μισό το εύρος ζώνης που απαιτείται. Στη δεύτερη περίπτωση, το BER του QPSK είναι ακριβώς το ίδιο με το BER του BPSK.

Το QPSK μεταδίδει διπλάσιο τον ρυθμό δεδομένων σε ένα δεδομένο εύρος ζώνης σε σχέση με το BPSK στο ίδιο BER. Ένα αρνητικό είναι ότι οι πομποί και οι δέκτες QPSK είναι πιο περίπλοκοι από αυτούς του BPSK.

2.2.7 Orthogonal frequency – division multiplexing

Στις τηλεπικοινωνίες το orthogonal frequency – division multiplexing (OFDM) είναι ένας τύπος ψηφιακής μετάδοσης και μια μέθοδος κωδικοποίησης ψηφιακών δεδομένων σε πολλαπλές συχνότητες. Το OFDM έχει εξελιχθεί σε ένα δημοφιλές σχήμα για ψηφιακή επικοινωνία ευρείας ζώνης, που χρησιμοποιείται στη ψηφιακή τηλεόραση, στη μετάδοση ήχου, ασύρματα δίκτυα DSL και σε άλλα.

Το OFDM είναι ένα frequency-division multiplexing (FDM) που χρησιμοποιείται ως μέθοδος διαμόρφωσης πολλαπλών φορέων. Στον OFDM, μεταδίδονται πολλαπλά ορθογώνια σήματα υπό-φέροντος σε απόσταση μεταξύ τους με επικαλυπτόμενα φάσματα για να μεταφέρουν δεδομένα παράλληλα. Η αποδιαμόρφωση βασίζεται σε αλγόριθμους Fast Fourier Transform. Το OFDM βελτιώθηκε με την εισαγωγή ενός διαστήματος φρουράς (guard interval), παρέχοντας καλύτερη ορθογωνικότητα στα κανάλια μετάδοσης που επηρεάζονται από τη διάδοση πολλαπλών διαδρομών. Κάθε σήμα διαμορφώνεται με ένα συμβατικό σχήμα διαμόρφωσης (όπως το QAM ή το QPSK) με χαμηλό ρυθμό συμβόλων.

Στις τηλεπικοινωνίες χρησιμοποιούνται τα διαστήματα φύλαξης (guard intervals) για να διασφαλιστεί ότι οι διακριτές μεταδόσεις δεν παρεμβαίνουν μεταξύ τους ή αλλιώς δεν πέφτει η μία πάνω στην άλλη. Ο σκοπός του διαστήματος φύλαξης (guard interval) είναι να εισάγει ανοσία σε ηχώ, καθυστέρηση διάδοσης και αντανάκλασεις στα οποία ένα σήμα είναι αρκετά ευαίσθητο.

Στο OFDM, η αρχή κάθε συμβόλου προηγείται ενός διαστήματος προστασίας. Όσο οι ηχώ εμπίπτουν σε αυτό το διάστημα δεν επηρεάζουν την ικανότητα του δέκτη να αποκωδικοποιεί με ασφάλεια τα δεδομένα, αφού τα δεδομένα ερμηνεύονται μόνο εκτός του διαστήματος προστασίας. Τα μεγαλύτερα διαστήματα προστασίας επιτρέπουν την προστασία από μεγαλύτερες ηχώ. Ωστόσο τα μεγαλύτερα διαστήματα φύλαξης μειώνουν τη απόδοση του καναλιού. Για παράδειγμα, στο DVB-T, διατίθενται τέσσερα διαστήματα προστασίας, $1/32$, $1/16$, $1/8$ και $1/4$ (δίνονται ως κλάσματα μιας περιόδου συμβόλων). Έτσι το $1/32$ παρέχει τη χαμηλότερη προστασία και τον υψηλότερο ρυθμό δεδομένων ενώ το $1/4$ παρέχει καλύτερη προστασία αλλά με τον χαμηλότερο ρυθμό δεδομένων.

2.4 Το πρωτόκολλο DVB-T2

Εκτός από το DVB-T για την επίγεια τηλεόραση υπάρχει και το DVB-T2. Το DVB-T2 (Digital Video Broadcasting – Second Generation Terrestrial) είναι δεύτερης γενιάς πρωτόκολλο για την ψηφιακή μετάδοση βίντεο. Πρόκειται για την επέκταση του τηλεοπτικού προτύπου DVB-T και σχεδιάστηκε τη μετάδοση ψηφιακής επίγειας τηλεόρασης.

Αυτό το σύστημα μεταδίδει συμπίεσμένο ψηφιακό ήχο, βίντεο και άλλα δεδομένα με τη χρήση της διαμόρφωσης OFDM, κωδικοποίηση καναλιών και interleaving. Σε σχέση με το DVB-T έχει υψηλότερο ρυθμό μετάδοσης bit και αυτό το καθιστά ένα σύστημα κατάλληλο για τη μεταφορά σημάτων υψηλής ευκρίνειας (HDTV) στο επίγειο τηλεοπτικό κανάλι αν και πολλοί ραδιοτηλεοπτικοί φορείς εξακολουθούν να χρησιμοποιούν το απλό DVB-T.

2.4.1 Διαφορές του DVB-T με το DVB-T2

Σε αντίθεση με το DVB-T που έχει μόνο ένα transport stream το DVB-T2 μπορεί να μεταδώσει πολλαπλά transport stream. Επίσης το DVB-T2 χρησιμοποιεί το Generic Stream Encapsulation (GSE). Το GSE πρόκειται για ένα πρωτόκολλο επιπέδου σύνδεσης που ορίζεται από το DVB. Το GSE παρέχει μέσα για τη μεταφορά πρωτοκόλλων προσανατολισμένων σε πακέτα, όπως για παράδειγμα την IP πάνω από μονοκατευθυντικά φυσικά στρώματα όπως είναι τα DVB-T2 και DVB-S2. Επίσης το DVB-T χρησιμοποιεί σταθερή κωδικοποίηση και διαμόρφωση σε σχέση με το DVB-T2 που χρησιμοποιεί μεταβλητή.

Από τη πλευρά των FEC (Forward Error Correction) αλγορίθμων το DVB-T χρησιμοποιεί έναν Reed-Solomon και έναν convolutional κωδικοποιητή από τον οποίο ανάλογα με το puncture code που χρησιμοποιούμε προκύπτουν οι ρυθμοί $1/2$, $2/3$, $3/4$, $5/6$ και $7/8$. Αντιθέτως το DVB-T2 χρησιμοποιεί έναν low-density parity-check (LDPC) και έναν BCH κώδικα και οι ρυθμοί που μπορεί να έχει είναι $1/2$, $3/5$, $2/3$, $3/4$, $4/5$, $5/6$, $6/7$ και $7/8$.

Η διαμόρφωση που χρησιμοποιούν και στα δύο είναι η OFDM. Τόσο το DVB-T όσο και το DVB-T2 χρησιμοποιούν τις διαμορφώσεις QPSK, 16QAM, 64QAM αλλά το DVB-T2 χρησιμοποιεί επιπλέον και τη διαμόρφωση 256QAM. Τέλος το DVB-T μπορεί να έχει 2k και 8k mode ενώ το DVB-T2 έχει 1k, 2k, 4k, 8k, 16k και 32k.

ΚΕΦΑΛΑΙΟ 3: ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΩΤΟΚΟΛΛΟΥ DVB-T ΜΕΣΩ ΤΟΥ MATLAB

Σε αυτό το κεφάλαιο θα αναλύσουμε το πρωτόκολλο DVB-T μέσω της προσομοίωσης που υπάρχει στο Matlab. Στη παρακάτω εικόνα φαίνεται η προσομοίωση του πρωτόκολλου DVB-T στο Matlab.

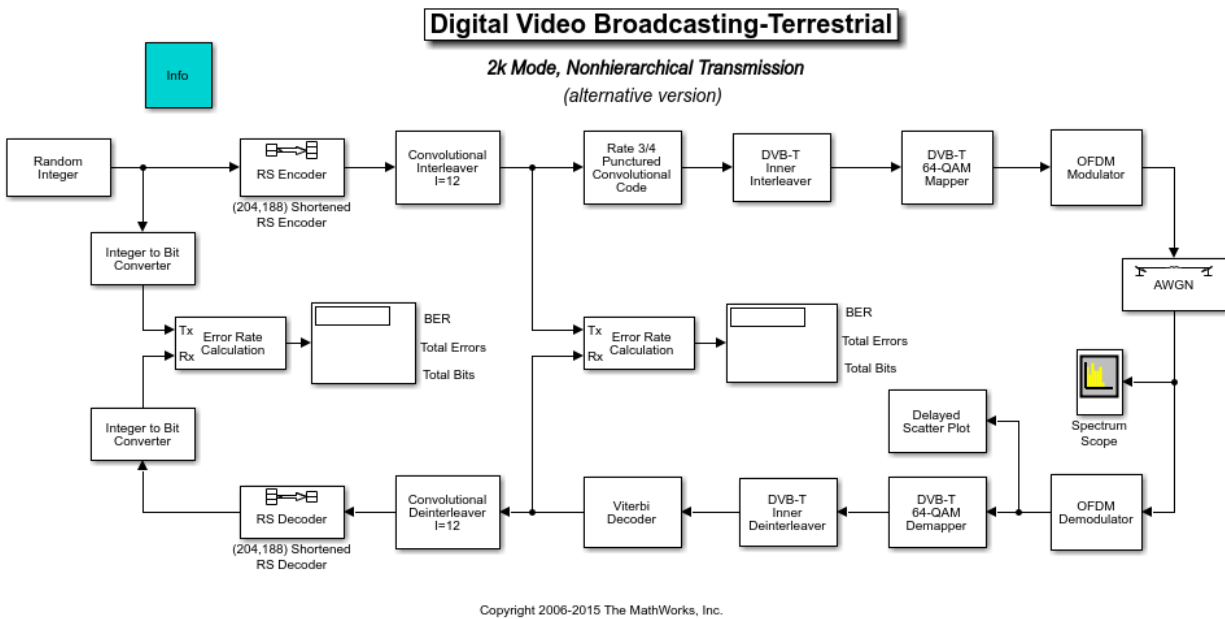


Figure 18 DVB-T protocol

Για κάθε ένα από τα παραπάνω μπλοκ θα αναλύσουμε από ποιες μεταβλητές αποτελείται και τι κάνει η κάθε μία στο πρωτόκολλο.

3.1 Random-integer Generator

Το πρώτο μπλοκ είναι ένας random integer generator και αντιπροσωπεύει το σήμα μας. Αυτό που κάνει είναι να παράγει τυχαίους ακέραιους αριθμούς από 0 έως M-1 όπου M είναι το **set size**. Στο **source of initial seed** επιλέγουμε τον τρόπο που θα μπει το initial seed. Υπάρχουν δύο διαφορετικοί τρόποι, το **Parameter** με το οποίο το βάζουμε εμείς και το **Auto** με το οποίο μπαίνει αυτόματα. Το **Initial seed** είναι μία ακολουθία αριθμών από τους οποίους εξαρτάται το πόσο τυχαίο θα είναι οι αριθμοί. Το **Sample time** δηλώνει το χρονικό διάστημα που χρειάζεται

για να φτιαχτεί ένα frame. Το **Samples per frame** μας δείχνει πόσα samples πρέπει να φτιαχτούν προκειμένου να τα στείλει στο επόμενο μπλοκ. Τέλος στο **Output data type** δηλώνουμε σε τι μορφή θέλουμε να είναι οι αριθμοί που θα βγουν.

Στη παρακάτω εικόνα θα μπορούμε να δούμε τις μεταβλητές που μπορούμε να βάλουμε.

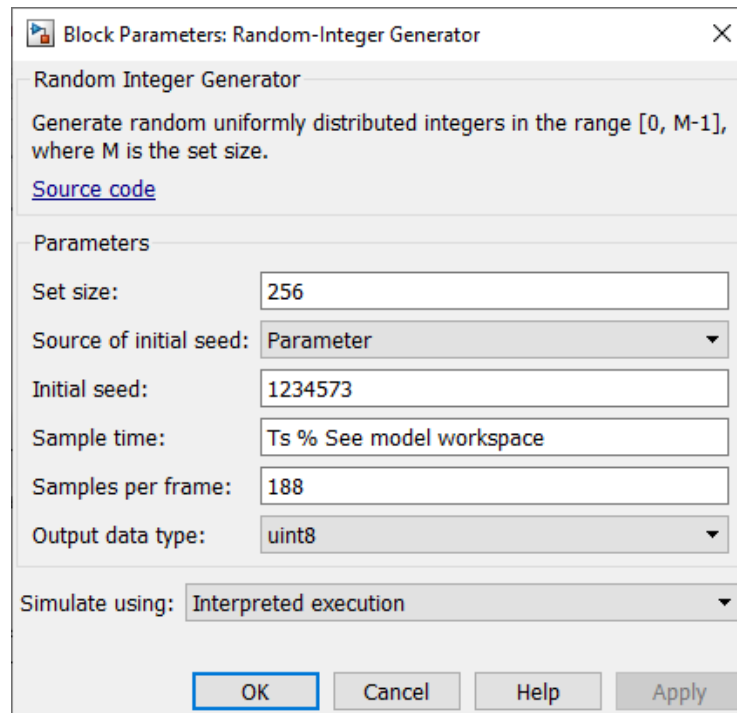


Figure 19 Random-Integer Generator

Το Set size το βάζουμε 256 γιατί θέλουμε οι αριθμοί που θα βγουν να είναι από 0 έως 255. Το Initial seed είναι ένας τυχαίος αριθμός ο οποίος έχει δοθεί από το standard του DVB-T όπως το ίδιο ισχύει και για το Sample time. Στο samples per frame βάζουμε 188 γιατί τόσους αριθμούς θέλουμε να στείλουμε στο επόμενο μπλοκ οι οποίοι θα είναι μέχρι 8 bit.

3.2 Reed-Solomon encoder και decoder

Το επόμενο μπλοκ είναι το μπλοκ του κωδικοποιητή Reed-Solomon στο οποίο δηλώνουμε ποιον Reed-Solomon θα χρησιμοποιήσουμε για τη κωδικοποίηση του σήματος μας.

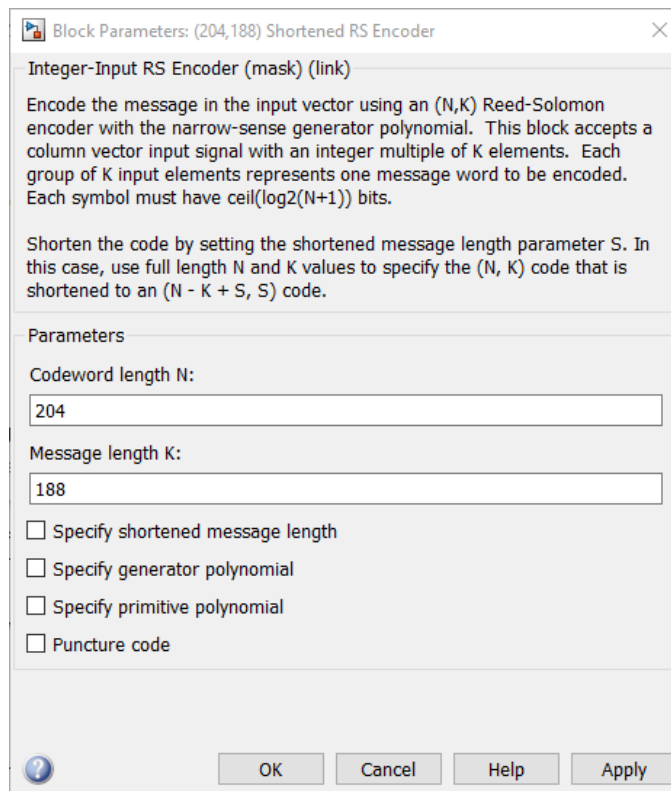


Figure 20 Reed-Solomon Encoder

Στο **Codeword length N** δηλώνουμε το μήκος του codeword και στο **Message length K** δηλώνουμε το μήκος του μηνύματος. Μπορούμε να τον κάνουμε shortened επιλέγοντας το **Specify shortened message length** και επίσης μπορούμε να χρησιμοποιήσουμε κάποιο **puncture code** επιλέγοντάς το. Στη δικιά μας περίπτωση το codeword είναι 204 ενώ το μήκος του μηνύματος είναι 188. Έτσι αυτός ο Reed-Solomon κωδικοποιητής προσθέτει 16 bit (σύμβολα ισοτιμίας). Έτσι ο μέγιστος αριθμός σφαλμάτων που μπορεί να διορθώσει είναι $16/2=8$ εσφαλμένα bit. Επίσης βλέπουμε ότι πρόκειται για έναν Reed-Solomon που δεν έχει puncture code και δεν είναι shortened.

