



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Συγχρονισμός - Άσκηση 4

Οικονομίδης Κωνσταντίνος (ΑΜ:2021030075)
Διδάσκων: Αθανασιος Π. Λιάβας

Πολυτεχνείο Κρήτης - Σχολή ΗΜΜΥ
Ασύρματες Επικοινωνίες

Χειμερινό εξάμηνο 2025-2026

Σε αυτή την άσκηση, θα μελετήσουμε τεχνικές συγχρονισμού, εκτίμησης CFO και καναλιού σε ιδανικά και μη-ιδανικά κανάλια.

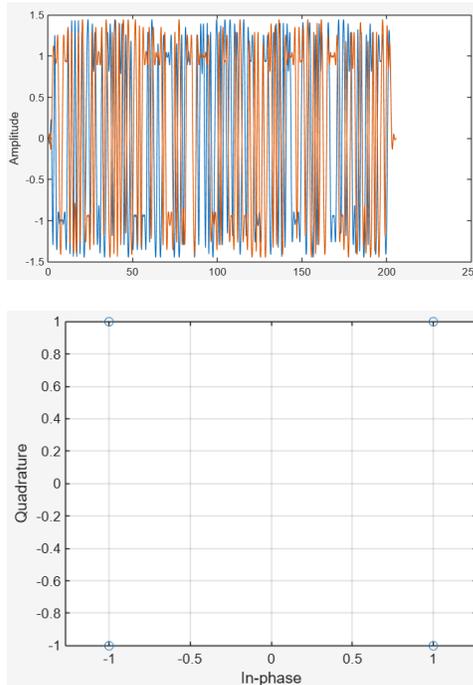
Στα παρακάτω, T και T_s είναι αντίστοιχα, η περίοδος συμβόλου και δείγματος, με $\frac{T}{T_s} = \text{over} \in \mathbb{N}$, N το πλήθος των συμβόλων σε κάθε πακέτο, και N_{tr} το πλήθος των συμβόλων εκπαίδευσης σε κάθε πακέτο. Τα φίλτρα στον πομπό και στο δέκτη είναι SRRC, με roll-off factor $\beta=0.4$ και μισό μήκος ίσο με $B=3$ περιόδους συμβόλου.

1. Αρχικά, θα ασχοληθούμε με ιδανικά κανάλια.

1.1 Να δημιουργήσετε πακέτο με 4-QAM σύμβολα, A_0, \dots, A_{N-1} . Να θεωρήσετε τα πρώτα N_{tr} σύμβολα ως σύμβολα εκπαίδευσης. Να περάσετε το πακέτο από το φίλτρο μορφοποίησης και να υπολογίσετε το χαμηλοπερατό ισοδύναμο της εισόδου του καναλιού

$$s(t) = \sum_{n=0}^{N-1} A_n g_T(t - nT). \quad (1)$$

Από τον κώδικα, πήραμε τα εξής αποτελέσματα:

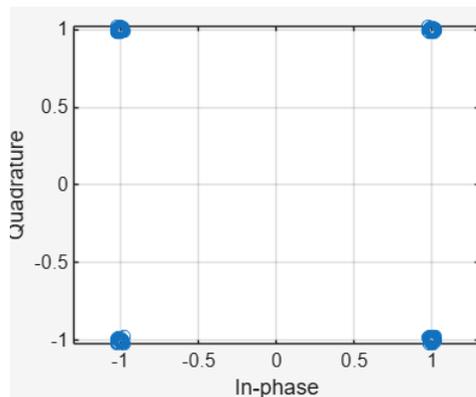


Εδώ δημιουργήθηκε ένα πακέτο 4-QAM συμβόλων με $N=200$. Τα πρώτα 50 ορίζονται ως training symbols και περνάμε το πακέτο από SRRC με στοιχεία: περίοδο συμβόλου T , oversampling factor = 10, roll-off $\beta = 0.4$ και $B=3$.

Αφού το περάσουμε, η έξοδος μας δίνει το διακριτό χαμηλοπερατό ισοδύναμο της κυματομορφής εισόδου (1). Από τις 2 γραφικές, φαίνεται ότι δημιουργούνται όντως 4-QAM σύμβολα και η χαρακτηριστική όπου με μπλε απεικονίζεται το πραγματικό μέρος και με πορτοκαλί το φανταστικό μέρος είναι η αναμενόμενη, όπως είχαμε δει στην τάξη.

- 1.2 Να θεωρήσεις ότι η κρουστική απόκριση του φυσικού καναλιού είναι $c(t) = \delta(t)$ και ότι το κανάλι είναι αθόρυβο.

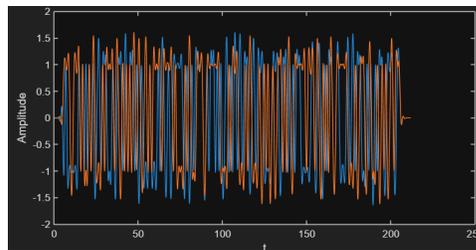
Με βάση τον κώδικα, λάβαμε το εξής αποτέλεσμα:



Σε αυτό το ερώτημα, θεωρήσαμε ότι η κρουστική απόκριση του καναλιού είναι $c(t) = \delta(t)$ και ότι το κανάλι είναι αθόρυβο. Στο γράφημα, φαίνεται ότι τα σύμβολα συγκεντρώνονται στα 4 σημεία του 4-QAM με κάποιες μικρές αποκλίσεις.

- 1.3 Να φιλτράρετε με το προσαρμοσμένο φίλτρο στο δέκτη (να θεωρήσετε ότι ο χρόνος στην έξοδο του προσαρμοσμένου φίλτρου αρχίζει στο $t=0$).

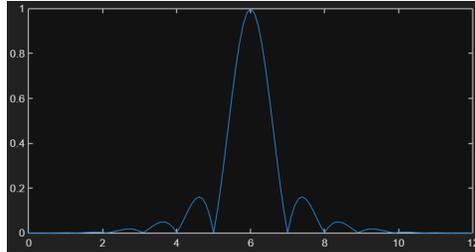
Από τον κώδικα, πήραμε το εξής αποτέλεσμα, για να σιγουρευτούμε ότι όλα είναι οκ με το φίλτρο στο δέκτη.



Σε αυτό το ερώτημα, το σήμα μπαίνει από φίλτρο SRRC στον δέκτη. Άρα η έξοδος δίνεται από την συνέλιξη: $y(t) = A(t) * g_R(t)$. Το σχήμα που πήραμε εδώ μοιάζει με το προηγούμενο αλλά είναι λίγο πιο φιλτραρισμένο.

- 1.4 Να υπολογίσετε το σύνθετο αναλογικό κανάλι, υποθέτοντας ότι ο χρόνος αρχίζει στο $t=0$. Να σχεδιάσετε την απόλυτη τιμή του.

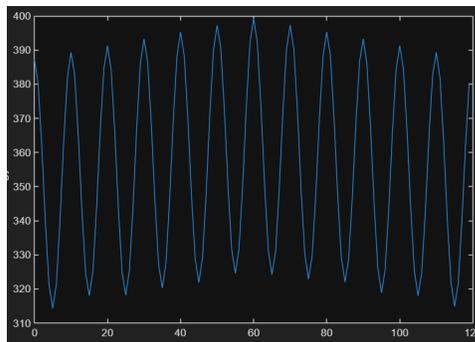
Στη συνέχεια παρατίθενται το αποτέλεσμα που έδωσε ο κώδικας:



Σε αυτό το ερώτημα, υπολογίζεται το σύνθετο αναλογικό κανάλι. Η συνέλιξη που πραγματοποιείται όπως είδαμε στην τάξη ισούται με $g_T(t) * c(t) * g_R(t)$. Η γραφική που παίρνουμε, μοιάζει αρκετά με raised-cosine.