Αντίστοιχα στη πλευρά του δέκτη έχουμε έναν Reed-Solomon decoder ο οποίος πρέπει και έχει τις ίδιες παραμέτρους προκειμένου να γίνει η αποκωδικοποίηση του μηνύματος.

3.3 Convolutional interleaver και deinterleaver

Στη συνέχεια έχουμε το μπλοκ του convolutional interleaver στον οποίο δηλώνουμε πόσους shift registers θα έχει ο interleaver που θα χρησιμοποιήσουμε.

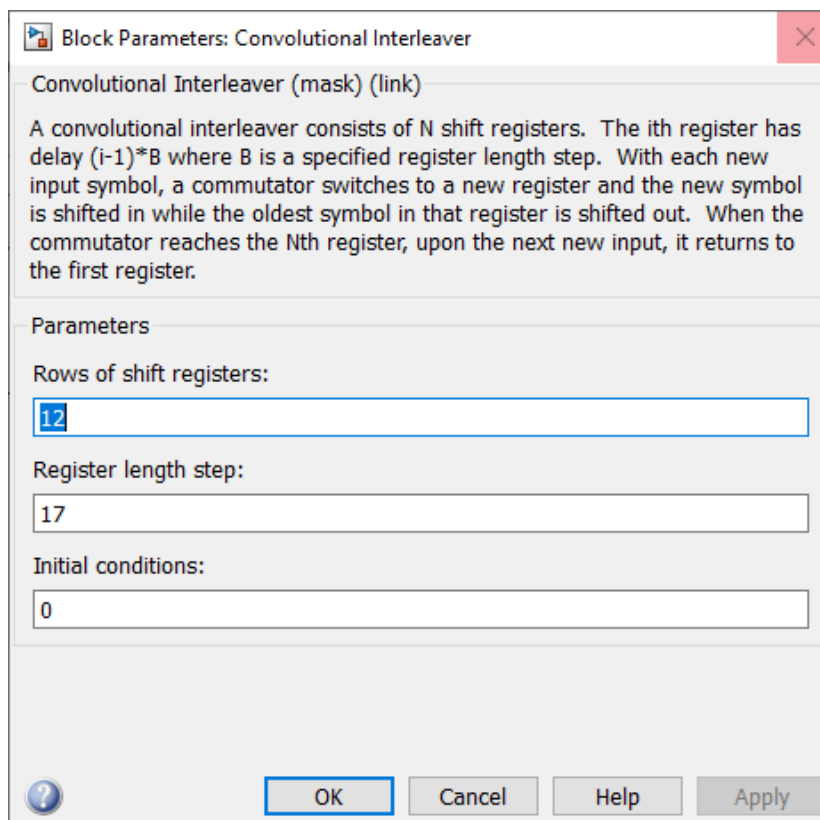


Figure 21 Convolutional interleaver

Στο **Rows of shift registers** επιλέγουμε πόσες σειρές από shift registers θα έχουμε. Στο **Register length step** βάζουμε το βήμα με το οποίο θα αυξάνονται οι shift registers ανά κάθε σειρά. Τέλος στο **Initial conditions** δηλώνουμε ποια θα είναι η αρχική τιμή των shift registers. Στη δικιά μας περίπτωση θα έχουμε έναν convolutional interleaver ο οποίος θα αποτελείται από 12 σειρές shift registers, σε κάθε σειρά θα προσθέτονται 17 shift registers και όλοι οι shift registers θα έχουν ως αρχική τιμή το 0. Έτσι η πρώτη γραμμή θα έχει έναν shift register

μηδενικής μνήμης, δηλαδή ότι μπαίνει βγαίνει αμέσως στην έξοδο, η δεύτερη γραμμή θα έχει 17, η τρίτη 34 κτλ. Μέσα στο μπλοκ του convolutional interleaver μετά τη διαδικασία του interleaving υπάρχει ένα ακόμα μπλοκ με το οποίο γίνεται μετατροπή των ακεραίων αριθμών σε bit. Αυτό συμβαίνει γιατί στο επόμενο μπλοκ θέλουμε το σήμα μας να είναι σε μορφή bit.

Στη πλευρά του δέκτη υπάρχει ακριβώς ο ίδιος interleaver στον οποίο γίνεται η αντίστροφη διαδικασία προκειμένου να επιστρέψει το σήμα μας στην αρχική του κατάσταση.

3.4 Convolutional code

Μετά ακολουθεί το μπλοκ του convolutional code. Είναι το μπλοκ που γίνεται η διαδικασία του **poly2trellis** (polynomial to trellis). Πρόκειται για έναν αριθμό από shift registers που βρίσκονται στην σειρά. Δηλαδή η έξοδος του πρώτου shift register είναι η είσοδος του δεύτερου, η έξοδος του δεύτερου είναι η είσοδος του τρίτου και ούτω καθεξής.

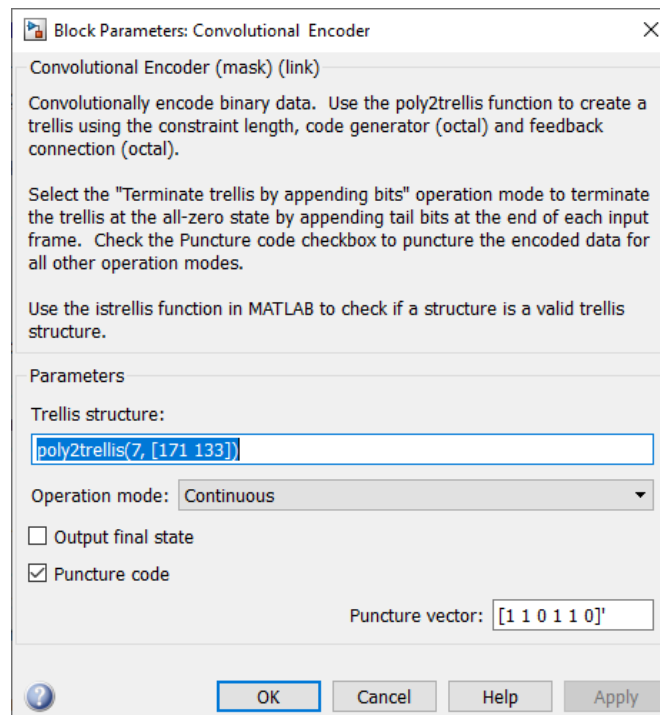


Figure 22 Convolutional code

Στο **Trellis structure** επιλέγουμε το poly2trellis που θέλουμε να χρησιμοποιήσουμε. Το **puncture code** το επιλέγουμε αν θέλουμε ο convolutional code να είναι punctured. Με αυτή την

επιλογή καθορίζουμε και τον ρυθμό μετάδοσης του σήματός μας. Στο puncture vector βάζουμε κάποιες προκαθορισμένες, από το standard του DVB-T πρωτοκόλλου, τιμές. Στη δικιά μας περίπτωση έχουμε έναν convolutional code ο οποίος αποτελείται από 6 shift registers. Τα 171 και 133 αποτελούν τους generator polynomials και είναι στο οκταδικό αν τους μετατρέψουμε στο δυαδικό θα έχουμε τους αριθμούς 01111001 και 01011011 αντίστοιχα. Αυτοί μπαίνουν με την σειρά στις εξόδους των shift registers και στην είσοδο του πρώτου shift register. Όσους generator polynomials έχουμε άλλους τόσους modulo 2 adders έχουμε. Σε όποιες εξόδους έχουμε μηδέν δεν πηγαίνουν προς τον αθροιστή ενώ αντίθετα όπου έχουμε άσους πηγαίνουν. Επίσης ο δικός μας convolutional code είναι punctured επειδή επιθυμούμε να έχουμε έναν συγκεκριμένο ρυθμό. Ο ρυθμός αυτός είναι τα 3/4 δηλαδή στο input μπαίνουν 3 και στο output βγαίνουν 4. Πιο αναλυτικά, μπαίνουν 3 γίνονται επί 2 φορές, αυτό γιατί έχουμε 2 modulo 2 adders, και πάνε στα 6. Από το συγκεκριμένο puncture vector που χρησιμοποιούμε βλέπουμε ότι αφήνουμε να περάσουν μόνο τα 4. Οπότε στο συγκεκριμένο μπαίνουν 3 και βγαίνουν 4.

Στο 4^ο κεφάλαιο μια αλλαγή που θα πραγματοποιήσουμε είναι να αλλάξουμε το puncture vector προκειμένου να επιτύχουμε διαφορετικούς ρυθμούς.

3.5 Inner Interleaver

Το μπλοκ του inner interleaver αποτελείται από άλλα τέσσερα μπλοκ. Τα τέσσερα αυτά μπλοκ είναι δύο **buffers** έναν **block interleaver** και έναν **symbol interleaver**. Στη παρακάτω εικόνα μπορούμε να δούμε τον **inner interleaver** του πρωτοκόλλου DVB-T.

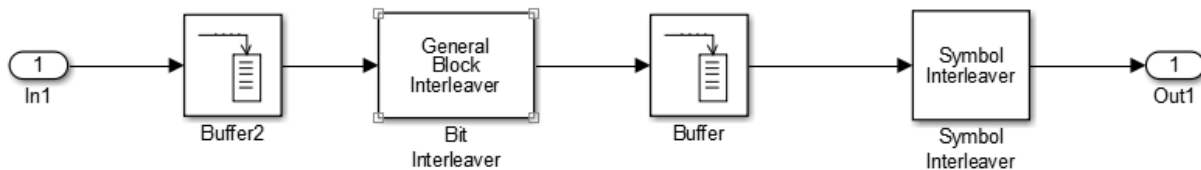


Figure 23 Inner Interleaver

Όπως βλέπουμε ο inner interleaver ξεκινάει με έναν buffer. Σκοπός του συγκεκριμένου buffer είναι να μαζέψει 756 bit προτού τα στείλει. Οπότε με λίγα λόγια αυτό που κάνει είναι να φτιάχνει πακέτα των 756 bit και μετά να τα στέλνει.

Στη συνέχεια ακολουθεί ο block interleaver στον οποίο μπορούμε να δηλώσουμε τον τύπο του interleaver. Στη δικιά μας περίπτωση έχουμε έναν block interleaver προκαθορισμένο από το standard του DVB-T.

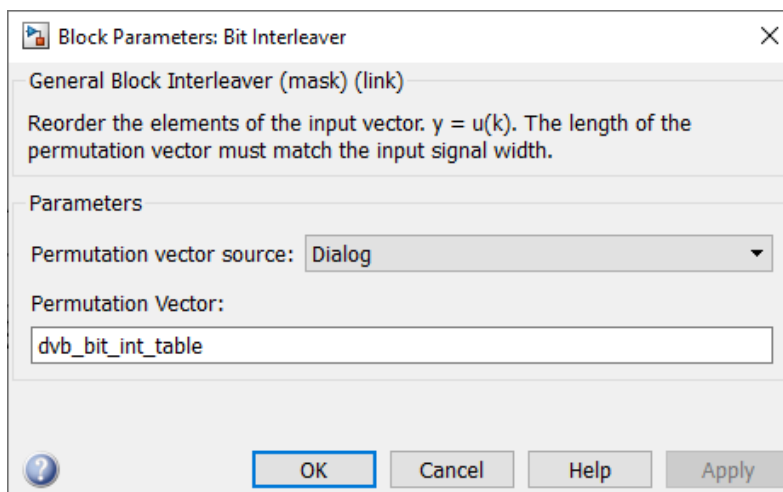


Figure 24 Block interleaver

Όπως βλέπουμε και στην παραπάνω εικόνα, ο συγκεκριμένος interleaver χρησιμοποιεί τον προκαθορισμένο, όπως είπαμε και προηγουμένως, block interleaver τον **dvb_bit_int_table** τον οποίο δηλώνουμε στο πεδίο **Permutation Vector**. Αυτός πρόκειται για έναν πίνακα ο οποίος τοποθετεί σε μια συγκεκριμένη θέση τα bit εισόδου.

Στην συνέχεια ακολουθεί ένας ακόμα **buffer** ο οποίος αυτή τη φορά μαζεύει 9072 bit προτού τα στείλει στον symbol interleaver. Ο symbol interleaver αποτελείται από άλλους δύο block interleaver, ένα switch και ένα control signal που ελέγχει το switch. Ο ένας από τους δύο block interleavers πρόκειται στη πραγματικότητα για έναν deinterleaver. Σκοπός του switch είναι να γίνεται εναλλαγή μεταξύ των δύο interleaver κάθε 9072 bits. Οπότε αυτό που γίνεται σε

αυτόν τον symbol interleaver είναι τα πακέτα των 9072 bit να περνάνε εναλλάξ από τους δύο αυτούς block interleaver. Έτσι το πρώτο πακέτο θα περάσει από τον πάνω interleaver, το δεύτερο από τον κάτω, το τρίτο από τον πάνω κτλ.

Στη παρακάτω εικόνα βλέπουμε τον symbol interleaver που βρίσκεται μέσα στον inner interleaver του DVB-T πρωτοκόλλου.

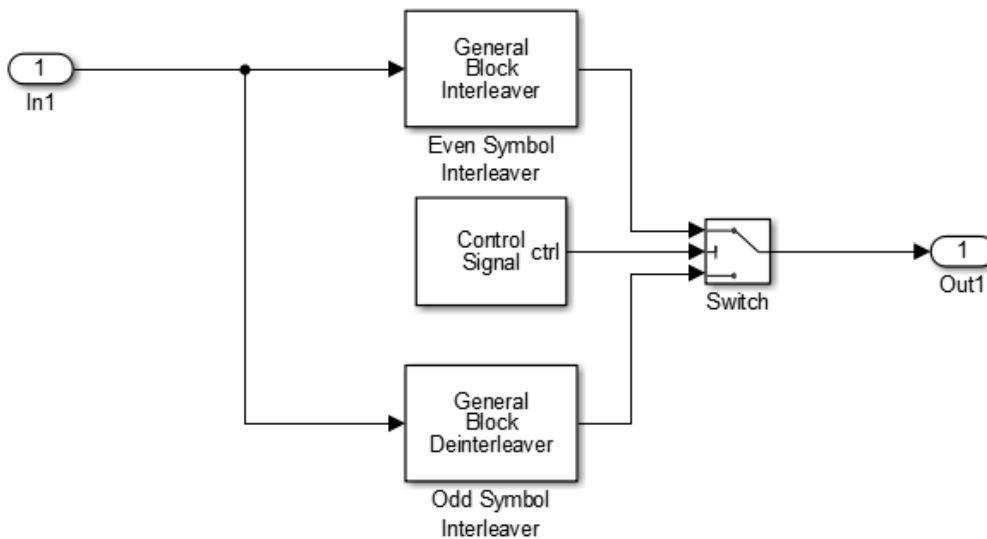


Figure 25 Symbol interleaver

Οι δύο αυτοί interleaver πρόκειται στη πραγματικότητα για τους ίδιους interleaver αλλά ο odd symbol interleaver κάνει την αντίστροφη διαδικασία αφού στη πραγματικότητα πρόκειται για έναν deinterleaver.

3.6 Mapper

Στη συνέχεια ακολουθεί το μπλοκ του **Mapper**. Είναι το μπλοκ που μέσα υπάρχει ο **Modulator** γίνεται η διαδικασία της διαμόρφωσης του σήματος. Ανάλογα με τη διαμόρφωση που θα χρησιμοποιήσουμε θα έχουμε και διαφορετικό constellation mapping. Μέσα στο Mapper υπάρχει επίσης ένα μπλοκ, **Send signal to 64-QAM Demapper**. Αυτό το μπλοκ κάνει αυτό που

λέει το όνομά του, δηλαδή στέλνει το σήμα έτσι όπως είναι, χωρίς τον θόρυβο, κατευθείαν στον **Demapper** στη πλευρά του δέκτη. Αυτό θα το αναλύσουμε όταν θα μιλήσουμε για τον Demapper και επίσης είναι κάτι που χρησιμοποιούμε μόνο στη συγκεκριμένη προσομοίωση και δεν πρόκειται για κάτι που υπάρχει στο πραγματικό πρωτόκολλο.

Στη συνέχεια ακολουθεί η εικόνα με τις παραμέτρους του Modulator.

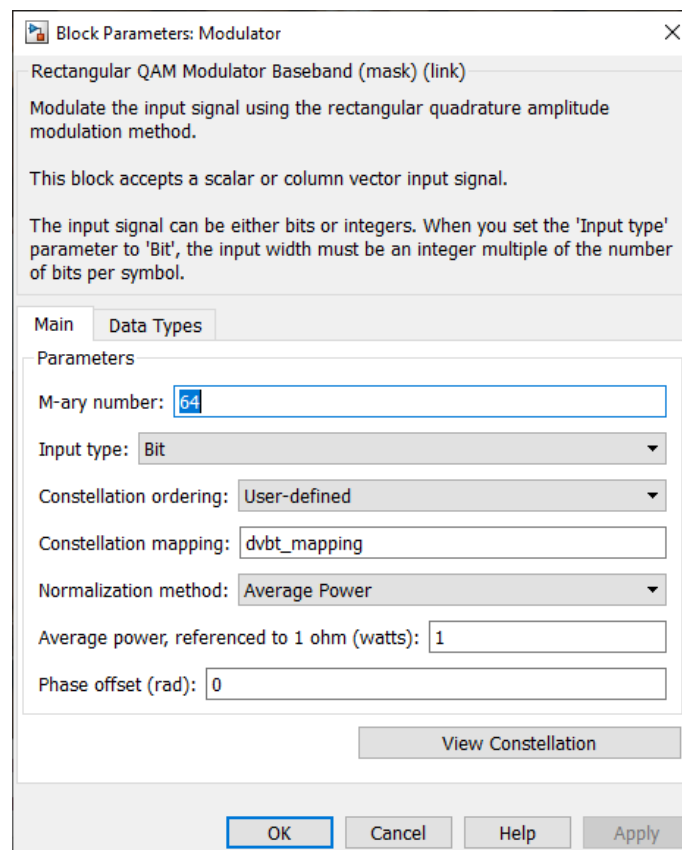


Figure 26 Modulator

Στο πεδίο **M-ary number** δηλώνουμε πόσα σημεία θα έχει το constellation mapping που θα έχει ο διαμορφωτής μας, 64 για το 64-QAM, 16 για το 16-QAM και 4 για το QPSK, που αυτές είναι οι τρεις διαφορετικές διαμορφώσεις που μπορεί να έχει το πρωτόκολλο DVB-T. Στο **Input type** επιλέγουμε τον τύπο των αριθμών που θα μπαίνουν στον modulator και έχει σαν επιλογές bit ή integer. Στο **Constellation ordering** επιλέγουμε user-defined γιατί θέλουμε το constellation mapping να το ορίσουμε εμείς. Στο **Constellation mapping** βάζουμε τη μεταβλητή

του constellation mapper που θα χρησιμοποιήσουμε και είναι σε μορφή κώδικα τον οποίο θα δούμε στο επόμενο κεφάλαιο. Επίσης αυτός ο κώδικας αλλάζει με την αλλαγή της διαμόρφωσης. Το **Normalization method** καθορίζει τη θέση των σημείων του constellation mapper. Σε αυτό το πεδίο έχουμε τρεις διαφορετικές επιλογές. Έχουμε το **Peak power** που με αυτό δηλώνουμε ποια θα είναι η μέγιστη ισχύς. Έχουμε το **Minimum distance between symbols** που με αυτό δηλώνουμε την ελάχιστη απόσταση μεταξύ των σημείων. Τέλος έχουμε το **Average power**, που είναι κι αυτό που θα χρησιμοποιήσουμε, που με αυτό καθορίζουμε ποια θα είναι η μέση ισχύς. Στο **Phase offset (rad)** δηλώνουμε αν θα υπάρχει περιστροφή στα σημεία του constellation mapper.

Στη δική μας περίπτωση έχουμε ένα constellation mapper 64-QAM, 64 σημείων. Είναι user defined γιατί δηλώνουμε εμείς τον τύπο του constellation mapper. Τα σημεία βρίσκονται σε τέτοια θέση έτσι ώστε η μέση ισχύς του σήματος να είναι 1 watt. Τέλος τα σημεία δεν έχουν phase offset δηλαδή δεν υπάρχει περιστροφή.

3.7 OFDM Modulator και Demodulator

Στο τελευταίο μπλοκ του DVB-T πρωτοκόλλου έχουμε τον OFDM modulator. Σε αυτό το μπλοκ γίνεται η μετατροπή του σήματός μας από ψηφιακό σε αναλογικό και αυτό συμβαίνει μετατρέποντας τους μιγαδικούς σε ημίτονα διαφορετικών συχνοτήτων. Επίσης προσθέτει guard bands δεξιά και αριστερά του σήματος για να προστατεύουν το σήμα από πιθανές παρεμβολές από άλλα σήματα.

Ακολουθεί η εικόνα με τις παραμέτρους του OFDM modulator.

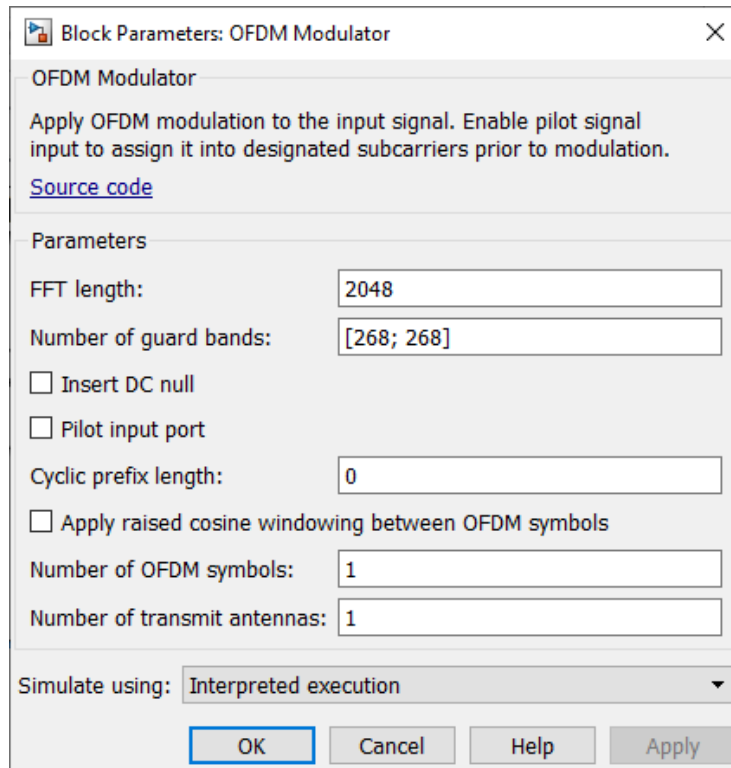


Figure 27 OFDM modulator

Στο πεδίο **FFT length** δηλώνουμε πόσες συχνότητες θα στείλουμε και αναλόγως από το mode που θα χρησιμοποιήσουμε 2k ή 8k η τιμή του αλλάζει σε 2048 ή 8192 αντίστοιχα και στο ακριβώς από κάτω, **Number of guard bands** δηλώνουμε πόσα θα είναι τα guard bands που θα βάλουμε. Στη δικιά μας περίπτωση το μήκος του σήματος μας είναι 1512 συν τα guard bands στην αρχή και στο τέλος του που είναι 268+268 που συνολικά μας κάνει το 2048.

Στη πλευρά του demodulator γίνεται η αντίστροφη διαδικασία. Δηλαδή αφαιρούνται από τη πληροφορία που έλαβε, ο demodulator, τα guard bands προκειμένου να κρατήσουμε μόνο το σήμα που στάλθηκε από τον πομπό.

3.8 AWGN

Για να έχουμε μια σωστή προσομοίωση σήματος θα πρέπει με κάποιο τρόπο να προσθέσουμε τον θόρυβο που θα υπάρχει και στη πραγματικότητα κατά τη μετάδοση του σήματος. Αυτό το επιτυγχάνει το μπλοκ AWGN (additive white Gaussian noise). Λέγεται

additive γιατί προστίθεται θόρυβος, **white** γιατί αναφέρεται στην ιδέα ότι έχει ομοιόμορφη ισχύ ο θόρυβος σε όλο το μήκος των συχνοτήτων και **Gaussian** επειδή έχει μια κανονική κατανομή στο τομέα του χρόνου με το μηδέν ως μια μέση τιμή τομέα χρόνου.

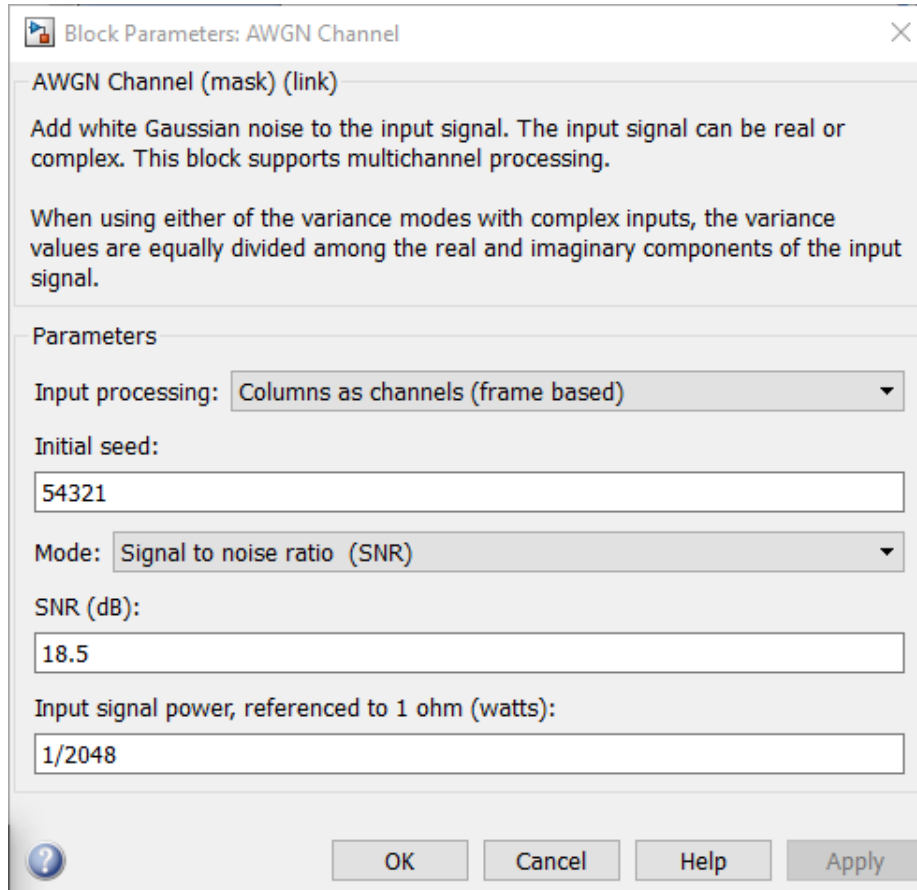


Figure 28 AWGN

Στη παραπάνω εικόνα φαίνεται το μπλοκ του AWGN. Όπως βλέπουμε υπάρχει και σε αυτό το μπλοκ ένα πεδίο για το **Initial seed** πρόκειται για ένα τυχαίο αριθμό που κάνει generate τον θόρυβο. Στο πεδίο του **Mode** καθορίζουμε τη διακύμανση του θορύβου που θα κάνουμε generate. Σε αυτό το πεδίο έχουμε 5 διαφορετικές επιλογές Signal to noise ratio (Eb/No), signal to noise ratio (Es/No), signal to noise ratio (SNR), variance from mask και variance from port. Αν επιλέξουμε το Eb/No δηλώνουμε σε ντεσιμπέλ την αναλογία της ενέργειας των bit πληροφορίας ανά σύμβολο ως προς τη φασματική πυκνότητα της ισχύος του θορύβου. Αν επιλέξουμε το Es/No δηλώνουμε σε ντεσιμπέλ την αναλογία της ενέργειας των συμβόλων πληροφορίας ανά σύμβολο ως προς τη φασματική πυκνότητα ισχύος θορύβου. Αν επιλέξουμε το SNR δηλώνουμε σε ντεσιμπέλ την αναλογία ισχύος του σήματος ως προς την ισχύ του θορύβου.

Τέλος στο πεδίο **Input signal power, reference to 1 ohm (watts)** δηλώνουμε σε watt τη μέση τετραγωνική ισχύ των συμβόλων εισόδου εάν έχουμε επιλέξει παραπάνω στο mode είτε το Eb/No είτε το Es/no ή τη μέση τετραγωνική ισχύ των δειγμάτων εισόδου εάν έχουμε επιλέξει το SNR.

3.9 Spectrum Scope και Scatter Plot

Μέσα στο πρωτόκολλο DVB-T υπάρχουν δυο μπλοκ, το ένα που είναι το Spectrum Scope, μας δείχνει το συχνοτικό περιεχόμενο του σήματος και το άλλο, που είναι το Scatter Plot μας δείχνει το constellation map που χρησιμοποιεί ο Mapper μας.

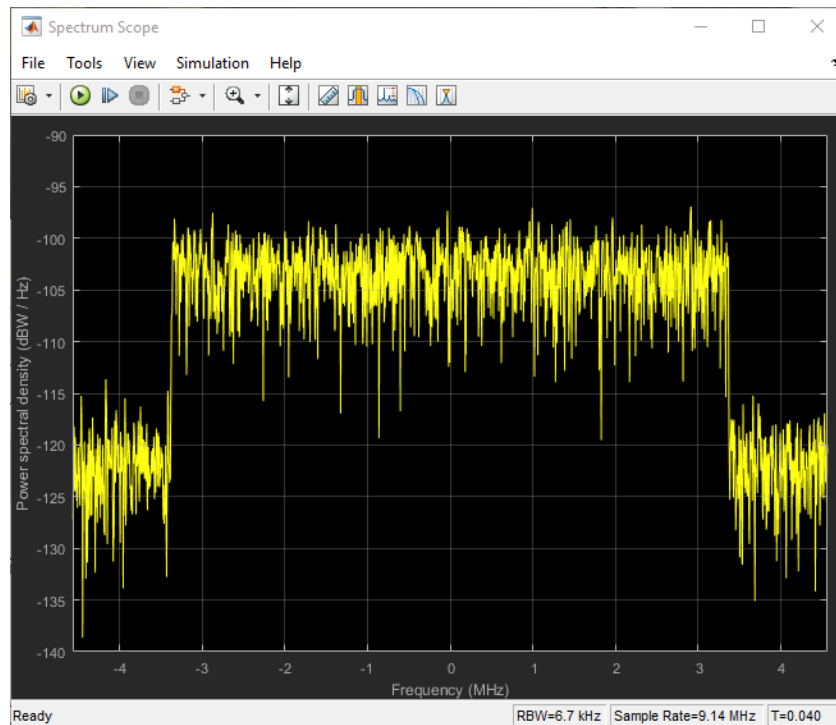


Figure 29 Spectrum Scope

Στο παραπάνω σχήμα βλέπουμε το Spectrum Scope του DVB-T. Πιο συγκεκριμένα βλέπουμε πόση ισχύ έχει το μεταδιδόμενο σήμα σε κάθε συχνότητα. Μπορούμε να διακρίνουμε στο παραπάνω σήμα και τα guard bands τα οποία βρίσκονται στις δυο άκρες στο φάσμα συχνοτήτων.

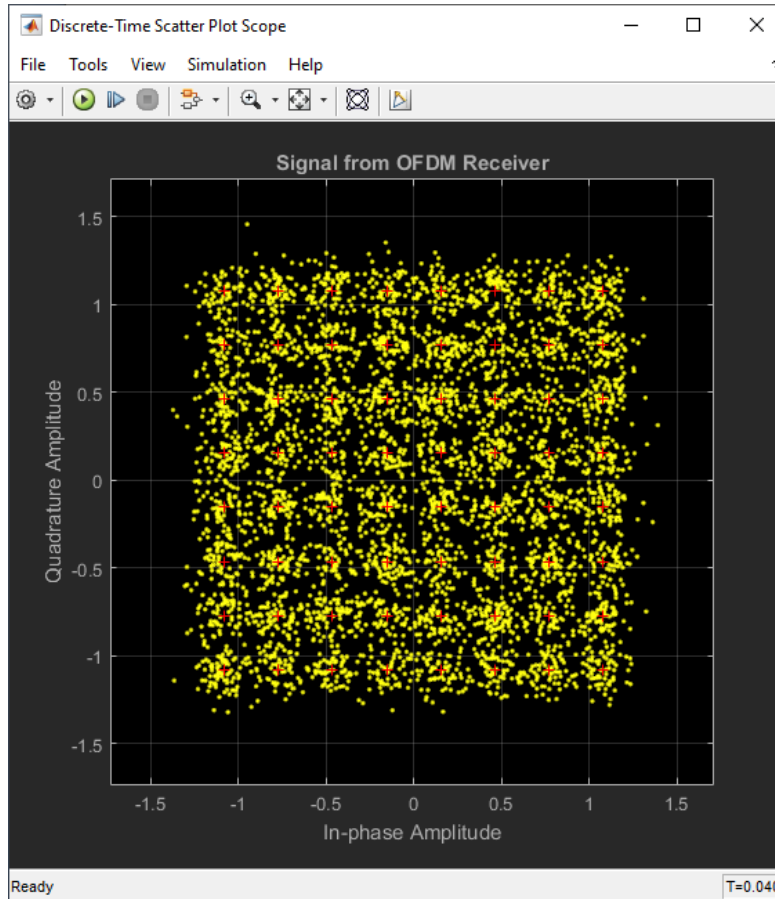


Figure 30 Scatter Plot

Εδώ βλέπουμε το constellation map που χρησιμοποιούμε, που σε αυτή τη περίπτωση πρόκειται για τον 64-QAM, μαζί με το θόρυβο που πρόσθεσε το AWGN.