- 1.5 Να προσπαθήσετε να συγχρονιστείτε με βάση την ενέργεια του ληφθέντος σήματος, θεωρώντας ότι $d=0, \dots, 4A_{over}-1$. Να σχεδιάσετε το κατάλληλο στατιστικό. Τι παρατηρείτε;

Το πρόγραμμα παράγει το εξής αποτέλεσμα:

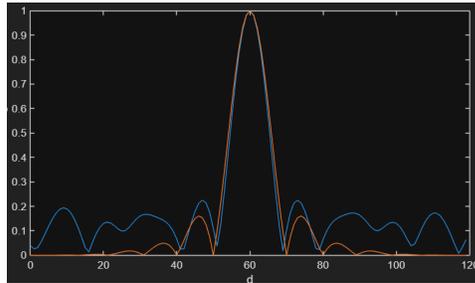


Σε αυτό το ερώτημα, προσπαθήσαμε να συγχρονιστούμε, θεωρώντας $d=0, \dots, 4A_{over}-1$.

Στο γράφημα, φαίνεται μια κυματομορφή η οποία έχει μέγιστο στο $d=60$ ($2A \cdot over = 2 \cdot 3 \cdot 10$). Στην τιμή αυτή δηλαδή η ενέργεια των δειγμάτων γίνεται μέγιστη άρα πετυχαίνεται μεγάλη ακρίβεια.

- 1.6 Να προσπαθήσετε να συγχρονιστείτε με βάση τα σύμβολα εκπαίδευσης, για τα ίδια d . Να σχεδιάσετε την απόλυτη τιμή του αντίστοιχου στατιστικού και να τη συγκρίνετε με την απόλυτη τιμή του σύνθετου καναλιού. Τι παρατηρείτε;

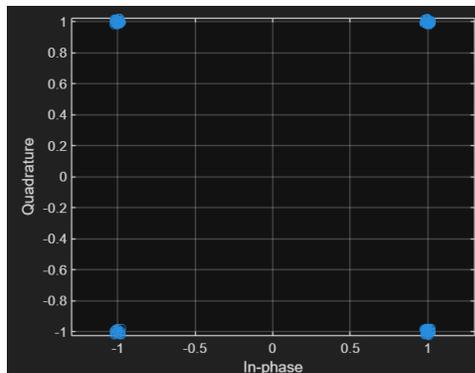
Ο κώδικας, έδωσε τα εξής αποτελέσματα:



Εδώ, χρησιμοποιούμε τα training symbols για να συγχρονιστούμε. Στο γράφημα, το μπλε (με τα training) έχει μέγιστο στο 60 και κάποιους μικρότερους λοβούς, λίγο παραμορφωμένους, πιθανών λόγω του πεπερασμένου μήκους των training symbols. Το πορτοκαλί, αντιπροσωπεύει το σύνθετο κανάλι και παρουσιάζει και αυτό μέγιστο στο 60 και κάποιους μικρότερους λοβούς δεξιά και αριστερά. Επομένως και τα δυο εντοπίζουν σωστά το σημείο χρονισμού.

- 1.7 Αφού συγχρονιστείτε με έναν από τους δύο παραπάνω τρόπους, να κατασκευάσετε την μήκους N symbol-spaced ακολουθία εξόδου και να τη σχεδιάσετε. Τι παρατηρείτε;

Ακολουθεί το αποτέλεσμα του κώδικα:



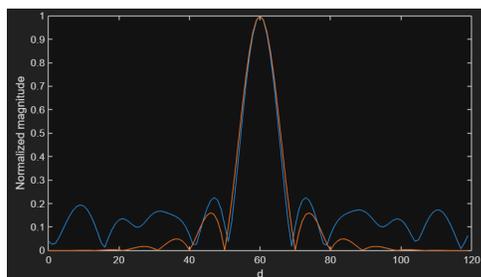
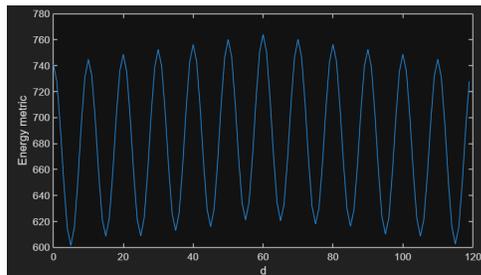
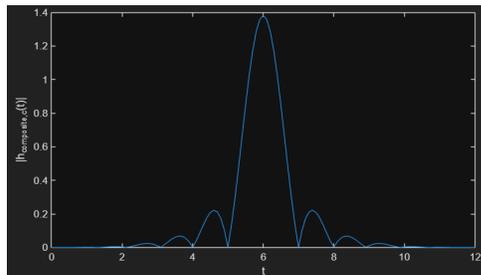
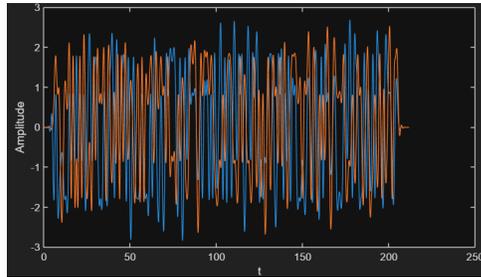
Εδώ, γίνεται πρώτα ο έλεγχος ότι βρέθηκε η σωστή καθυστέρηση με μια από τις 2 μεθόδους συγχρονισμού. Μετά, κανονικοποιούμε την ακολουθία ως προς το κέρδος του συνολικού παλμού για να αντιστοιχίσουμε σωστά τα δείγματα με τα εκτιμώμενα σύμβολα.

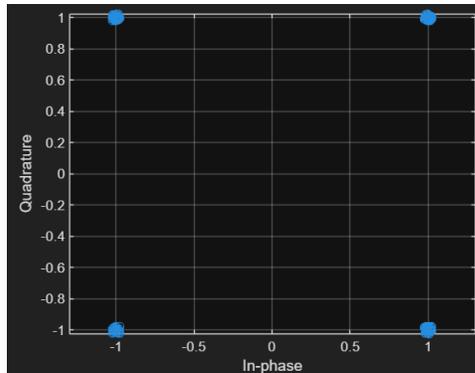
Από το γράφημα, φαίνεται ότι τα σύμβολα συγκεντρώνονται γύρω από τα 4 σημεία του 4-QAM, δηλαδή παίρνουμε τα αρχικά σύμβολα με μικρές αποκλίσεις.

- 1.8 Να επαναλάβετε τα παραπάνω με τη μόνη διαφορά στο ότι $c(t) = c\delta(t)$, με $c \in \mathbb{C}$. Τι

παρατηρείτε;

Τα αποτελέσματα του κώδικα, είναι τα εξής:



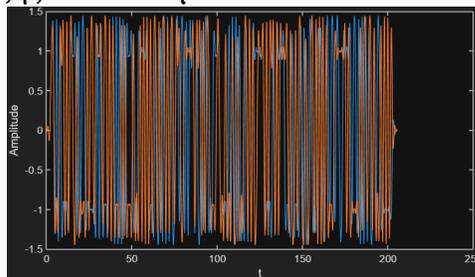


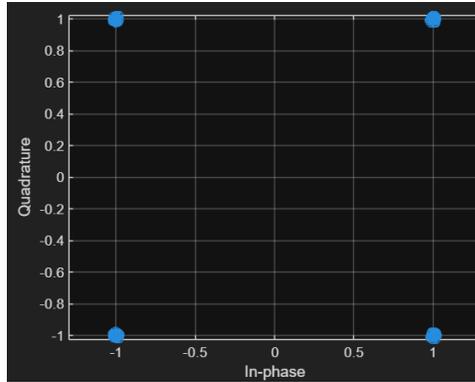
Σε αυτό το ερώτημα, επαναλήφθηκε η ίδια διαδικασία για $c(t) = c\delta(t)$, με $c \in \mathbb{C}$. Δηλαδή το σήμα στην είσοδο πολλαπλασιάζεται με έναν παράγοντα c . Παρατηρείται, ότι τα γραφήματα δεν έχουν αλλάξει σχεδόν καθόλου (απόκριση σύνθετου καναλιού έχει λίγο διαφορετικό ύψος - τα στατιστικά συγχρονισμού έχουν ίδιο σχήμα και μέγιστο στην ίδια τιμή). Τέλος η σχεδίαση της ακολουθίας εξόδου μας δίνει παρόμοιο γράφημα με αυτό που είχαμε στο ερώτημα (1.7), άρα ο c δεν αλλάζει τη δομή του σήματος, αλλά λίγο το περιστρέφει.