Στη συνέχεια θα αναφερθούμε σε δυο μπλοκ του δέκτη στα οποία αξίζει να αναφέρουμε κάποια πράγματα μιας και στα άλλα μπλοκ του δέκτη στην ουσία γίνονται οι αντίστροφες διαδικασίες από αυτές που μιλήσαμε παραπάνω (για παράδειγμα στον RS decoder, Convolutional Deinterleaver, Inner Deinterleaver και OFDM Demodulator οι διαδικασίες είναι οι αντίστροφες από τα αντίστοιχα που βρίσκονται στον πομπό). Τα δύο αυτά μπλοκ είναι ο Demapper και ο convolutional decoder (Viterbi decoder).

3.10 Demapper

Όπως είχαμε πει όταν αναλύαμε το μπλοκ του mapper υπήρχε ένα μπλοκ μέσα στον mapper που ονομαζόταν **Send signal to 64-QAM Demapper**. Αυτό το μπλοκ στέλνει το σήμα έτσι όπως είναι, χωρίς τον θόρυβο, κατευθείαν στον **Demapper** στη πλευρά του δέκτη. Τώρα θα δούμε ποιος είναι ο σκοπός αυτού του μπλοκ.

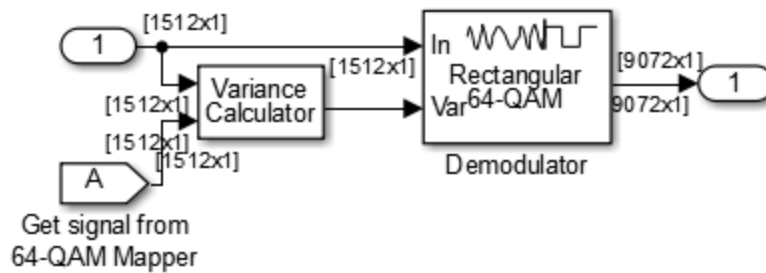


Figure 31 Demapper

Όπως βλέπουμε σαν input έχουμε το σήμα που στείλαμε από τον mapper στο οποίο δεν υπάρχει ο θόρυβος και το κανονικό σήμα με τον θόρυβο. Στη συνέχεια αυτά τα δυο σήματα μπαίνουν στο **Variance Calculator**.

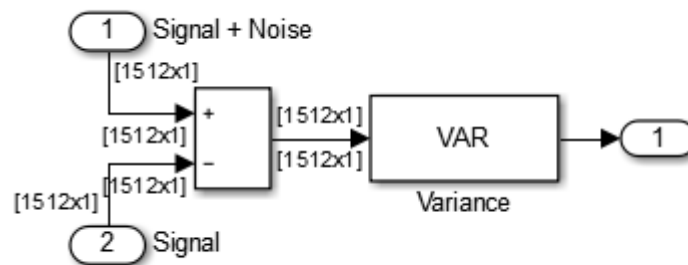


Figure 32 Variance Calculator

Στο μπλοκ του **Variance Calculator** βλέπουμε ότι παίρνουμε το σήμα με τον θόρυβο και αφαιρούμε το σήμα χωρίς το θόρυβο με αποτέλεσμα να πάρουμε μόνο τον θόρυβο. Στη συνέχεια ο θόρυβος περνά από το μπλοκ του VAR (Variance). Το μπλοκ αυτό υπολογίζει τη διακύμανση του θορύβου. Στη συνέχεια, όπως φαίνεται και στο **Figure 31**, το σήμα με τον θόρυβο και η διακύμανση του θορύβου μπαίνουν μέσα στον Demodulator. Αυτό με τον υπολογισμό της διακύμανσης του θορύβου όπως είπαμε και προηγουμένως δεν πρόκειται για διαδικασία που συμβαίνει στη πραγματικότητα, απλά η χρήση τους εδώ είναι απαραίτητη προκειμένου να τρέξει η προσομοίωση.

Στη συνέχεια ακολουθεί ο Demodulator οποίος, όπως είπαμε και προηγουμένως, δέχεται σαν input το σήμα με τον θόρυβο και τη διακύμανση του θορύβου.

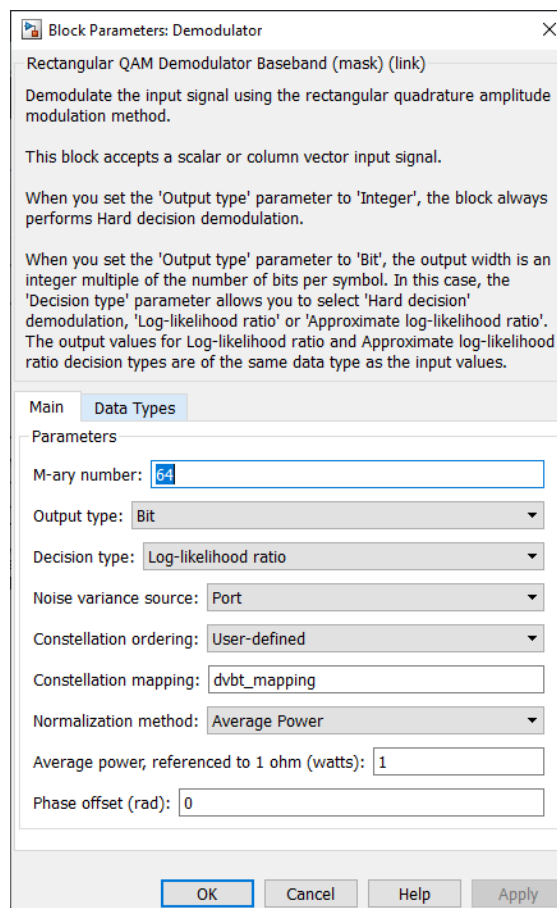


Figure 33 Demodulator

Όπως και στον modulator, στο **M-ary number** δηλώνουμε πόσα σημεία θα έχει το constellation mapping που θα έχει ο από-διαμορφωτής μας και πρέπει να είναι το ίδιο με αυτό που έχουμε βάλει στον διαμορφωτή στη πλευρά του δέκτη. Στο **Output type** δηλώνουμε σε τι μορφή θα είναι το output του από-διαμορφωτή. Στο **Decision type** επιλέγουμε τον αλγόριθμο που αποφασίζει ποιο σύμβολο είναι αυτό που ήρθε. Δηλαδή σε ποιο σημείο στο constellation map ανήκει ένα σύμβολο με βάση την διακύμανση του θορύβου. Αν επιλέξουμε το Log-likelihood ratio, ο αλγόριθμος αυτός λαμβάνει υπόψη του όλα τα κοντινά σημεία του constellation map για να αποφασίσει σε ποιο σημείο ανήκει το κάθε σύμβολο. Ενώ αν επιλέξουμε τον **Approximate log-likelihood ratio** λαμβάνει υπόψη του μόνο τα πολύ κοντινά σημεία. Οι παραπάνω αλγόριθμοι για να τρέξουν χρειάζονται τη διακύμανση του θορύβου. Η διακύμανση του θορύβου δείχνει πόσο ετερογενής είναι ο θόρυβος.

3.11 Viterbi Decoder

Στη συνέχεια έχουμε το μπλοκ του Inner Decoder. Ο λόγος που επέλεξα να αναλύσω το μπλοκ του inner decoder είναι η ύπαρξη ενός delay μπλοκ και μιας μεταβλητής (**traceback depth**) μέσα στον decoder που αξίζει να αναφέρουμε ποια είναι η δουλειά τους. Οι άλλες μεταβλητές μέσα στον Viterbi Decoder στην ουσία είναι οι ίδιες με τις αντίστοιχες μεταβλητές του Inner Encoder που αναλύσαμε προηγουμένως.

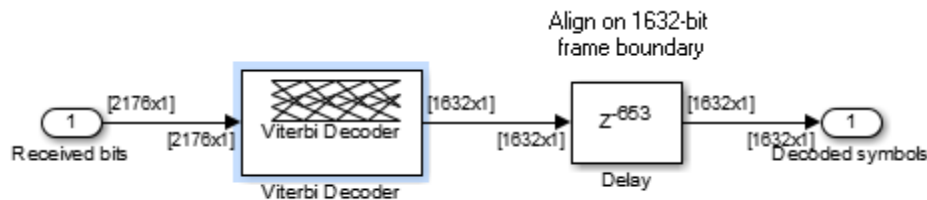


Figure 34 Inner Decoder

Στη παραπάνω εικόνα φαίνεται το μπλοκ του inner decoder. Η μεταβλητή βρίσκεται μέσα στον Viterbi Decoder (inner decoder) και το Delay μπλοκ που βρίσκεται αμέσως μετά. Η

δουλειά του delay μπλοκ είναι να προσθέσει κάποιο delay στο delay του σήματος προκειμένου να είναι πολλαπλάσιο του 1632, όπου 1632 είναι τα frames που στέλνουμε. Δηλαδή κάνει το delay να είναι πολλαπλάσιο των frames.

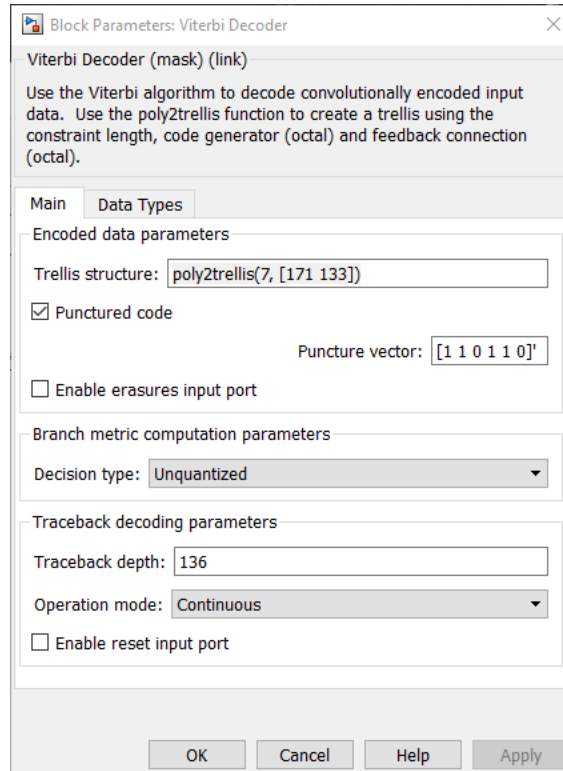


Figure 35 Viterbi Decoder

Όσον αφορά το traceback depth που υπάρχει μέσα στον inner decoder είναι η μεταβλητή με την οποία δηλώνουμε τον αριθμό των bit που θα χρειαστεί ο decoder για να αποκωδικοποιήσει το σήμα. Επίσης αυτή η μεταβλητή καθορίζεται από το στάνταρ του DVB-T και προσθέτει έξτρα delay στο σήμα.

3.12 Error Rate Calculation

Τέλος έχουμε δύο μπλοκ στα οποία γίνεται η μέτρηση για τον αριθμό των errors. Ποιο συγκεκριμένα, τα μπλοκ αυτά, μετράνε τον αριθμό των συνολικών σφαλμάτων, τον συνολικό αριθμό των bit καθώς και το BER (Bit Error Rate). Τα μπλοκ αυτά είναι δύο στον αριθμό. Ο

λόγος είναι για να δούμε πόσα errors έχουμε πριν το σήμα περάσει από τον Reed Solomon και πόσα έχουμε αφού περάσει. Ο τρόπος που λειτουργούν είναι να συγκρίνουν το σήμα αφού περάσει όλες τις διαδικασίες και αφού περάσουν όλες πλην του Reed Solomon.

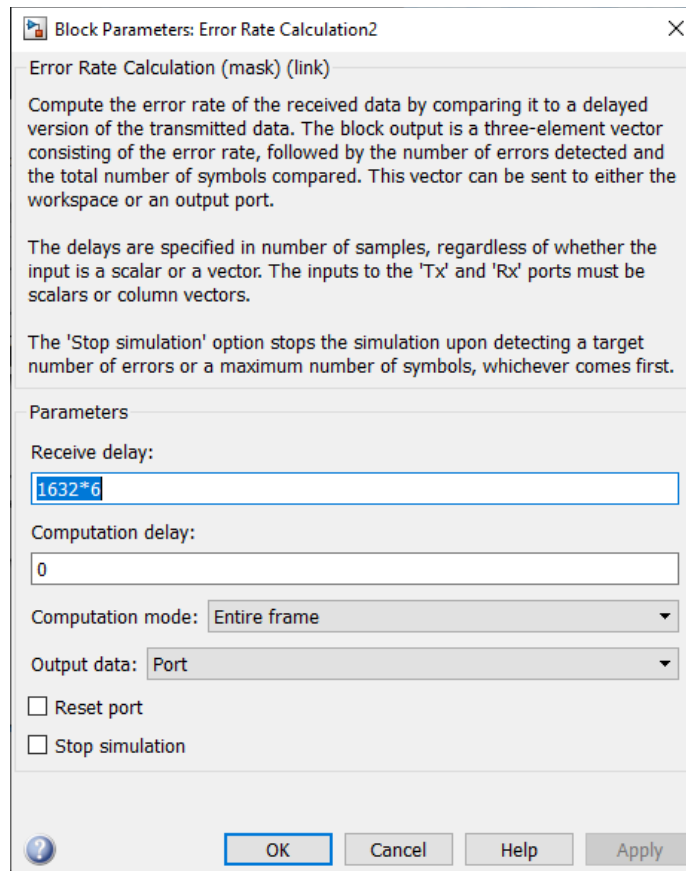


Figure 36 Error Rate Calculation

Όπως βλέπουμε για να δουλέψει ο error rate calculator θα πρέπει να του δηλώσουμε πόσο θα είναι το delay του σήματός μας. Το delay είναι αυτό που προστίθεται στο σήμα μας μέσα στον Viterbi Decoder από τη μεταβλητή traceback depth καθώς και από τον buffer μέσα στον inner deinterleaver.

ΚΕΦΑΛΑΙΟ 4: ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΜΕΣΩ MATLAB

Σε αυτό το κεφάλαιο θα τρέξουμε κάποιες προσομοιώσεις μέσω Matlab. Σε αυτές τις προσομοιώσεις θα πραγματοποιήσουμε κάποιες αλλαγές στις μεταβλητές του κυκλώματος και θα βγάλουμε κάποια συμπεράσματα.

Οι αλλαγές που θα πραγματοποιήσουμε είναι η αλλαγή της διαμόρφωσης ανάμεσα σε QPSK, 16-QAM και 64-QAM, αλλαγή του mode μεταξύ 2k και 8k και τέλος αλλαγή του ρυθμού μεταξύ 1/2, 2/3, 3/4, 5/6 και 7/8. Θα δούμε την επίδραση που θα έχουν αυτές οι αλλαγές στο σήμα μας. Πιο συγκεκριμένα θα δούμε την αλλαγή στο BER για κάποιες τιμές του SNR.

Για να μπορέσω όμως να τρέξω αυτές τις προσομοιώσεις πρέπει να αλλάξω κάποια πράγματα στο κώδικα του κυκλώματος. Έτσι γι αυτό το λόγο έφτιαξα το tablegen για κάθε ρυθμό, διαμόρφωση και mode. Επίσης έκανα κάποιες αλλαγές και στο κύκλωμα. Στη συνέχεια θα δούμε ποιες είναι αυτές οι αλλαγές.

4.1 Κώδικας του DVB-T Πρωτοκόλλου

Προκειμένου να μπορέσει να τρέξει σωστά η προσομοίωση του DVB-T πρωτοκόλλου στο Matlab υπάρχει ένας κώδικας που τρέχει πρώτα. Ο κώδικας αυτός, ο οποίος είναι ο default, δηλαδή 2k mode, 64-QAM και ρυθμό 3/4, είναι ο παρακάτω:

```
% COMMDVBT_TABLEGEN Generate interleaver and modulator lookup tables for
% COMMDVBT example

% Copyright 1996-2012 The MathWorks, Inc.
%
%
% Section numbers refer to the ETSI DVB-T Standard,
% EN 300 744 V1.1.2 (1997-08)
%
```

```

%
% Set the sample time Ts:
% 0.000224 - per Section 4.4 parameters for 2K mode
% 2176 - message size into the 'DVB-T Inner Interleaver' subsystem
% 9072 - message size into the 'DVB-T Inner Deinterleaver' subsystem
% 188 - message size into 'RS Encoder' block
%
Ts = 0.000224*2176/9072/188;

%
% Compute lookup table for inner bit-wise interleaver
% as per section 4.3.4.1
%
% Initial table setup
dvb_bit_int_table = 1:756;
dvb_bit_int_table = reshape(dvb_bit_int_table, 6, 126);

% 64-QAM non-hierarchical demultiplexing
dvb_bit_int_table = dvb_bit_int_table([1 4 2 5 3 6],:);

% Six individual interleavers
dvb_bit_int_table(1,:) = dvb_bit_int_table(1,mod([0:125] + 0,126) + 1);
dvb_bit_int_table(2,:) = dvb_bit_int_table(2,mod([0:125] + 63,126) + 1);
dvb_bit_int_table(3,:) = dvb_bit_int_table(3,mod([0:125] +105,126) + 1);
dvb_bit_int_table(4,:) = dvb_bit_int_table(4,mod([0:125] + 42,126) + 1);
dvb_bit_int_table(5,:) = dvb_bit_int_table(5,mod([0:125] + 21,126) + 1);
dvb_bit_int_table(6,:) = dvb_bit_int_table(6,mod([0:125] + 84,126) + 1);

% Flatten matrix into lookup table
dvb_bit_int_table = dvb_bit_int_table(:);

%
% Compute lookup table for inner symbol interleaver
% as per section 4.3.4.2
%
% Following Figure 8a and accompanying equations
q = 0;
h = zeros(1,2048);
R = zeros(1,10);
for indx = 0:2047,
    if indx == 2,
        R(1) = 1;
    end
    Rsum = R(3)*2^9+R(6)*2^8+R(9)*2^7+R(4)*2^6+...
        R(8)*2^5+R(1)*2^4+R(2)*2^3+R(5)*2^2+R(7)*2^1+R(10)*2^0;

```



```

-3 + 7i
 3 - 7i
-3 - 7i
 5 + 1i
-5 + 1i
 5 - 1i
-5 - 1i
 3 + 1i
-3 + 1i
 3 - 1i
-3 - 1i
 7 + 5i
-7 + 5i
 7 - 5i
-7 - 5i
 1 + 5i
-1 + 5i
 1 - 5i
-1 - 5i
 7 + 3i
-7 + 3i
 7 - 3i
-7 - 3i
 1 + 3i
-1 + 3i
 1 - 3i
-1 - 3i
 5 + 5i
-5 + 5i
 5 - 5i
-5 - 5i
 3 + 5i
-3 + 5i
 3 - 5i
-3 - 5i
 5 + 3i
-5 + 3i
 5 - 3i
-5 - 3i
 3 + 3i
-3 + 3i
 3 - 3i
-3 - 3i ];

```

```

% Normalization factor from Section 4.4

```

```

dvbt_qam = dvbt_qam/sqrt(42);

```

```

% 64-QAM Constellation mapping from Figure 9a. Note that the bit words are
% flipped. This mapping is used by alternative version of the example.

```

```

dvbt_const_mapping = [ ...
    1  17  21  5  4  20  16  0;...
    33 49  53 37 36 52 48 32;...
    41 57  61 45 44 60 56 40;...
    9  25  29 13 12 28 24  8;...
    11 27  31 15 14 30 26 10;...
    43 59  63 47 46 62 58 42;...
    35 51  55 39 38 54 50 34;...
    3  19  23  7  6  22  18  2 ...
    ];

dvbt_mapping = dvbt_const_mapping(:);

```

Αυτός είναι ο κώδικας που τρέχει και προγραμματίζει τους δύο εξωτερικούς interleavers (interleaver πομπού και δέκτη) και τους δύο encoders (encoder πομπού και δέκτη). Στη συνέχεια θα αναλύσουμε τι κάνει το κάθε κομμάτι του κώδικα. Όλες οι σειρές που ξεκινάνε με το σύμβολο % πρόκειται για σχόλια και δεν κάνουν κάτι στο κώδικα μας.

Στη αρχή δηλώνουμε τη τιμή του T_s . Όπως βλέπουμε και στα σχόλια, και σύμφωνα με το standard του DVB-T, για να το υπολογίσουμε χρειαζόμαστε το μήκος του μηνύματος που μπαίνει στον inner interleaver, το μήκος του μηνύματος που μπαίνει στο inner deinterleaver, το μήκος του μηνύματος που μπαίνει στον RS encoder και αυτόν τον συγκεκριμένο αριθμό που ορίζει το standard του DVB-T για το 2k mode. Άμα θέλουμε να αλλάξουμε το mode από 2k σε 8k τότε ο αριθμός αυτός θα αλλάξει.

Στη συνέχεια ακολουθεί ο προγραμματισμός του πρώτου block interleaver. Αρχικά φτιάχνεται ένας πίνακας με μία γραμμή και 756 στήλες. Ο αριθμός των στηλών αλλάζει ανάλογα με τη διαμόρφωση που χρησιμοποιούμε. Έτσι για 16-QAM έχουμε 504 στήλες και για QPSK έχουμε 252 στήλες. Στη συνέχεια αυτός ο πίνακας μετατρέπεται σε ένα πίνακα με 6 γραμμές και 126 στήλες. Αυτό συμβαίνει γιατί έχουμε τη διαμόρφωση 64-QAM όπου έχουμε 6 interleavers. Αντίστοιχα στην 16-QAM έχουμε 4 και στην QPSK έχουμε 2. Στη συνέχεια ανακατεύει τις γραμμές αυτού του πίνακα. Δηλαδή στη πρώτη γραμμή μένει η πρώτη γραμμή, στη δεύτερη γραμμή πάει η τέταρτη, στη τρίτη γραμμή πάει η δεύτερη κτλ. Μετά πηγαίνει σε καθεμία γραμμή και μεταφέρει τα σύμβολα κάποιες θέσεις αριστερά. Όπως βλέπουμε στη πρώτη γραμμή δεν κάνει κάποια αλλαγή, στη δεύτερη τα μεταφέρει 63 θέσεις αριστερά, στη τρίτη γραμμή τα

μεταφέρει 105 θέσεις αριστερά κτλ. Αφού ολοκληρωθεί και αυτή η διεργασία μετατρέπει ξανά τον πίνακα σε πίνακα μία σειράς και 756 στηλών.

Στο επόμενο κομμάτι γίνεται ο προγραμματισμός των άλλων δύο interleaver (symbol interleaver). Επίσης αυτό το κομμάτι αλλάζει αν αντί για 2k mode έχουμε 8k mode. Σε αυτό το κομμάτι φτιάχνει τον πίνακα έτσι ώστε να μετατρέπει τα νούμερα σε 6-bit. Αν έχουμε 16-QAM έχουμε 4-bit νούμερα και στο QPSK έχουμε 2-bit νούμερα.

Το επόμενο κομμάτι πρόκειται για ένα διάνυσμα 64 μιγαδικών αριθμών το οποίο, επειδή εμείς χρησιμοποιούμε το alternative version του DVB-T, χρησιμοποιείται μόνο για το scatter plot. Στο 16-QAM θα έχουμε 16 μιγαδικούς και στο QPSK θα έχουμε 4 μιγαδικούς. Παρακάτω φαίνεται πως αλλάζει ο κώδικας ανάλογα με τη διαμόρφωση.

Έτσι στο 16-QAM αυτό το κομμάτι γίνεται:

```
dvbt_qam = [ 3 + 3i
            -3 + 3i
              3 - 3i
            -3 - 3i
              1 + 3i
            -1 + 3i
              1 - 3i
            -1 - 3i
              3 + 1i
            -3 + 1i
              3 - 1i
            -3 - 1i
              1 + 1i
            -1 + 1i
              1 - 1i
            -1 - 1i ] ;
```

Και στο QPSK γίνεται:

```
dvbt_qam = [ 1 + 1i
            -1 + 1i
              1 - 1i
            -1 - 1i ] ;
```

Τέλος στο τελευταίο κομμάτι γίνεται το mapping δηλαδή μας δείχνει σε ποια θέση στο constellation map θα μπουν οι αριθμοί από 0-63 όπου πρόκειται για 6-bit αριθμούς. Κάθε μία

από αυτές τις θέσεις αντιστοιχεί σε έναν από τους 64 μιγαδικούς. Έτσι για παράδειγμα ο αριθμός 4 αντιστοιχεί στον μιγαδικό $1 + 7i$ και έτσι με αυτό το τρόπο όλοι οι αριθμοί μεταξύ 0-63 αντιστοιχούνται σε έναν από τους 64 μιγαδικούς. Όπως και στα άλλα κομμάτια του κώδικα, αυτό το κομμάτι αλλάζει ανάλογα με τη διαμόρφωση που χρησιμοποιούμε. Έτσι για 16-QAM θα έχουμε τους αριθμούς από 0-15 όπου πρόκειται για 4-bit αριθμούς και στο QPSK θα έχουμε αριθμούς από 0-3 όπου πρόκειται για 2-bit αριθμούς. Η αλλαγή στο κώδικα φαίνεται παρακάτω.

Στο 16-QAM αυτό το κομμάτι θα γίνει:

```
dvbt_const_mapping = [ ...  
    1  5  4  0;...  
    9 13 12  8;...  
   11 15 14 10;...  
    3  7  6  2 ...  
                                ];
```

Και στο QPSK αυτό το κομμάτι θα γίνει:

```
dvbt_const_mapping = [ ...  
    1  0;...  
    3  2          ];
```

Κάθε μιγαδικός μετατρέπεται στη συνέχεια σε ημίτονα με διαφορετικό πλάτος και ισχύ και περνάνε στο φάσμα συχνοτήτων.

4.2 Αλλαγές στο κύκλωμα του DVB-T στο Matlab

Όταν δοκίμασα να τρέξω τη προσομοίωση του DVB-T στο Matlab με διαφορετικά code rate, σε κάποιες περιπτώσεις αντιμετώπισα κάποια προβλήματα. Έτσι αναγκάστηκα να κάνω κάποιες αλλαγές στο κύκλωμα.

Μία αλλαγή ήταν να προσθέσω έναν **Buffer** πριν τον convolutional code (poly2trellis). Αυτή την αλλαγή τη κάνουμε στις περιπτώσεις που το code rate που επιλέγουμε είναι είτε 5/6 είτε 7/8. Η χρήση του buffer πριν το poly2trellis είναι γιατί θέλουμε τα bits που θα εισέλθουν σε αυτόν να είναι πολλαπλάσια του 5 και του 7 αντίστοιχα. Έτσι στη περίπτωση που το code rate είναι 5/6 αυτός ο Buffer θα μαζέψει 1632*5 bits πριν τα στείλει ενώ όταν το code rate είναι 7/8 ο Buffer θα μαζέψει 1632*7 πριν τα στείλει. Και στις δύο περιπτώσεις στη πλευρά του δέκτη βάζουμε άλλον ένα Buffer που μαζεύει 1632 πριν τα στείλει δηλαδή το γυρνάει στο αρχικό του μέγεθος. Αυτοί οι δύο Buffer προσθέτουν σε κάθε περίπτωση έξτρα delay στο εξωτερικό Error Rate Calculation.

Ένα άλλο πρόβλημα που αντιμετώπισα ήταν με το Delay του σήματος. Στο μπλοκ του Viterbi decoder υπάρχει ένα μπλοκ που προσθέτει το απαιτούμενο delay στο σήμα. Ουσιαστικά αυτό που κάνει είναι να κάνει το delay να είναι πολλαπλάσιο των frames, δηλαδή της μορφής $k*1632$ (όπου k είναι ένας ακέραιος). Δηλαδή προσθέτει ένα delay στο delay του σήματος προκειμένου να είναι πολλαπλάσιο του 1632.

Στη παρακάτω εικόνα μπορούμε να δούμε την αλλαγή που έκανα:

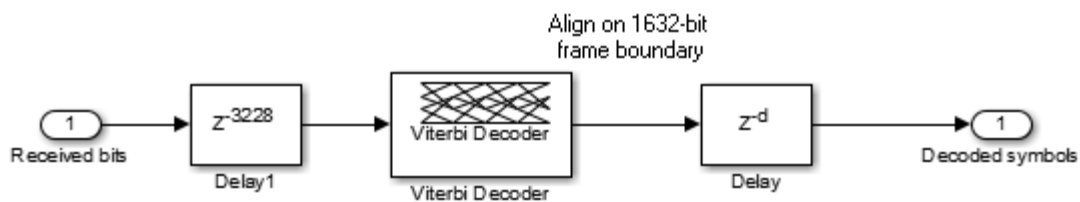


Figure 37 Inner Decoder

Όπως βλέπουμε στη παραπάνω εικόνα έχω προσθέσει ένα delay μπλοκ πριν τον Viterbi Decoder. Όταν, σε κάποιες περιπτώσεις, ο ρυθμός αλλάζει σε 5/6 και 7/8 το delay που πρέπει να προστεθεί δεν είναι ακέραιος αριθμός. Για αυτό το λόγο μεταφέρουμε το delay πριν τον Viterbi

Decoder ώστε να προκύπτει ακέραιος αριθμός. Μετά από τον Viterbi Decoder υπάρχει ακόμα ένα μπλοκ delay. Επειδή ο Viterbi Decoder προσθέτει καινούργιο delay στο σήμα ίσο με traceback depth, προσθέτουμε το άλλο delay μπλοκ αμέσως μετά ώστε να ευθυγραμμιστούν ξανά σε frame. Για το τι τιμές θα βάλουμε στο μπλοκ του delay πριν τον Viterbi Decoder έφτιαξα ένα script που θα το δούμε στη συνέχεια. Στο μπλοκ του delay μετά τον Viterbi βάζουμε πάντα $1632 - \text{traceback_depth}$ αφού όπως είπαμε και παραπάνω θέλουμε να ευθυγραμμιστούν ξανά σε frame (frame=1632).

Στη συνέχεια θα δούμε τα δύο script που έφτιαξα προκειμένου να κάνουν τη διαδικασία πιο γρήγορη. Το ένα script υπολογίζει τις τιμές του delay σε κάθε περίπτωση και το άλλο τρέχει τη προσομοίωση για διάφορες τιμές του SNR και μας βγάζει ένα γράφημα για τις τιμές του BER (Bit Error Rate) ως προς το SNR και για τον εσωτερικό και τον εξωτερικό Error Rate calculator.

4.3 Script για τα delay και για τις μετρήσεις

Όπως είπαμε και παραπάνω προκειμένου να κάνω τη διαδικασία πιο γρήγορη έφτιαξα δύο script. Το ένα υπολογίζει τα delay που θα βάλουμε στον Inner Decoder και στους δύο Error Rate Calculator και το άλλο τρέχει τη προσομοίωση για κάποιες τιμές SNR που εμείς επιλέξαμε και μας δίνει πίσω στο BER (Bit Error Rate).