2. Στο δεύτερο μέρος, στο ιδανικό κανάλι θα προσθέσουμε CFO.

2.1 Να επαναλάβετε τα βήματα 1.1 και 1.2.

Ο κώδικας, έδωσε τα εξής αποτελέσματα:





Σε αυτό το βήμα επαναλαμβάνουμε τα βήματα 1.1 και 1.2. Δημιουργούμε νέο 4-QAM πακέτο συμβόλων $0, \dots, -1$. Υποθέτουμε και ιδανικό κανάλι $c(t)=\delta(t)$ χωρίς θόρυβο και χρησιμοποιούμε SRRC φίλτρο στον δέκτη. Τώρα δειγματοληπτούμε στην περίοδο συμβόλου και κανονικοποιούμε ως προς το κέρδος του συνολικού παλμού. Παίρνουμε λοιπόν το διάγραμμα του αστερισμού όπου φαίνονται τα 4 σημεία του αστερισμού.

2.2 Να υποθέσετε την ύπαρξη CFO ΔF . Τότε, το σήμα στην είσοδο του δέκτη θα ισούται με

$$Y(t) = e^{j(2\pi Ft + \phi)} A(t) \quad (2)$$

Να δειγματοληπτήσετε, λαμβάνοντας

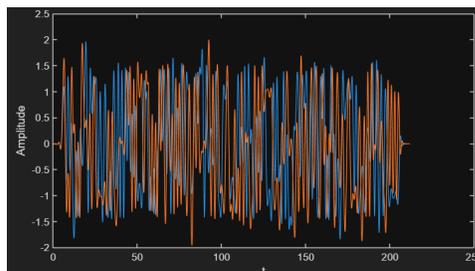
$$Y_n := Y(nT_s) = e^{j(2\pi(FT_s)n + \phi)} A(nT_s) \quad (3)$$

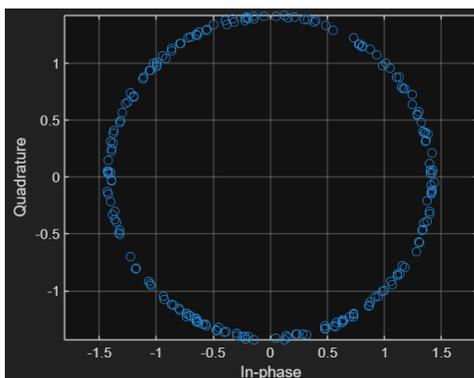
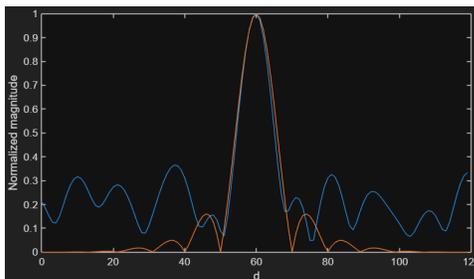
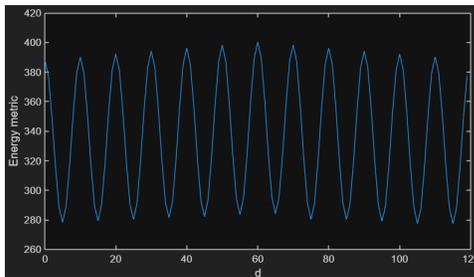
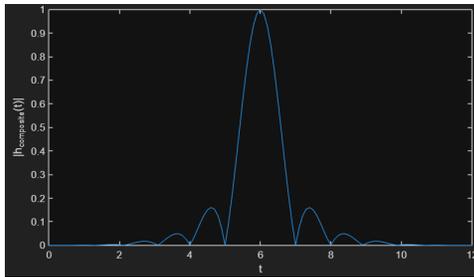
Να θέσετε $F_s := FT_s \approx 10^{-3}$ και $\phi=0$.

Σε αυτό το ερώτημα, εισάγουμε CFO πολλαπλασιάζοντας την κυματομορφή $A(t)$ με τον όρο $e^{j(2\pi Ft + \phi)}$, και ορίζουμε $\Delta F_s = \Delta FT_s \approx 10^{-3}$ και $\phi=0$. Έτσι στην είσοδο του δέκτη έχουμε $Y_n = e^{j(2\pi F_s n + \phi)} A(nT_s)$.

2.3 Να επαναλάβετε τα βήματα 1.3 μέχρι 1.7. Τι παρατηρείτε;

Η εκτέλεση του κώδικα, δίνει τα εξής αποτελέσματα:

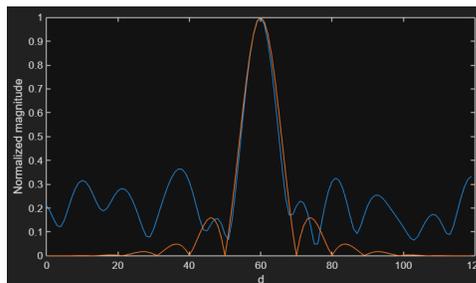
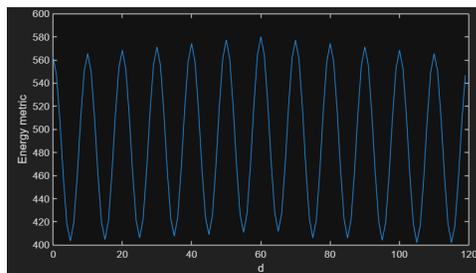
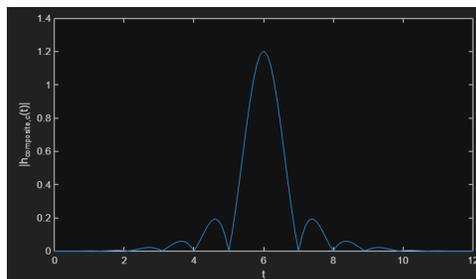
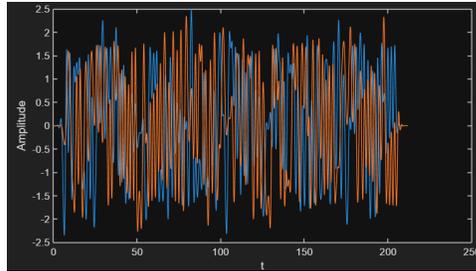


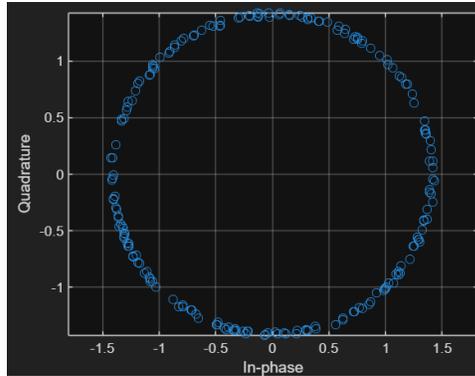


Εδώ, παρατηρούμε ότι η κρουστική απόκριση του σύνθετου καναλιού μένει ίδια. Το στατιστικό συγχρονισμού με βάση την ενέργεια μένει και αυτό ίδιο αλλά το στατιστικό συγχρονισμού με βάση τα training symbols είναι λίγο αλλοιωμένο. Για το διάγραμμα αστερισμού της εξόδου, τα σημεία δεν συγκεντρώνονται στα 4 σημεία 4-QAM αλλά σχηματίζουν έναν κύκλο.

2.4 Να υλοποιήσετε και το βήμα 1.8 ή/και να θέσετε τυχαία φάση Φ , με $\Phi \sim \mathcal{U}[0,2\pi)$. Τι παρατηρείτε;

Ο κώδικας παράγει τα εξής αποτελέσματα:



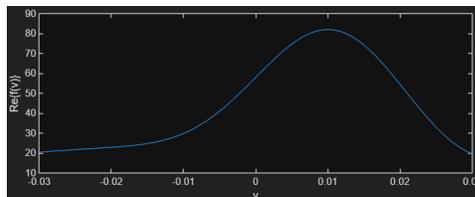


Από τα γραφήματα, φαίνεται ότι η κρουστική απόκριση του σύνθετου καναλιού μένει σχεδόν ίδια, απλά είναι λίγο πιο ψηλό, το στατιστικό συγχρονισμού με βάση την ενέργεια μένει ίδιο, το στατιστικό συγχρονισμού με βάση τα training symbols έχει μέγιστο στο ίδιο σημείο αλλά δεν ακολουθεί τέλεια την κρουστική απόκριση του σύνθετου καναλιού και στο διάγραμμα αστερισμού έχουμε έναν κύκλο.