Στη συνέχεια ακολουθεί το script για τον υπολογισμό των delays.

```
function [ align, inner, outer ] = commdvbt_delays( rate, mode, modulation )
% COMMDVBT_DELAYS Calculate delays for the commdvbt model
%
% PARAMETERS:
% rate: inner coding code rates
% possible values: 1/2, 2/3, 3/4, 5/6, 7/8
%
% mode: '2K' or '8K'
%
% modulation: '64-QAM', '16-QAM' or 'QPSK'
%
% OUTPUTS:
% align: delay to align on frame boundary
%
```

```

% inner: receive delay for inner error rate calculation
%
% outer: receive delay for outer error rate calculation

if strcmp(modulation, '64-QAM')
    interleavers = 6;
elseif strcmp(modulation, '16-QAM')
    interleavers = 4;
elseif strcmp(modulation, 'QPSK')
    interleavers = 2;
else
    error('unknown modulation type')
end

if strcmp(mode, '2K')
    groups = 12;
elseif strcmp(mode, '8K')
    groups = 48;
else
    error('unknown mode')
end

if rate == 5/6
    frame = 5 * 1632;
elseif rate == 7/8
    frame = 7 * 1632;
else
    frame = 1632;
end

buffer1 = 126 * interleavers;
buffer2 = buffer1 * groups;
buffer3 = frame / rate;
total = buffer1 + buffer2 + buffer3;
align = buffer3 - rem(total, buffer3);
inner = (total + align) * rate / 1632 + 1 ;

outer = 11 + inner;
if rate == 5/6
    outer = outer + 5;
elseif rate == 7/8
    outer = outer + 7;
end

end

```

Αυτό που κάνει αυτό το script είναι να παίρνει από τον χρήστη το code rate, το mode και τη διαμόρφωση που θέλουμε να χρησιμοποιήσουμε κάθε φορά. Για τον υπολογισμό των delays πήρα βοήθεια από το help που υπάρχει στο Matlab για τη προσομοίωση του DVB-T. Αυτά που χρειάζονται είναι αυτά που ζητάει από τον χρήστη το script. Έτσι στην αρχή ανάλογα με τη διαμόρφωση που χρησιμοποιούμε, 64-QAM, 16-QAM και QPSK έχουμε 6, 4 και 2 interleavers

αντίστοιχα. Στη συνέχεια ανάλογα με το mode που χρησιμοποιούμε, 2k και 8k έχουμε 12 και 48 groups αντίστοιχα. Και τέλος όπως είπαμε και προηγουμένως όταν έχουμε code rate 5/6 και 7/8 αλλάζουμε το frame σε 5*1632 και 7*1632 αντίστοιχα. Τέλος αφού πάρει ως δεδομένα τις πληροφορίες αυτές ακολουθούν κάποιες πράξεις που υπολογίζουν τα τρία διαφορετικά μπλοκ delays που έχουμε στο κύκλωμά μας. Αυτές τις πράξεις, όπως ανέφερα και παραπάνω τις πήρα από το help που υπάρχει στο Matlab.

Στη συνέχεια ακολουθεί το script που έφτιαξα προκειμένου να βγάλω τα αποτελέσματα από τις μετρήσεις τις οποίες θα δούμε αργότερα.

```
clear ber_inner errors_inner total_inner
clear ber_outer errors_outer total_outer
clear corrected corrected_outer
snrs = 17 : 0.2 : 20.5;
for i = 1 : length(snrs)
    snr = snrs(i);
    % sim('Copy_of_commdvbt_2K_64QAM_7_8', 'StopTime', '1e-2')
    sim('Copy_of_commdvbt_2K_64QAM_7_8')
    ber_inner(i) = e1(1);
    errors_inner(i) = e1(2);
    total_inner(i) = e1(3);

    ber_outer(i) = e2(1);
    errors_outer(i) = e2(2);
    total_outer(i) = e2(3);
    corrected_inner(i) = errors_inner(i) - errors_outer(i);
end
plot(snrs, ber_inner, 'rx--')
xlabel('SNR (dB)')
ylabel('BER')
title('BER vs SNR for commdvbt mode 2K, 64-QAM, coderate 7/8')
hold
plot(snrs, ber_outer, 'bx--')
legend('Inner', 'Outer')
```

Στην αρχή του κώδικα κάνουμε clear τις μεταβλητές προκειμένου να σβήσουμε όποιες πιθανές μεταβλητές έχουν αποθηκευτεί σε προηγούμενες μετρήσεις. Στη συνέχεια βάζουμε ποιες τιμές θέλουμε να πάρει το SNR και με ποιο βήμα να αυξάνεται. Σε αυτή τη περίπτωση το SNR θα πάρει τις τιμές από 17 μέχρι 20.5 με βήμα 0.2. Μετά έχουμε φτιάξει μία for όπου δηλώνουμε ότι θα τρέχει όσο είναι το length του SNR. Τη γραμμή που έχω βάλει σε σχόλιο είναι σε περιπτώσεις που το script ήθελε να αλλάζουμε και το StopTime, δηλαδή για το πόσο χρονικό

διάστημα θα τρέχει η προσομοίωσή μας, προκειμένου να τρέξει σωστά. Με την εντολή `sim` τρέχει η προσομοίωση που του βάζουμε να τρέξει. Παρακάτω οι μεταβλητές `e1` και `e2` είναι οι μεταβλητές του inner και του outer error calculation αντίστοιχα. Έχουν τρεις διαφορετικές τιμές γιατί έχουμε τρία διαφορετικά αποτελέσματα σε κάθε error calculation τα οποία είναι τα BER, Total Errors και Total Bits. Μετά την `for` ακολουθεί η εντολή `plot` που φτιάχνει ένα διάγραμμα στους `x,y` άξονες όπου στον άξονα `x` θα μπει το SNR και στον άξονα `y` θα μπει το BER. Το `'rx—'` δηλώνει ότι το διάγραμμα θα έχει γραμμές με κόκκινο χρώμα και τα σημεία θα φαίνονται με ένα `x`. Στα `xlabel` και `ylabel` δηλώνουμε τον τίτλο των αξόνων `x` και `y` και στο `title` βάζουμε το τίτλο του διαγράμματος. Τέλος με την εντολή `hold` του λέμε να κρατήσει το plot που μόλις δημιουργήσαμε.

4.4 Μετρήσεις και συμπεράσματα

Στη συνέχεια θα δούμε κάποιες μετρήσεις που έκανα και τα συμπεράσματα που έβγαλα για κάθε μία.

Η πρώτη μέτρηση που έκανα ήταν με τις τρεις διαφορετικές διαμορφώσεις και κοινά τα code rate (3/4) και mode (2k). Έτσι για τη πρώτη μέτρηση ακολουθούν οι εικόνες από τα scatter plot και spectrum scope:

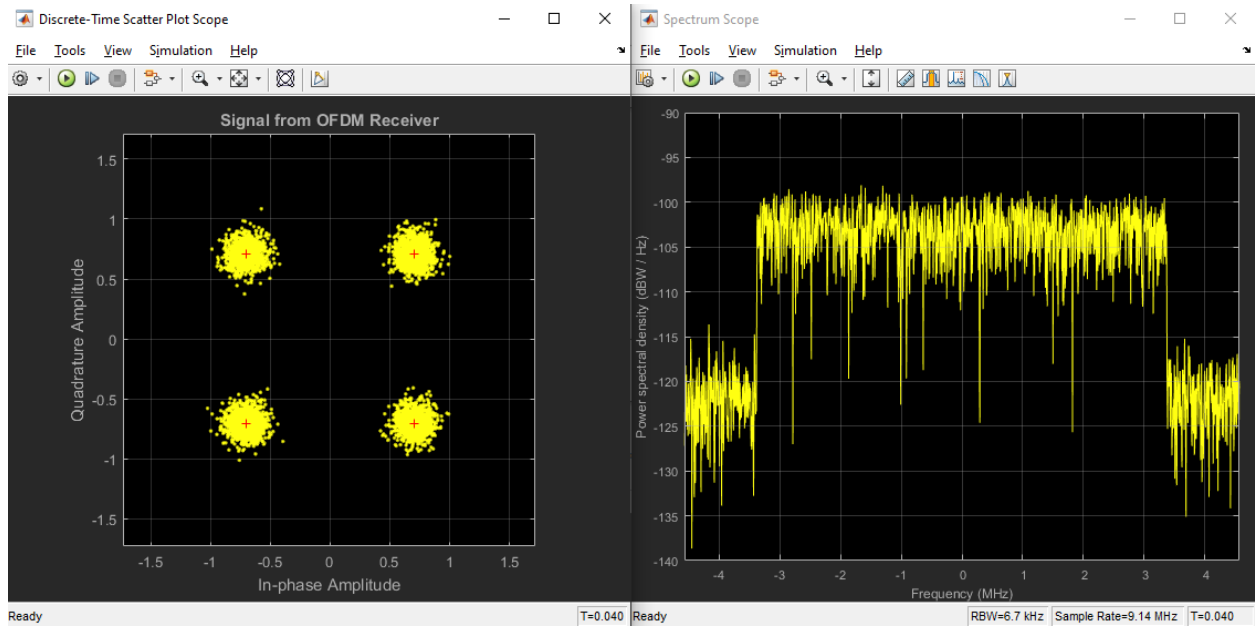


Figure 38 Scatter Plot & Spectrum Scope for QPSK

Μπορούμε να διακρίνουμε το QPSK αφού έχουμε 4 διαφορετικά σημεία στο διάγραμμα αστερισμού και το σήμα μας πάνω στο φάσμα συχνοτήτων.

Ακολουθεί το αποτέλεσμα από το script:

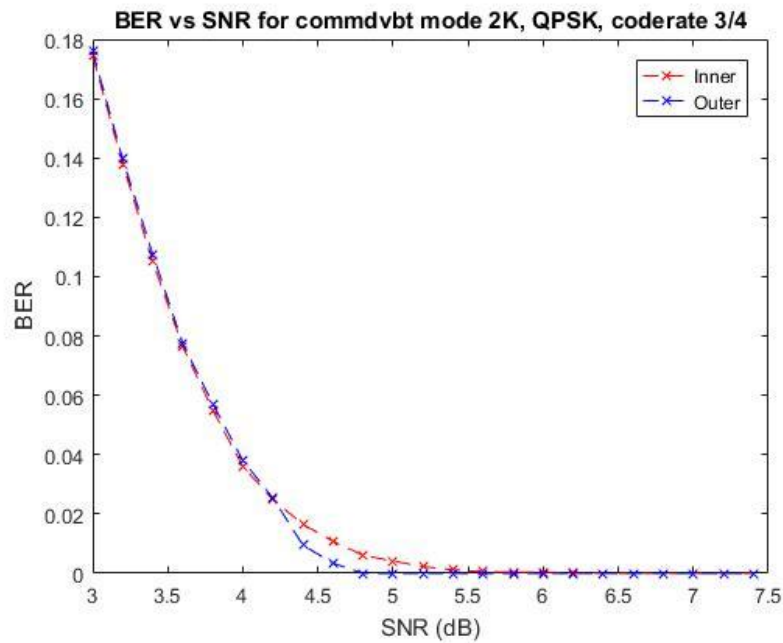


Figure 39 Διάγραμμα BER vs SNR for QPSK

Ακολουθούν τα αποτελέσματα για 16-QAM:

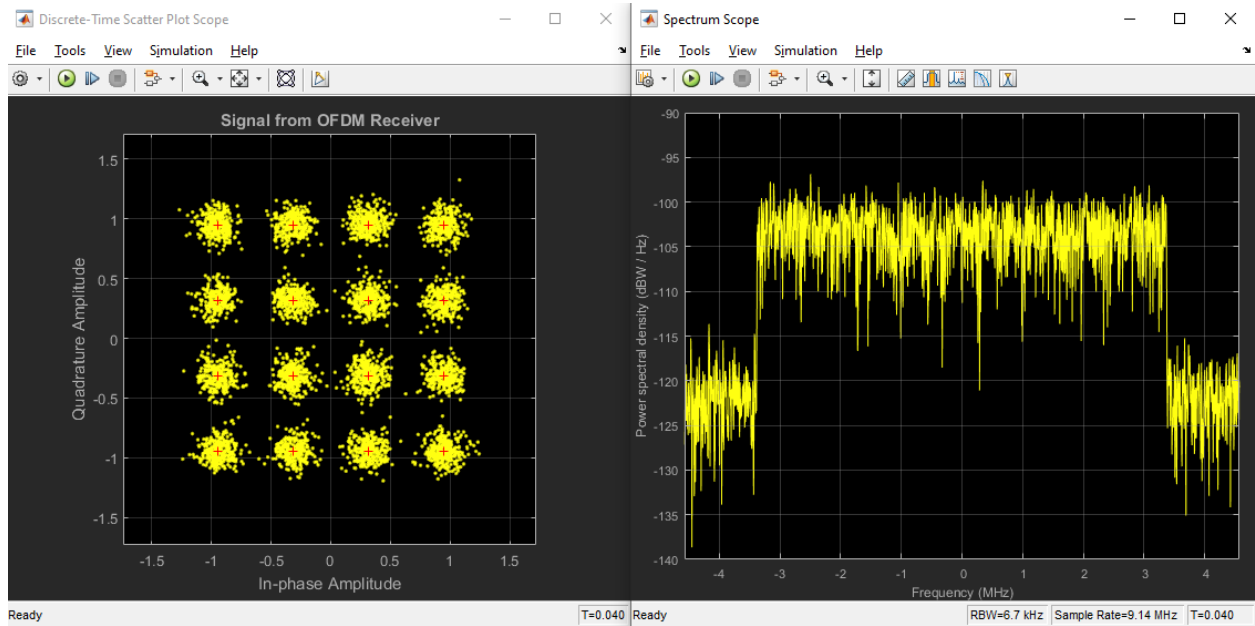


Figure 40 Scatter Plot & Spectrum Scope for 16-QAM

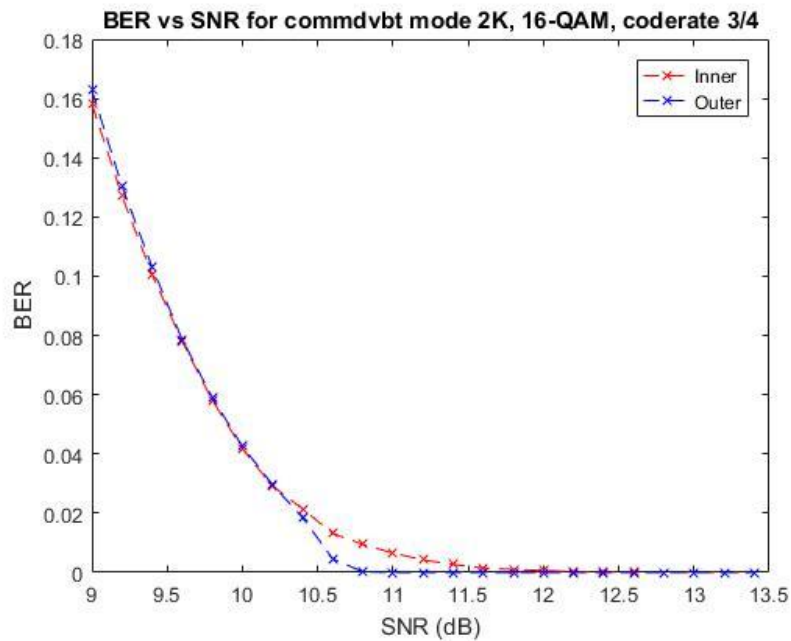


Figure 41 Διάγραμμα BER vs SNR for 16-QAM

Τέλος έχουμε και τα αποτελέσματα για την 64-QAM:

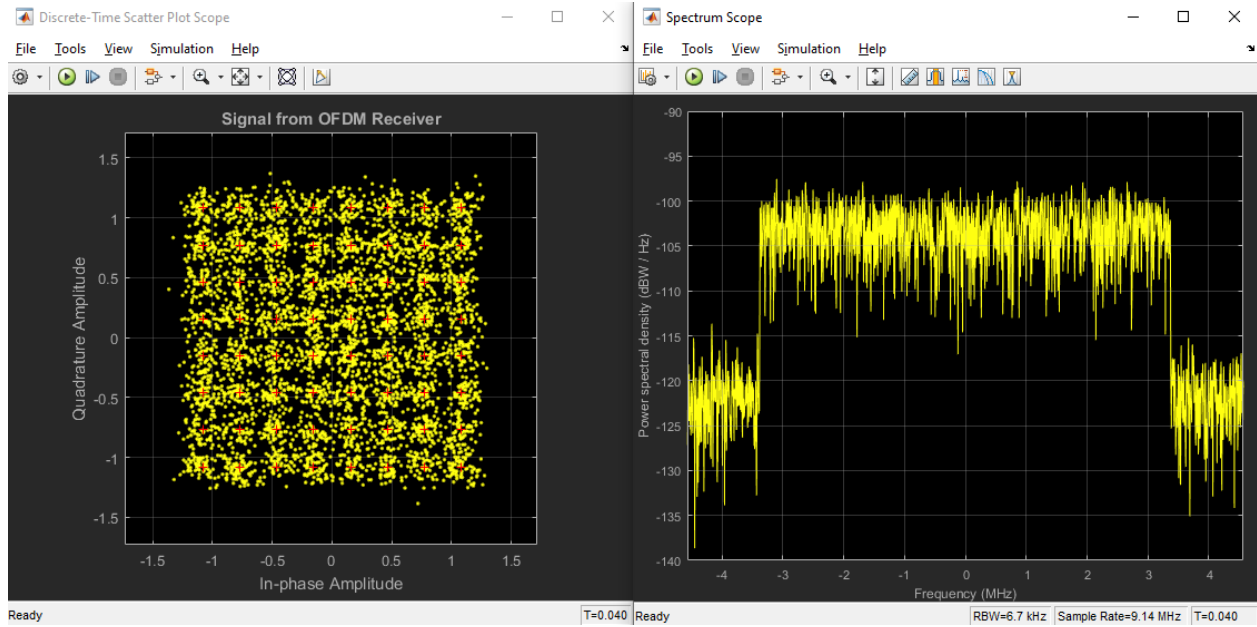


Figure 42 Scatter Plot & Spectrum Scope for 64-QAM

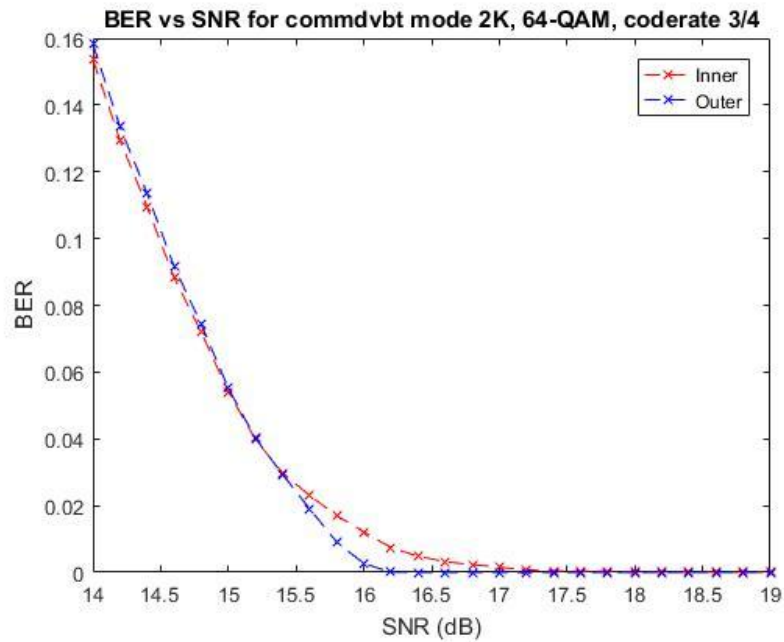


Figure 43 Διάγραμμα BER vs SNR for 64-QAM

Στη συνέχεια ακολουθούν κάποια συμπεράσματα που μπορέσαμε να βγάλουμε από τις παραπάνω μετρήσεις.

Μέθοδος Διαμόρφωσης	N1 (dB)	N2 (dB)	N3 (dB)	Bit Rate (Mbit/sec)
QPSK	4.25	4.75	5.5	8.797
16-QAM	10.3	10.75	11.6	18.047
64-QAM	15.4	16.2	17.2	27.325

Στο παραπάνω πινακάκι βλέπουμε πως για κάθε διαμόρφωση έχουμε 4 διαφορετικά αποτελέσματα. Το N1 είναι η τιμή του θορύβου στην οποία αρχίζει να διορθώνει σφάλματα το εξωτερικό σύστημα, το N2 είναι η τιμή θορύβου στην οποία το συνολικό σύστημα έχει BER πρακτικά 0 και N3 είναι η τιμή θορύβου στην οποία το εσωτερικό σύστημα έχει BER πρακτικά 0. Έχουμε και το Bit Rate για κάθε διαμόρφωση το οποίο μπορούμε να το υπολογίσουμε διαιρώντας τα Total Bits με το χρόνο που τρέχει η προσομοίωσή μας.

Αυτό που παρατηρούμε είναι ότι οι τιμές του Bit Rate είναι λίγο μεγαλύτερες από τις πραγματικές. Αυτό οφείλεται στο γεγονός ότι στο πραγματικό σύστημα κάποια bit χρησιμοποιούνται για συγχρονισμό. Επίσης παρατηρούμε ότι με το εξωτερικό σύστημα η ανοχή στο θόρυβο αυξάνεται κατά 1 db περίπου στο 64-QAM. Δηλαδή από 17.2 SNR στα 16.2 SNR. Το ίδιο ισχύει και για τα 16-QAM και QPSK. Τέλος βλέπουμε ότι το 64-QAM έχει το μεγαλύτερο Bit Rate με την μικρότερη ανοχή στο θόρυβο ενώ αντίθετα το QPSK έχει το μικρότερο Bit Rate με τη μεγαλύτερη ανοχή στο θόρυβο.

Στη συνέχεια θα τρέξουμε προσομοιώσεις με κοινά τις διαμορφώσεις (64-QAM) και τα code rate (3/4) αλλά με διαφορετικό mode. Για το διαφορετικό mode αξίζει να δούμε τη διαφορά στο Spectrum Scope για να δούμε την αλλαγή που βλέπουμε πάνω στο φάσμα των συχνοτήτων.

Ακολουθεί το Spectrum Scope για 2k και 8k mode αντίστοιχα καθώς και τα αποτελέσματα από το script:

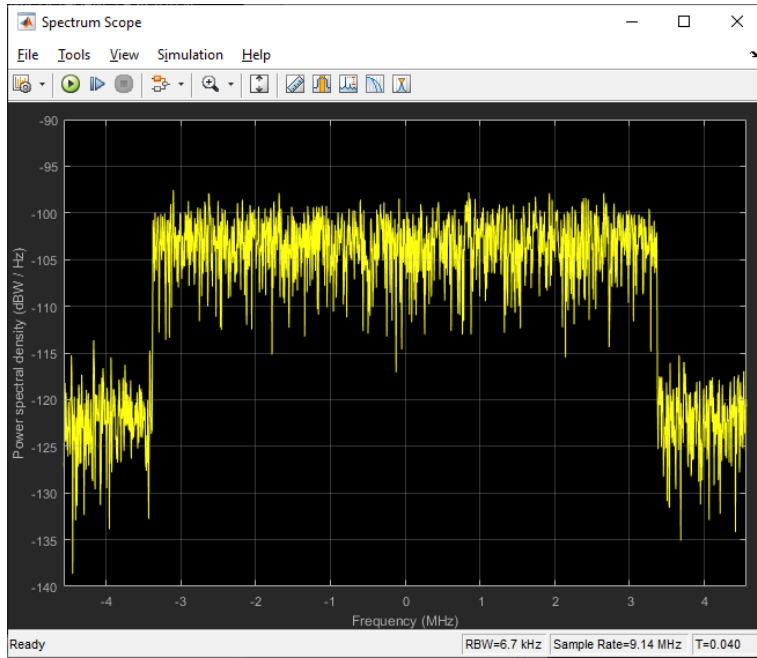


Figure 44 Spectrum Scope 2k mode

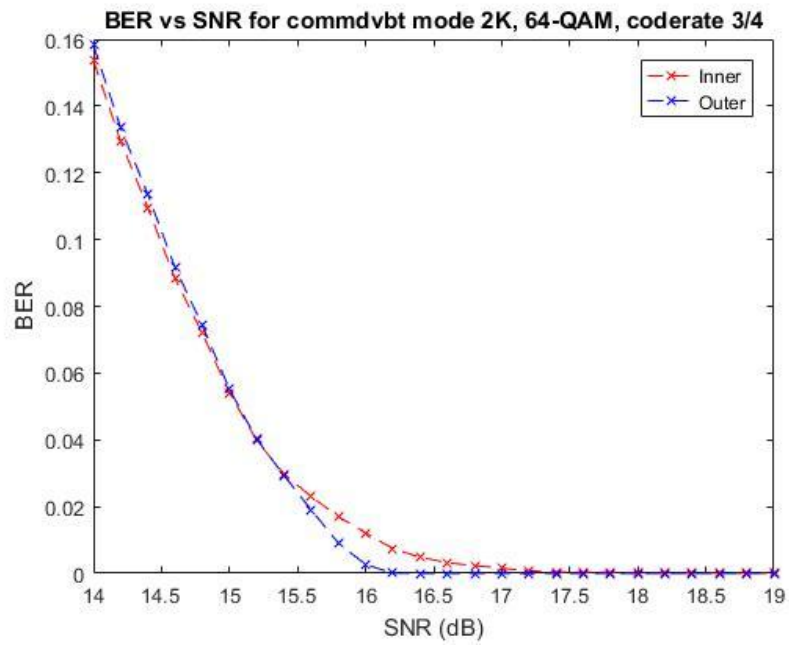


Figure 45 Διάγραμμα 2k mode

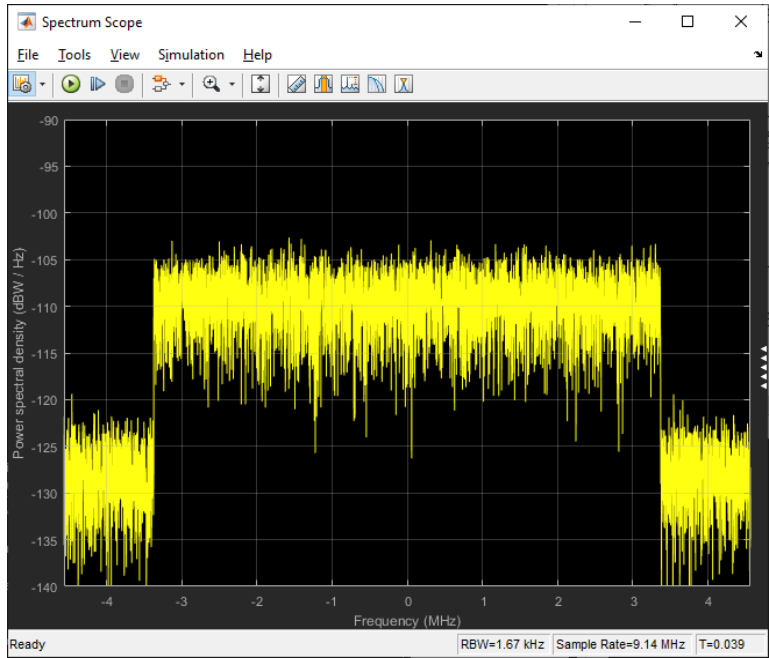


Figure 46 Spectrum Scope 8k mode

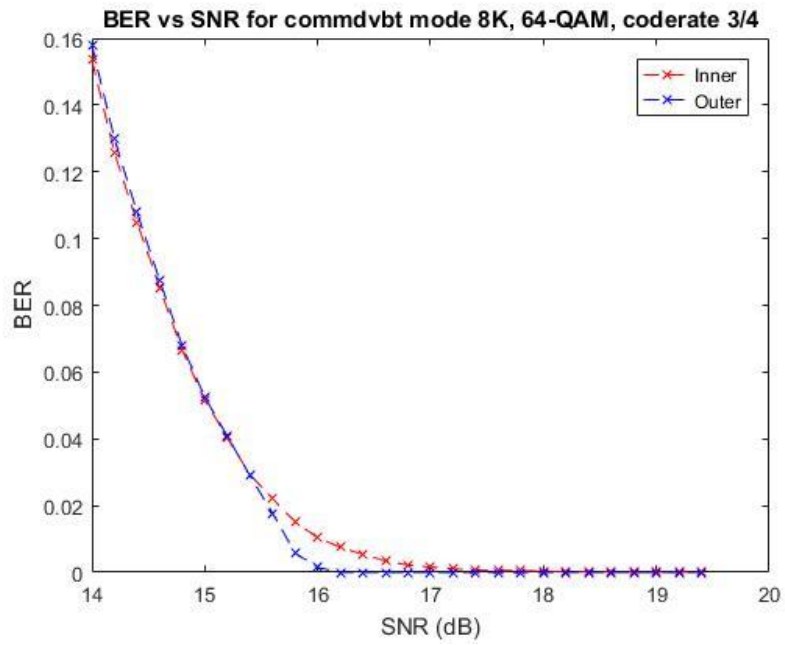


Figure 47 Διάγραμμα 8k mode

Ακολουθούν τα συμπεράσματα από τις παραπάνω μετρήσεις.

Mode	N1 (dB)	N2 (dB)	N3 (dB)	Bit Rate (Mbit/sec)
2k Mode	15.4	16.2	17.2	27.325
8k Mode	15.2	16.2	17.9	26.85

Αυτό που παρατηρούμε είναι ότι το 2k mode με το 8k mode έχουν σχεδόν την ίδια ανοχή στο θόρυβο και επίσης το Bit Rate είναι σχεδόν το ίδιο. Η διαφορά που έχουν μεταξύ τους έχει να κάνει με το ότι το 8k mode είναι πιο ακριβό και ότι η χρήση του σε μεγάλα SFN (Single Frequency Network) είναι προτιμότερη αφού έχει μεγαλύτερα guard intervals σε σχέση με το 2k. Τα guard interval βρίσκονται στην αρχή και στο τέλος του κάθε frame προκειμένου να μην υπάρξουν παρεμβολές από frames άλλων πομπών που πιθανώς να έχουν κάποιο delay.

Στη συνέχεια έτρεξα κάποιες προσομοιώσεις με κοινά τα mode (2k) και τις διαμορφώσεις (64-QAM) και άλλαξα τα code rates. Στην συνέχεια μπορούμε να δούμε τα αποτελέσματα από τις μετρήσεις.

Οπότε για τα code rate 1/2, 2/3, 3/4, 5/6 και 7/8 έχουμε αντίστοιχα τις παρακάτω μετρήσεις.