2.5 Αν έχετε καταφέρει να συγχρονιστείτε, να εκτιμήσετε το CFO με χρήση συμβόλων εκπαίδευσης.

Σύμφωνα με τον κώδικα, τα αποτελέσματα που λαμβάνουμε είναι τα εξής:

```
True ΔF_s = 0.001
Estimated ΔF_s ≈ 0.001001
Estimated ν = ΔF*T ≈ 0.01001
```

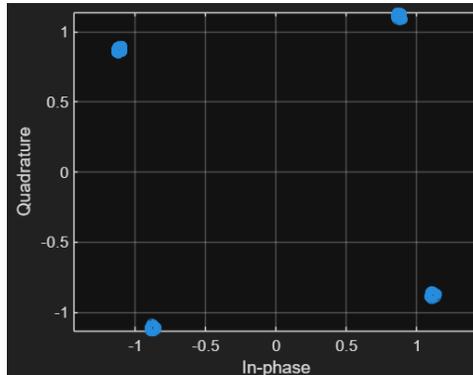


Για να εκτιμήσουμε το CFO, χρησιμοποιούμε τα training symbols στην έξοδο του δέκτη. Από αυτά σχηματίζουμε μια Toeplitz μήτρα η οποία μας βοηθά να μετρήσουμε πόσο καλά ταιριάζουν τα training symbols για κάθε τιμή της CFO. Στο γράφημα, φαίνεται ένα καθαρό μέγιστο στο 0.01001, το οποίο μας βοηθά να βρούμε το $\Delta F_s \approx \frac{0.01001}{\text{over}} \approx 0.001001$, η οποία είναι μια τιμή πολύ κοντά στο $\Delta F_s = 10^{-3}$. Αυτό μας δείχνει ότι λειτουργεί καλά.

2.6 Να αναιρέσετε την επίδραση του CFO και να σχεδιάσετε την μήκους N

symbol-spaced ακολουθία εξόδου. Τι παρατηρείτε;

Με βάση τον κώδικα, λαμβάνουμε το αποτέλεσμα:



Με βάση την εκτιμώμενη τιμή του CFO που βρέθηκε στο προηγούμενο ερώτημα, εφαρμόζουμε τον όρο $e^{-j2\pi u_{est}n}$ για $n=1, \dots, N$ και στη συνέχεια αντισταθμίζουμε το κέρδος του σύνθετου καναλιού.

Στο διάγραμμα αστερισμού που προκύπτει, φαίνεται ότι τα σημεία συγκεντρώνονται γύρω από τα 4-QAM σημεία με κάποιες μικρές αποκλίσεις, άρα η διόρθωση του CFO αντισταθμίζει σε έναν μεγάλο βαθμό την παραμόρφωση περιστροφής.

- 2.7 Να εκτιμήσετε το διακριτό ισοδύναμο κανάλι (στην περίπτωση αυτή το κανάλι έχει μόνο ένα συντελεστή) με τη μέθοδο των ελαχίστων τετραγώνων.

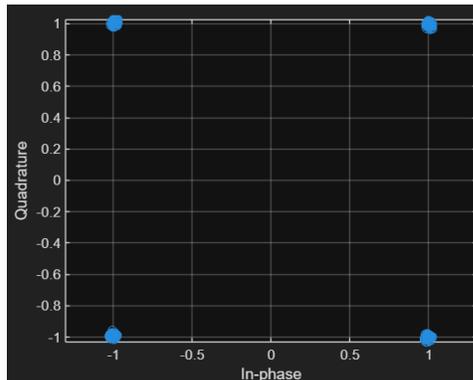
Ο κώδικας, μας έδωσε το εξής αποτέλεσμα:

```
h_LS = 0.99381+0.12413i
h_eq_true = 0.99957
|h_LS - h_eq_true| = 0.12426
```

Σε αυτό το ερώτημα, κάνουμε την εκτίμηση του διακριτού ισοδύναμου καναλιού. Υποθέτουμε το μοντέλο $Y = h \cdot A$ και εφαρμόζουμε τη μέθοδο των ελαχίστων τετραγώνων, από την οποία παίρνουμε την εκτίμηση h_{LS} . Όπως φαίνεται από τον κώδικα το μέτρο της εκτίμησης είναι κοντά στο 1, άρα το ισοδύναμο κανάλι εκτιμάται σωστά.

- 2.8 Να προσπαθήσετε να αναιρέσετε την επίδραση του καναλιού και να σχεδιάσετε την ακολουθία που λαμβάνετε μετά από αυτό. Τι παρατηρείτε;

Ο κώδικας, έδωσε το εξής αποτέλεσμα:

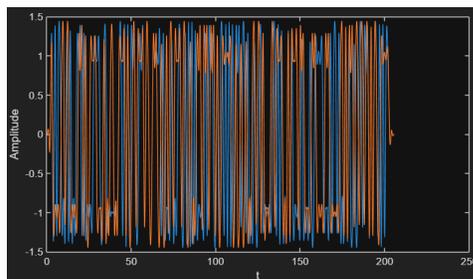


Εδώ, θέλουμε να αναιρέσουμε την επίδραση του καναλιού χρησιμοποιώντας την h_{LS} εκτίμηση από το προηγούμενο ερώτημα. Και εδώ υποθέτουμε το μοντέλο $Y = h \cdot A$ και εφαρμόζουμε ισοσταθμιστή με κέρδος $\frac{1}{h_{LS}}$. Στο διάγραμμα αστερισμού, φαίνονται τα σημεία να συγκεντρώνονται στα 4 σημεία του 4-QAM.

3. Στο τρίτο μέρος, θα ασχοληθούμε με μη-ιδανικά κανάλια χωρίς CFO.

3.1 Να υλοποιήσετε το βήμα 1.1

Από την εκτέλεση του κώδικα λάβαμε τα εξής αποτελέσματα:



Εδώ, όπως και στο ερώτημα 1.1 δημιουργήθηκε ένα νέο πακέτο 4-QAM συμβόλων με τα πρώτα N_{tr} ως σύμβολα εκπαίδευσης. Η γραφική που παίρνουμε, αποτελεί την είσοδο και είναι παρόμοια με αυτή που πετύχαμε στο ερώτημα 1.1.

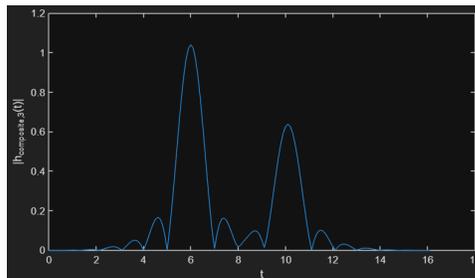
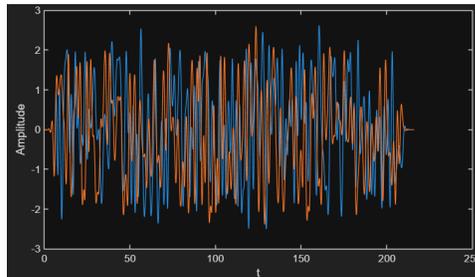
3.2 Να θερήσετε ότι η κρουστική απόκριση του φυσικού καναλιού είναι

$$c(t) = c_0\delta(t) + c_1\delta(t - KT_s) \quad (4)$$

Σε αυτό το βήμα, θεωρούμε το κανάλι με κρουστική απόκριση:
 $c(t) = c_0\delta(t) + c_1\delta(t - KT_s)$, όπου ο c_0 δεν έχει κάποια καθυστέρηση και ο c_1 καθυστερεί κατά $t - KT_s$.

3.3 Να υλοποιήσετε τα βήματα 1.3 και 1.4.

Ακολουθούν τα αποτελέσματα του κώδικα για το συγκεκριμένο ερώτημα:

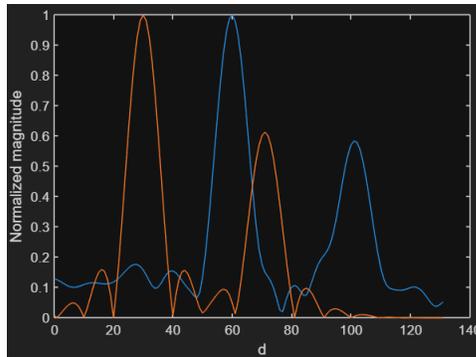


Σε αυτό το ερώτημα, υπολογίζουμε το σύνθετο αναλογικό κανάλι ως:
 $h_{composite}(t) = g_T(t) * c(t) * g_R(t)$. Όπως φαίνεται στο γράφημα, το σύνθετο αναλογικό κανάλι εμφανίζει δύο λοβούς, έναν για τον όρο $c_0\delta(t)$ και έναν για τον όρο $c_1\delta(t - KT_s)$
 Επίσης για την έξοδο του φίλτρου, φαίνεται ότι είναι λίγο πιο μπλεγμένη από το ιδανικό κανάλι λόγω της ύπαρξης των 2 όρων.

3.4 Να υλοποιήσετε το βήμα 1.6, υπολογίζοντας την ποσότητα $corr_d$, για

$$d = 0, \dots, (4A_{over} + 1) + (K + 2) - 2$$

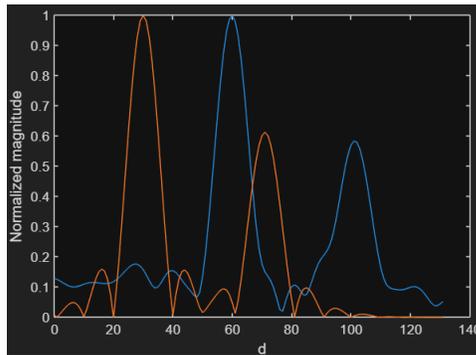
Ο κώδικας οδηγεί στο παρακάτω αποτέλεσμα



Σε αυτό το ερώτημα, υλοποιούμε το βήμα 1.6 και όπως φαίνεται, τα 2 μέγιστα των 2 καμπυλών εμφανίζονται περίπου στα ίδια σημεία, πράγμα που υποδεικνύει ότι ο συγχρονισμός είναι σωστός.

- 3.5 Να υπολογίσετε την ποσότητα \mathcal{E}_d , υποθέτοντας μήκος ισοδύναμου διακριτού καναλιού M (να βρείτε το κατάλληλο διάστημα για τις τιμές του d), και να συγχρονιστείτε, λαμβάνοντας μία symbol-spaced ακολουθία μήκους $N+M-1$.

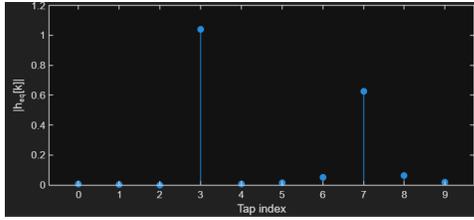
Τρέχοντας τον κώδικα, προκύπτει το παρακάτω αποτέλεσμα:



Από το γράφημα, φαίνεται ότι το \mathcal{E}_d παρουσιάζει μέγιστα κάθε μια περίοδο συμβόλου και η ενέργεια μεγιστοποιείται για $d=30$, κάτι το οποίο δίνεται και από την κονσόλα του matlab. Έτσι η ακολουθία εξόδου έχει μήκος $N+M-1 = 209$.

- 3.6 Να εκτιμήσετε το διακριτό ισοδύναμο κανάλι μήκους M , με τη μέθοδο των ελαχίστων τετραγώνων.

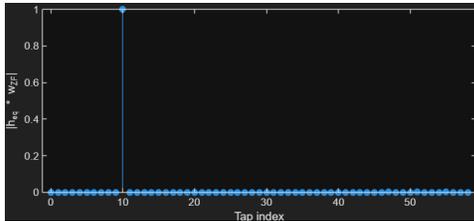
ο κώδικας επιστρέφει τα ακόλουθα αποτελέσματα:



Φαίνεται από το σχήμα, ότι υπάρχουν 2 μεγάλα taps στους δείκτες 3 και 7 και τα υπόλοιπα είναι πολύ μικρότερα.

3.7 Να υπολογίσετε τον ZF ισοσταθμιστή, μήκους $L \approx 5M$, για καθυστέρηση $\delta=M$.

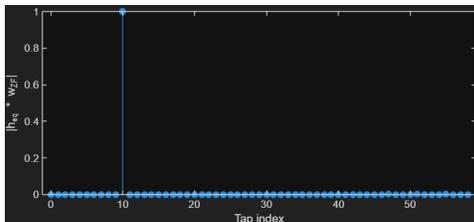
Ο κώδικας, έδωσε τα εξής αποτελέσματα:



Σε αυτό το ερώτημα, υπολογίζουμε έναν ZF ισοσταθμιστή με μήκος $L \approx 5M$ και καθυστέρηση $\delta=M$. Στο γράφημα, φαίνεται ένα μεγάλο tap στη θέση $k=M$ και τα υπόλοιπα είναι σχεδόν μηδεν, άρα ο ισοσταθμιστής δουλεύει σωστά.

3.8 Να φιλτράρετε την symbol-spaced ακολουθία εξόδου και να αποκόψετε τα πρώτα δ δείγματα στην έξοδο του ισοσταθμιστή. Να σχεδιάσετε τα επόμενα N δείγματα. Τι παρατηρείτε;

Η εκτέλεση του κώδικα, έδωσε το εξής αποτέλεσμα:

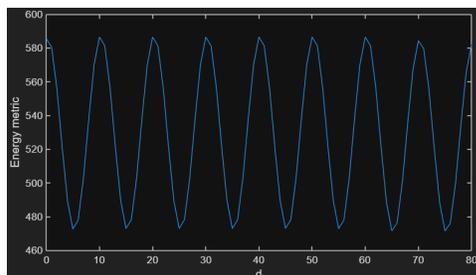
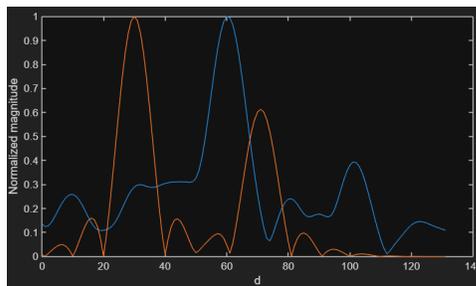
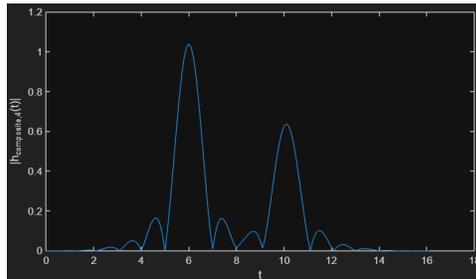
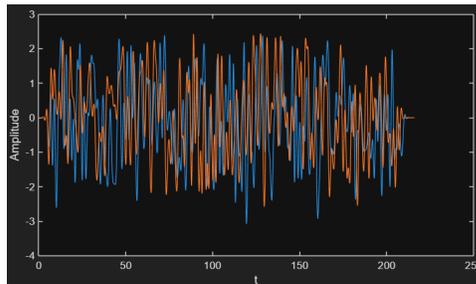


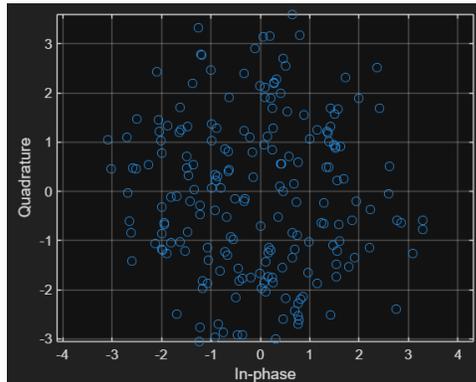
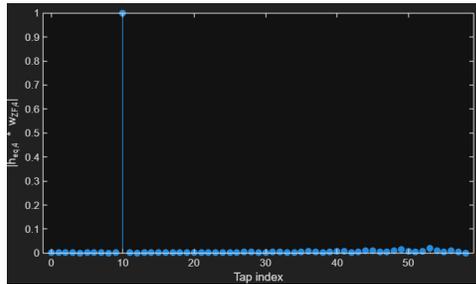
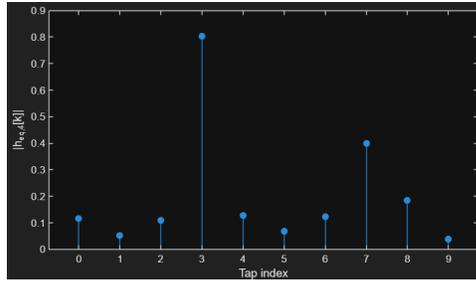
Φαίνεται ότι τα σύμβολα συγκεντρώνονται στα 4 σημεία του 4-QAM, άρα ο ισοσταθμιστής κάνει καλή διόρθωση στο κανάλι.