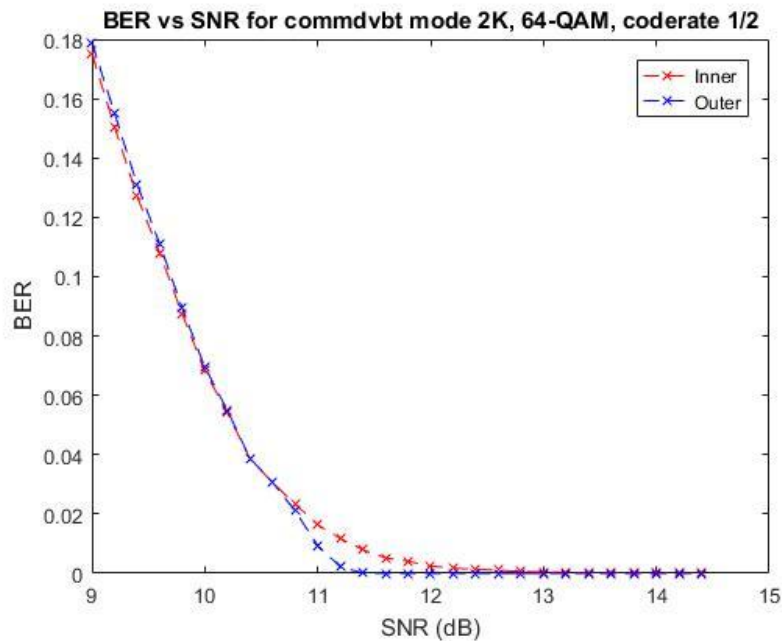


Figure 48 Code Rate 1/2

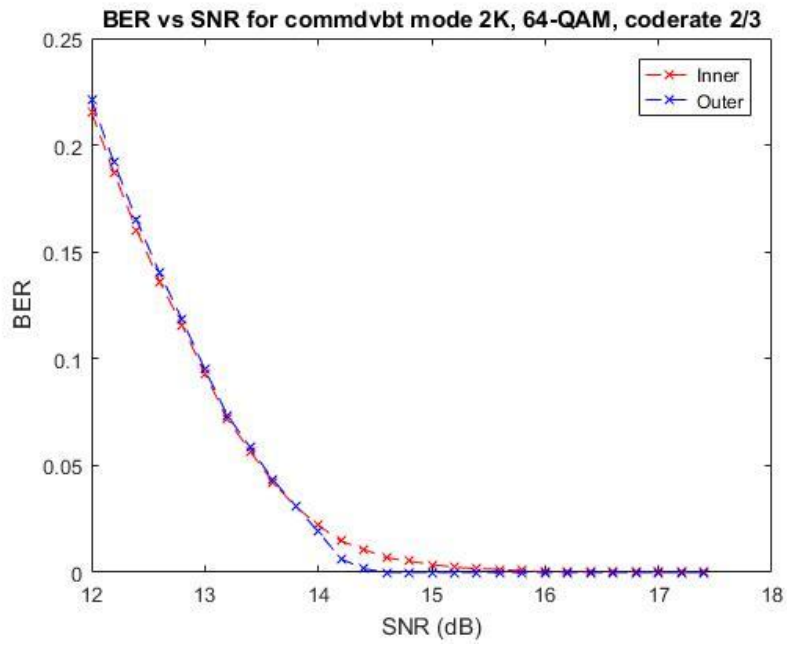


Figure 49 Code Rate 2/3

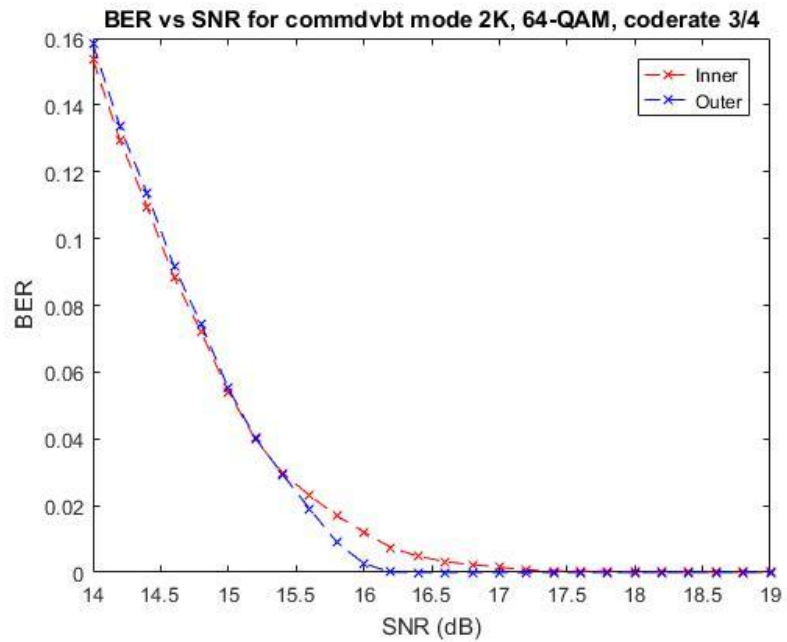


Figure 50 Code Rate 3/4

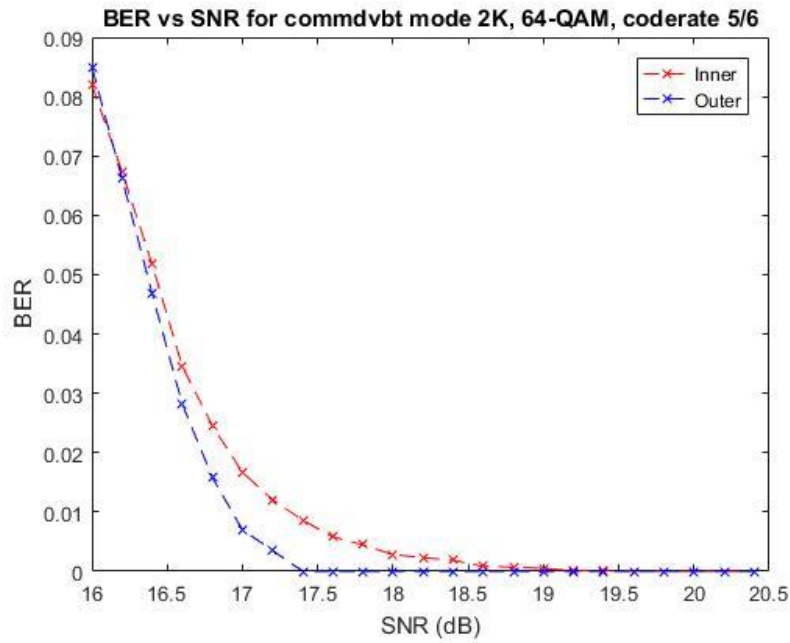


Figure 51 Code Rate 5/6

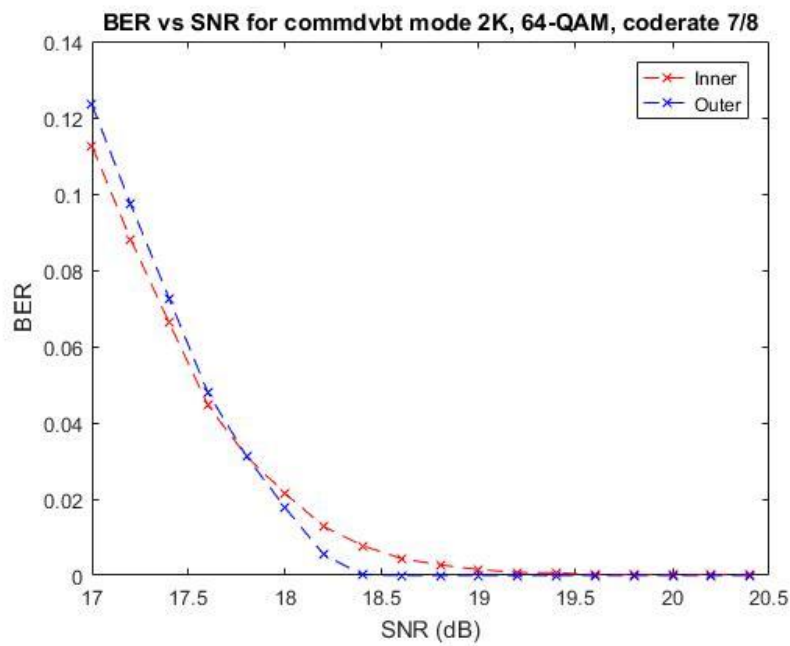


Figure 52 Code Rate 7/8

Τα συμπεράσματα που βγάζουμε από τις παραπάνω μετρήσεις είναι τα εξής:

Code Rate	N1 (dB)	N2 (dB)	N3 (dB)	Bit Rate (Mbit/sec)
1/2	10.5	11.4	13	18.047
2/3	13.7	14.7	15.9	24.215
3/4	15.4	16.2	17.2	27.325
5/6	16.2	17.4	19.2	29.8
7/8	17.8	18.4	19.5	30.46

Βλέπουμε πως όσο μεγαλώνει το Code Rate τόσο μεγαλώνει και το Bit Rate. Επίσης βλέπουμε ότι όσο πιο μικρό είναι το Code Rate τόσο πιο ανθεκτικό είναι προς τον θόρυβο.

Ένα γενικό συμπέρασμα που μπορούμε να βγάλουμε από τις παραπάνω μετρήσεις είναι ότι υπάρχει trade-off μεταξύ Bit Rate και ανοχής στο θόρυβο. Οπότε δεν μπορούμε να έχουμε και μεγάλο Bit Rate και μεγάλη ανοχή στο θόρυβο. Οπότε μπορούμε να καταλήξουμε στο ότι το πιο ανθεκτικά δυνατό σύστημα που μπορούμε να φτιάξουμε είναι το με διαμόρφωση QPSK mode 2K και code rate 1/2. Αντίστοιχα το λιγότερο ανθεκτικό αλλά με το μεγαλύτερο Bit Rate είναι με διαμόρφωση 64-QAM, 8k mode και code rate 7/8. (Standard, 1997-2008)

ΚΕΦΑΛΑΙΟ 5: ΣΥΝΟΨΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η τηλεόραση αποτελεί ένα μέσο του οποίου η ανακάλυψη έγινε, όχι από έναν, αλλά από τη συνεργασία πολλών ατόμων. Πρόκειται για ένα μέσο που αρχικά δεν ήταν ιδιαίτερα προσιτό για αρκετές οικογένειες μιας και δεν είχαν όλα τα σπίτια πρόσβαση σε αυτό. Αλλά με τη πάροδο του χρόνου αυτό άλλαξε και πλέον αποτελεί ένα μέσο το οποίο είναι πολύ προσιτό και πλέον κάθε οικογένεια έχει στη κατοχή της τουλάχιστον μία. Αποτελεί τη συσκευή η οποία παρέχει σε κάθε οικογένεια ένα μέσο ενημέρωσης και ψυχαγωγίας. Η τηλεόραση ξεκίνησε ως μια συσκευή που μετέδιδε ασπρόμαυρη εικόνα και με τη πάροδο του χρόνου και την εξέλιξη της τεχνολογίας, εξελίχθηκε και η τηλεόραση. Πλέον όχι μόνο έχει τη δυνατότητα μετάδοσης έγχρωμης εικόνας αλλά πλέον μπορείς μέσω αυτής να σερφάρεις στο internet και να παρακολουθήσεις το πρόγραμμα που επιθυμείς την ώρα που το επιθυμείς χωρίς να βασίζεσαι στο πρόγραμμα του εκάστοτε παρόχου.

Το θέμα της πτυχιακής μου είναι "Μελέτη και παραμετροποίηση του DVB-T πρωτοκόλλου ανάλογα με το περιεχόμενο". Σκοπός της πτυχιακής μου ήταν να μελετήσω το τρόπο με τον οποίο γίνεται η μετάδοση του σήματος επίγεια και στη συνέχεια να τρέξω κάποιες προσομοιώσεις, με τη χρήση του Matlab, αλλάζοντας τις μεταβλητές που μπορούσα και επέτρεπε το standard του DVB να αλλάξω και τέλος να βγάλω κάποια συμπεράσματα. Πιο συγκεκριμένα μελέτησα το πρωτόκολλο DVB-T που χρησιμοποιείται για το σκοπό αυτό.

Το σήμα μας περνάει από διάφορα στάδια τόσο κατά την εκπομπή του όσο και κατά την λήψη του. Πιο συγκεκριμένα στην αρχή το σήμα μας περνάει από έναν external encoder. Ο external encoder που χρησιμοποιούμε είναι ένας Reed-Solomon του οποίου η χρήση γίνεται για να διορθώσει τυχόν σφάλματα από πιθανό θόρυβο κατά τη μετάδοση. Στην συνέχεια το σήμα μας περνάει από έναν convolutional interleaver ο οποίος αποτελείται από κάποιους shift registers. Σκοπός του convolutional interleaver είναι το ανακάτεμα της πληροφορίας έτσι ώστε να μην υπάρχουν μαζεμένα λάθη στο ίδιο μέρος της πληροφορίας που θέλουμε να στείλουμε. Στο επόμενο βήμα το σήμα μας περνάει από έναν convolutional code. Σε αυτό φτιάχνει ένα διάγραμμα Trellis με τη βοήθεια του πολυωνύμου που του ορίζουμε εμείς. Το διάγραμμα Trellis αποτελείται από έναν αριθμό shift registers που βρίσκονται σε σειρά και από το πολυώνυμο που του έχουμε ορίσει προκύπτει ο αριθμός των αθροιστών modulo 2. Από αυτό το βήμα προκύπτει

και το code rate. Στη συνέχεια περνάει από έναν inner interleaver ο οποίος αποτελείται από 2 block interleavers. Οι δύο αυτοί block interleavers πρόκειται στην ουσία για πίνακες που στη πραγματικότητα ανακατεύουν παραπάνω τη πληροφορία που αποστέλλουμε. Μετά ακολουθεί ο mapper ο οποίος ανάλογα τη διαμόρφωση παίρνει συγκεκριμένο αριθμό bit (π.χ. σε 64-QAM παίρνει 6-bit αριθμούς, στο 16-QAM παίρνει 4-bit κτλ). Στη συνέχεια τους βάζει σε ένα διάγραμμα αστερισμών σύμφωνα με τις θέσεις που του έχουμε βάλει ανάλογα τη διαμόρφωση. Έτσι καταλήγουμε σε μιγαδικούς αριθμούς και ο καθένας έχει το δικό του πλάτος και φάση. Μετά ακολουθεί ο OFDM ο οποίος αυτό που κάνει είναι να παίρνει τους μιγαδικούς αυτούς και να τους βάζει στο φάσμα συχνοτήτων. Επίσης προσθέτει τα guard bands στην αρχή και στο τέλος της πληροφορίας που στέλνουμε των οποίων σκοπός είναι να αποφύγουμε τυχόν παρεμβολές από άλλα σήματα. Τέλος γίνεται η αποστολή του σήματος. Τώρα από τη πλευρά του δέκτη γίνεται η ίδια διαδικασία απλά με ανάποδη σειρά.

Στη συνέχεια με τη βοήθεια του Matlab έτρεξα κάποιες προσομοιώσεις αλλάζοντας τη διαμόρφωση (QPSK, 16-QAM και 64-QAM), το code rate ($1/2$, $2/3$, $3/4$, $5/6$ και $7/8$) και το mode (2k και 8k). Όσον αφορά τις διαμορφώσεις, η 64-QAM είχε το μεγαλύτερο bit rate αλλά τη μικρότερη ανοχή στο θόρυβο ενώ αντίθετα η QPSK είχε τη μεγαλύτερη ανοχή στο θόρυβο αλλά το μικρότερο bit rate. Η 16-QAM αποτελεί τη μέση λύση μιας και ούτε το μεγαλύτερο bit rate έχει αλλά ούτε και τη μεγαλύτερη ανοχή στο θόρυβο.

Όσον αφορά τα code rate διαπιστώσαμε ότι όσο μεγαλύτερο είναι το code rate τόσο μεγαλύτερο είναι και το bit rate και αντίθετως μικραίνει η ανοχή στο θόρυβο. Έτσι το $1/2$ έχει το μικρότερο bit rate αλλά τη μεγαλύτερη ανοχή και το $7/8$ έχει το μεγαλύτερο bit rate αλλά τη μικρότερη ανοχή στο θόρυβο.

Τέλος στα διαφορετικά mode (2k και 8k) δεν βλέπουμε κάποια διαφορά ως προς την ανοχή στο θόρυβο και στο bit rate. Η διαφορά που έχουν μεταξύ τους έχει να κάνει με το ότι το 8k mode είναι πιο ακριβό και ότι η χρήση του σε μεγάλα SFN (Single Frequency Network) είναι προτιμότερη αφού έχει μεγαλύτερα guard intervals σε σχέση με το 2k. Τα guard interval βρίσκονται στην αρχή και στο τέλος του κάθε frame προκειμένου να μην υπάρξουν παρεμβολές από frames άλλων πομπών που πιθανώς να έχουν κάποιο delay.

Ένα γενικό συμπέρασμα που μπορούμε να βγάλουμε από τα παραπάνω είναι ότι υπάρχει ένα trade-off μεταξύ bit rate και ανοχής στο θόρυβο. Οπότε δεν μπορούμε να έχουμε και μεγάλο bit rate και μεγάλη ανοχή στο θόρυβο. Για αυτό το λόγο χρησιμοποιούμε ανάλογα με τα δεδομένα το αντίστοιχο code rate, mode και την αντίστοιχη διαμόρφωση.

Bibliography

cs.cmu.edu. (1998). *www.cs.cmu.edu*. Retrieved from [www.cs.cmu.edu](http://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html):
https://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html

radio-electronics. (n.d.). Retrieved Αύγουστος 19, 2015, from <http://www.radio-electronics.com/>:
<http://www.radio-electronics.com/info/broadcast/digital-video-broadcasting/what-is-dvb-t-basics-tutorial.php>

Standard, E. (1997-2008). *Digital Video Broadcasting (DVB)*.

Wikipedia. (2015, Αύγουστος 6). *Wikipedia*. Retrieved Αύγουστος 18, 2015, from en.wikipedia.org:
https://en.wikipedia.org/wiki/Philo_Farnsworth

Wikipedia. (2015, Αυγουστος 16). *Wikipedia*. Retrieved Αύγουστος 19, 2015, from en.wikipedia.org:
https://en.wikipedia.org/wiki/History_of_television

Wikipedia. (2015, Ιούλιος 25). *Wikipedia*. Retrieved Σεπτέμβριος 1, 2015, from en.wikipedia.org:
<https://en.wikipedia.org/wiki/DVB-T>

wikipedia. (2019, May 6). *wikipedia*. Retrieved from wikipedia:
https://en.wikipedia.org/wiki/Burst_error-correcting_code

wikipedia. (2019, May 6). *wikipedia*. Retrieved from wikipedia:
https://en.wikipedia.org/wiki/Burst_error-correcting_code

wikipedia. (2020, March 6). *wikipedia*. Retrieved from wikipedia:
https://en.wikipedia.org/wiki/Viterbi_decoder

Wikipedia. (2020, May 5). *Wikipedia*. Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/Convolutional_code

Σαραντής, Κ. Τ.-Π. (2014). *Αναλογική και ψηφιακή τηλεόραση*. ΑΘ. ΣΤΑΜΟΥΛΗΣ.