4. Στο τέταρτο μέρος, θα ασχοληθούμε με μη-ιδανικά κανάλια με CFO.

4.1 Να επαναλάβετε όλα τα βήματα του 3ου μέρους, μόνο που τώρα να υποθέσετε και CFO.

Η εκτέλεση του κώδικα, δίνει τα εξής αποτελέσματα:





Σε αυτό το ερώτημα, επαναλαμβάνουμε όλα τα βήματα του 3ου μέρους, απλά υποθέτουμε μη-ιδανικό κανάλι και CFO $F_s = 10^{-3}$. Το σήμα εξόδου πολλαπλασιάζεται με τον όρο $e^{j(2\pi\Delta F_s n + \phi)}$ και φιλτράρεται από το φίλτρο SRRC. Τα αποτελέσματα του κώδικα φαίνονται να είναι αρκετά καλά, το διάγραμμα αστερισμού στο τέλος όμως δείχνει τα σημεία διασκορπισμένα, αντί να συγκεντρώνονται στα 4 ιδανικά σημεία του 4-QAM. Αυτό το πρόβλημα, πιθανώς οφείλεται στην περιστροφή που εισάγει ο CFO, την οποία ο ισοσταθμιστής δυσκολεύεται να αντισταθμίσει.

Παρακάτω ακολουθεί ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση της άσκησης και βασίστηκε στον κώδικα που αναρτήθηκε στο eclass:

Listing 1: Κώδικας MATLAB

```

1 %
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % M file for testing synchronization for single-carrier frequency selective
  channels %
3 %     with CFO, using the energy of the output packet
  %
4 %
  %
5 %     This is the version to upload
  %
6 %
  %
7 % A. P. Liavas, Dec. 8, 2025
  %
8 %
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 clear; clc; close all;
11 rng(1);
12
13 T = 1;
14 over = 10;
15 Ts = T/over;
16 roll_off = 0.4;
17 B= 3;
18
19 % 1.1
20 % SRRC
21 g_T = srrc_pulse(T, Ts, B, roll_off);
22
23 % M     & training
24 N = 200; % #     (A_0,...,A_{N-1})
25 N_tr = 50; % #
26
27 % 4-QAM     2
28 sA2 = 2;
29 qam_norm = sqrt(sA2/2);
30
31 % Δ     A_n
32 A_seq = qam_norm * ((2*randi([0 1], N, 1)-1) + 1j*(2*randi([0 1], N, 1)-1));
33
34

```

```

35 train_data = A_seq(1:N_tr); % training symbols
36 data = A_seq(N_tr+1:end); % data symbols
37
38 x_upsampled = upsample(A_seq, over); % 0 (upsampling)
39
40 A_t = conv(x_upsampled, g_T); %
41
42 t=(0:length(A_t)-1).' * Ts;
43
44 %  $\Gamma$   $A(t)$ 
45 figure;
46 plot(t, real(A_t)); hold on;
47 plot(t, imag(A_t));
48 xlabel('t');
49 ylabel('Amplitude');
50 legend('Re\{A(t)\}', 'Im\{A(t)\}');
51
52 %  $\Gamma$ 
53 figure;
54 plot(real(A_seq), imag(A_seq), 'o');
55 axis equal; grid on;
56 xlabel('In-phase');
57 ylabel('Quadrature');
58
59 % 1.2
60
61
62 c_disc = 1; %  $c(t) = (t)$ 
63
64 Y_channel = conv(A_t, c_disc); %  $Y = A_t$ 
65
66 % Matched filter
67 g_R = g_T;
68
69 %  $E$  matched filter
70 Y_matched = conv(Y_channel, g_R) * Ts;
71
72 %  $\Sigma$ 
73 h_rc= conv(g_T, g_R) * Ts;
74
75 h0_index = 2*B*over + 1; %  $\Gamma = 0$ 
76
77 h0 = h_rc(h0_index); % gain
78
79 %  $\Delta$  ,  $h0\_index$ 
80 Y_sym = Y_matched(h0_index : over : h0_index + (N-1)*over);
81
82 %  $K$   $h0$ 
83 Y_hat = Y_sym/h0;
84
85 %  $\Delta$ 

```

```

86 figure;
87 plot(real(Y_hat), imag(Y_hat), 'o');
88 axis equal; grid on;
89 xlabel('In-phase');
90 ylabel('Quadrature');
91
92 % 1.3
93
94 %  $\Sigma$ 
95 Y_channel = A_t;
96
97 %
98 g_R = g_T;
99
100 % E matched filter
101 Y_mf = conv(Y_channel, g_R) * Ts;
102
103 % X ( 0)
104 t_mf = (0:length(Y_mf)-1).' * Ts;
105
106 %  $\Gamma$  MF
107 figure;
108 plot(t_mf, real(Y_mf)); hold on;
109 plot(t_mf, imag(Y_mf));
110 xlabel('t');
111 ylabel('Amplitude');
112 legend('Re\{y(t)\}', 'Im\{y(t)\}');
113
114 % 1.4
115
116 %  $c(t) = (t)$ 
117 c_disc = 1;
118
119 %  $\Sigma$  :  $g_T(t) * c(t) * g_R(t)$ 
120 h_tx_c = conv(g_T, c_disc);
121 h_composite = conv(h_tx_c, g_R) * Ts;
122
123 % X ( t=0)
124 t_h = (0:length(h_composite)-1).' * Ts;
125
126 %  $\Sigma$ 
127 figure;
128 plot(t_h, abs(h_composite));
129 xlabel('t');
130 ylabel('|h_{composite}(t)|');
131
132 % 1.5
133
134 % E
135 d_range = 0:(4*B*over - 1);
136 energy_d = zeros(size(d_range));

```

```

137
138 for idx=1:length(d_range)
139     d=d_range(idx);
140     y_candidate = Y_mf(d+1 : over : d+1 + (N-1)*over);
141     energy_d(idx) = sum(abs(y_candidate).^2);
142 end
143
144
145 %B
146 [~, idx_max] = max(energy_d);
147 d_hat = d_range(idx_max);
148
149 %  $\Sigma$ 
150 figure;
151 plot(d_range, energy_d);
152 xlabel('d');
153 ylabel('Energy metric');
154
155 %1.6
156
157 d_range =0:(4*B*over-1);
158 stat_tr=zeros(size(d_range));
159 abs_stat=zeros(size(d_range));
160
161 for idx= 1:length(d_range)
162     d=d_range(idx);
163     y_tr=Y_mf(d+1 : over : d+1 +(N_tr-1)*over);
164     stat_tr(idx)=train_data' * y_tr;
165     abs_stat(idx)=abs(stat_tr(idx));
166 end
167
168 % E training-based
169 [~, idx_max_tr] = max(abs_stat);
170 d_hat_tr = d_range(idx_max_tr);
171 disp(['Training-based delay estimate d_hat_tr = ' num2str(d_hat_tr)]);
172
173 % A
174 h_abs = abs(h_composite); % |h_composite|
175 h_abs_trunc = h_abs(1:length(d_range));
176
177 %K
178 abs_stat_norm = abs_stat / max(abs_stat);
179 h_abs_norm = h_abs_trunc / max(h_abs_trunc);
180
181 %  $\Sigma$ 
182 figure;
183 plot(d_range, abs_stat_norm); hold on;
184 plot(d_range, h_abs_norm);
185 xlabel('d');
186 ylabel('Normalized magnitude');
187 legend('|S(d)| from training', '|h_{composite}(t)|');

```

```

188
189 %% 1.7
190 if exist('d_hat_tr','var')
191     d_sync = d_hat_tr; %           training
192 else
193     d_sync = d_hat; %
194 end
195
196 % E           matched filter
197 Y_sym_sync = Y_mf(d_sync+1 : over : d_sync+1 + (N-1)*over);
198
199 % Δ           symbol-spaced
200 figure;
201 plot(real(Y_sym_norm), imag(Y_sym_norm), 'o');
202 axis equal; grid on;
203 xlabel('In-phase');
204 ylabel('Quadrature');
205
206 % 1.8
207 % 0            $c(t) = c(t)$ ,
208 %
209
210 %
211 %c=(randn + 1j*randn)/sqrt(2);
212
213 %c_disc2 = c; %  $c(t) = c(t)$ 
214 %Y_ch_c = conv(A_t, c_disc2); %
215
216 % 3) Matched filter
217 %g_R = g_T;
218 %Y_mf_c =conv(Y_ch_c, g_R) * Ts; %
219
220 %           MF (t=0)
221 %t_mf_c = (0:length(Y_mf_c)-1).' * Ts;
222
223 % Δ           matched filter
224 %figure;
225 %plot(t_mf_c, real(Y_mf_c)); hold on;
226 %plot(t_mf_c, imag(Y_mf_c));
227 %xlabel('t');
228 %ylabel('Amplitude');
229 %legend('Re\{y_c(t)\}', 'Im\{y_c(t)\}');
230
231 %g_T * c * g_R
232 %h_tx_c2=conv(g_T, c_disc2);
233 %h_composite_c =conv(h_tx_c2, g_R) * Ts;
234
235 %
236 %t_h_c=(0:length(h_composite_c)-1).' * Ts;
237
238 % Δ           h_composite_c(t)

```

```

239 %figure;
240 %plot(t_h_c, abs(h_composite_c));
241 %xlabel('t');
242 %ylabel('/h_{composite,c}(t)');
243
244 %Σ
245 %d_range_c = 0:(4*B*over - 1);
246 %energy_c = zeros(size(d_range_c));
247
248 %for idx=1:length(d_range_c)
249 %   d = d_range_c(idx);
250 %   y_cand=Y_mf_c(d+1 : over : d+1 + (N-1)*over);
251 %   energy_c(idx)=sum(abs(y_cand).^2);
252 %end
253
254 [%~, idx_max_c]=max(energy_c);
255 %d_hat_c=d_range_c(idx_max_c);
256 %disp(['Energy-based delay with scalar c: d_hat_c = ' num2str(d_hat_c)]);
257
258 % Δ      energy metric
259 %figure;
260 %plot(d_range_c, energy_c);
261 %xlabel('d');
262 %ylabel('Energy metric');
263
264 %Σ      training symbols
265 %stat_tr_c=zeros(size(d_range_c));
266 %abs_stat_c=zeros(size(d_range_c));
267
268 %for idx = 1:length(d_range_c)
269 %   d=d_range_c(idx);
270 %   y_tr_c=Y_mf_c(d+1 : over : d+1 + (N_tr-1)*over);
271 %   stat_tr_c(idx) =train_data' * y_tr_c;
272 %   abs_stat_c(idx) = abs(stat_tr_c(idx));
273 %end
274
275 [%~, idx_max_tr_c]=max(abs_stat_c);
276 %d_hat_tr_c=d_range_c(idx_max_tr_c);
277 %disp(['Training-based delay with scalar c: d_hat_tr_c = ' num2str(d_hat_tr_c)
278     ]);
279
280 %Σ      |S_c(d)|   |h_composite_c(t)|
281 %h_abs_c = abs(h_composite_c);
282 %h_abs_c_tr=h_abs_c(1:length(d_range_c));
283 %abs_stat_n_c = abs_stat_c/max(abs_stat_c);
284 %h_abs_n_c =h_abs_c_tr/max(h_abs_c_tr);
285
286 % Δ      training-based      |h_composite_c|
287 %figure;
288 %plot(d_range_c, abs_stat_n_c); hold on;
289 %plot(d_range_c, h_abs_n_c);

```

```

289 %xlabel('d');
290 %ylabel('Normalized magnitude');
291 %legend('|S_c(d)| from training', '|h_{composite,c}(t)|');
292
293 %Symbol-spaced
294 %d_sync_c=d_hat_tr_c;
295
296 %Y_sym_c=Y_mf_c(d_sync_c+1 : over : d_sync_c+1 + (N-1)*over);
297
298 %h0_index_c = 2*B*over + 1;
299 %h0_c = h_composite_c(h0_index_c);
300
301 %Y_sym_c_n = Y_sym_c / h0_c;
302
303 %                c(t)
304 %figure;
305 %plot(real(Y_sym_c_n), imag(Y_sym_c_n), 'o');
306 %axis equal; grid on;
307 %xlabel('In-phase');
308 %ylabel('Quadrature');
309
310 %% 2.1
311
312 %N      4-QAM
313 A_seq_CFO = qam_norm * ( (2*randi([0 1], N, 1)-1) + 1j*(2*randi([0 1], N, 1)
    -1) );
314
315 train_data_CFO = A_seq_CFO(1:N_tr); %training symbols
316 data_CFO = A_seq_CFO(N_tr+1:end); %data symbols
317
318 %A(t) = \sum A_n g_T(t-nT)
319 x_upsampled_CFO = upsample(A_seq_CFO, over);
320 A_t_CFO = conv(x_upsampled_CFO, g_T);
321
322 t_A_CFO = (0:length(A_t_CFO)-1).' * Ts; %      A(t)
323
324 %\Delta      A(t)
325 figure;
326 plot(t_A_CFO, real(A_t_CFO)); hold on;
327 plot(t_A_CFO, imag(A_t_CFO));
328 xlabel('t');
329 ylabel('Amplitude');
330 legend('Re\{A(t)\}', 'Im\{A(t)\}');
331
332 %      c(t) = (t)
333 c_disc_ideal=1;
334 Y_channel_CFO = conv(A_t_CFO, c_disc_ideal);
335
336 %Matched filter
337 g_R = g_T;
338 Y_mf_CFO = conv(Y_channel_CFO, g_R) * Ts;

```

```

339 t_mf_CFO=(0:length(Y_mf_CFO)-1).' * Ts;
340
341 %
342 h_tx_ideal_CFO = conv(g_T, c_disc_ideal);
343 h_composite_ideal_CFO=conv(h_tx_ideal_CFO, g_R) * Ts;
344
345 h0_index_CFO=2*B*over+1;
346 h0_ideal_CFO= h_composite_ideal_CFO(h0_index_CFO);
347 Y_sym_CFO=Y_mf_CFO(h0_index_CFO : over : h0_index_CFO + (N-1)*over);
348 Y_hat_CFO = Y_sym_CFO/h0_ideal_CFO;
349
350 % Δ
351 figure;
352 plot(real(Y_hat_CFO), imag(Y_hat_CFO), 'o');
353 axis equal; grid on;
354 xlabel('In-phase');
355 ylabel('Quadrature');
356
357 % 2.2
358
359 DeltaFs = 1e-3; % ΔF_s = ΔF * T_s
360 phi_CFO = 0; % = 0
361
362 n_samp = 0:length(A_t_CFO)-1;
363
364 % CFO
365 CFO_term_ts=exp(1j*(2*pi*DeltaFs*n_samp + phi_CFO)).';
366
367 % Y_n := Y(nT_s)=e^{jΔ(2F_s n + )} A(nT_s)
368 Y_n_CFO = CFO_term_ts .* A_t_CFO;
369
370 Y_rx_CFO_ts = Y_n_CFO;
371
372 % 2.3
373
374 g_R = g_T;
375
376 Y_mf_CFO2 = conv(Y_rx_CFO_ts, g_R) * Ts;
377 t_mf_CFO2 = (0:length(Y_mf_CFO2)-1).' * Ts;
378
379 figure;
380 plot(t_mf_CFO2, real(Y_mf_CFO2)); hold on;
381 plot(t_mf_CFO2, imag(Y_mf_CFO2));
382 xlabel('t');
383 ylabel('Amplitude');
384 legend('Re\{y_{CFO}(t)\}', 'Im\{y_{CFO}(t)\}');
385
386 %
387 c_disc_ideal = 1;
388 h_tx_CFOch = conv(g_T, c_disc_ideal);
389 h_composite_CFO = conv(h_tx_CFOch, g_R)*Ts;

```

```

390 t_h_CFO = (0:length(h_composite_CFO)-1).' * Ts;
391
392
393 figure;
394 plot(t_h_CFO, abs(h_composite_CFO));
395 xlabel('t');
396 ylabel('|h_{composite}(t)|');
397
398 %
399 d_range_CFO=0:(4*B*over - 1);
400 energy_CFO=zeros(size(d_range_CFO));
401
402 for idx = 1:length(d_range_CFO)
403     d=d_range_CFO(idx);
404     y_cand=Y_mf_CFO2(d+1 : over : d+1 + (N-1)*over);
405     energy_CFO(idx)=sum(abs(y_cand).^2);
406 end
407
408 [~, idx_max_CFO]=max(energy_CFO);
409 d_hat_CFO_energy=d_range_CFO(idx_max_CFO);
410 disp(['[CFO] Energy-based delay d\hat = ' num2str(d_hat_CFO_energy)]);
411
412 figure;
413 plot(d_range_CFO, energy_CFO);
414 xlabel('d');
415 ylabel('Energy metric');
416
417 %           training symbols
418 stat_tr_CFO=zeros(size(d_range_CFO));
419 abs_stat_CFO=zeros(size(d_range_CFO));
420
421 for idx= 1:length(d_range_CFO)
422     d=d_range_CFO(idx);
423     y_tr_CFO=Y_mf_CFO2(d+1 : over : d+1 + (N_tr-1)*over);
424     stat_tr_CFO(idx)=train_data_CFO' * y_tr_CFO;
425     abs_stat_CFO(idx) = abs(stat_tr_CFO(idx));
426 end
427
428 [~, idx_max_tr_CFO]=max(abs_stat_CFO);
429 d_hat_CFO_tr =d_range_CFO(idx_max_tr_CFO);
430 disp(['[CFO] Training-based delay d\hat\_tr = ' num2str(d_hat_CFO_tr)]);
431
432 %  $\Sigma$  |S(d)| |h_composite(t)|
433 h_abs_CFO = abs(h_composite_CFO);
434 h_abs_CFO_tr = h_abs_CFO(1:length(d_range_CFO));
435 abs_stat_CFO_norm = abs_stat_CFO / max(abs_stat_CFO);
436 h_abs_CFO_norm = h_abs_CFO_tr / max(h_abs_CFO_tr);
437
438 figure;
439 plot(d_range_CFO, abs_stat_CFO_norm); hold on;
440 plot(d_range_CFO, h_abs_CFO_norm);

```

```

441 xlabel('d');
442 ylabel('Normalized magnitude');
443 legend('|S_{CF0}(d)| from training','|h_{composite}(t)|');
444
445 d_sync_CF0 = d_hat_CF0_energy;
446
447 Y_sym_CF02 =Y_mf_CF02(d_sync_CF0+1 : over : d_sync_CF0+1+(N-1)*over);
448
449 h0_index_CF0 = 2*B*over + 1;
450 h0_CF0 = h_composite_CF0(h0_index_CF0);
451
452 Y_sym_CF02_n =Y_sym_CF02 / h0_CF0;
453
454 figure;
455 plot(real(Y_sym_CF02_n),imag(Y_sym_CF02_n), 'o');
456 axis equal; grid on;
457 xlabel('In-phase');
458 ylabel('Quadrature');
459
460 % 2.4
461
462 %  $\Phi \sim U[0, 2)$ 
463 DeltaFs=1e-3; %  $\Delta F_s = \Delta F * T_s$ 
464 phi_rand=2*pi*rand; %
465
466 n_samp =0:length(A_t_CF0)-1;
467 CF0_term_ts_rand = exp(1j * (2*pi*DeltaFs*n_samp + phi_rand)).'; %  $e^{j\Delta(2F_s n + \Phi)}$ 
468
469 %  $\Sigma$  CF0
470 Y_rx_CF0_ts_rand = CF0_term_ts_rand .* A_t_CF0;
471
472 % $c(t) = c(t)$ 
473 cCF0 = (randn + 1j*randn)/sqrt(2);
474 Y_ch_CF0c = cCF0 * Y_rx_CF0_ts_rand;
475
476 %Matched filter
477 g_R = g_T;
478 Y_mf_CF0c = conv(Y_ch_CF0c, g_R) * Ts;
479 t_mf_CF0c = (0:length(Y_mf_CF0c)-1).' * Ts;
480
481 figure;
482 plot(t_mf_CF0c, real(Y_mf_CF0c)); hold on;
483 plot(t_mf_CF0c, imag(Y_mf_CF0c));
484 xlabel('t');
485 ylabel('Amplitude');
486 legend('Re\{y_{CF0,c}(t)\}','Im\{y_{CF0,c}(t)\}');
487
488 %  $c$ 
489 h_tx_CF0c = conv(g_T, cCF0);
490 h_composite_CF0c = conv(h_tx_CF0c, g_R) * Ts;

```

```

491 t_h_CF0c = (0:length(h_composite_CF0c)-1).' * Ts;
492
493 figure;
494 plot(t_h_CF0c, abs(h_composite_CF0c));
495 xlabel('t');
496 ylabel('|h_{composite,c}(t)|');
497
498 %
499 d_range_CF0c = 0:(4*B*over - 1);
500 energy_CF0c=zeros(size(d_range_CF0c));
501
502 for idx=1:length(d_range_CF0c)
503     d=d_range_CF0c(idx);
504     y_cand=Y_mf_CF0c(d+1 : over : d+1 + (N-1)*over);
505     energy_CF0c(idx)=sum(abs(y_cand).^2);
506 end
507
508 [~, idx_max_CF0c]=max(energy_CF0c);
509 d_hat_CF0c_energy=d_range_CF0c(idx_max_CF0c);
510 disp(['[CF0+c] Energy-based delay d_hat = ' num2str(d_hat_CF0c_energy)]);
511
512 figure;
513 plot(d_range_CF0c, energy_CF0c);
514 xlabel('d');
515 ylabel('Energy metric');
516
517 %           training symbols
518 stat_tr_CF0c = zeros(size(d_range_CF0c));
519 abs_stat_CF0c = zeros(size(d_range_CF0c));
520
521 for idx = 1:length(d_range_CF0c)
522     d=d_range_CF0c(idx);
523     y_tr_CF0c = Y_mf_CF0c(d+1 : over : d+1+(N_tr-1)*over);
524     stat_tr_CF0c(idx) =train_data_CF0'*y_tr_CF0c;
525     abs_stat_CF0c(idx) = abs(stat_tr_CF0c(idx));
526 end
527
528
529 [~, idx_max_tr_CF0c] = max(abs_stat_CF0c);
530 d_hat_CF0c_tr = d_range_CF0c(idx_max_tr_CF0c);
531 disp(['[CF0+c] Training-based delay d_hat_tr = ' num2str(d_hat_CF0c_tr)]);
532
533 %           |S(d)|   |h_{composite,c}(t)|
534 h_abs_CF0c = abs(h_composite_CF0c);
535 h_abs_CF0c_tr =h_abs_CF0c(1:length(d_range_CF0c));
536
537 abs_stat_CF0c_norm = abs_stat_CF0c/max(abs_stat_CF0c);
538 h_abs_CF0c_norm = h_abs_CF0c_tr/max(h_abs_CF0c_tr);
539
540 figure;
541 plot(d_range_CF0c, abs_stat_CF0c_norm); hold on;

```

```

542 plot(d_range_CF0c, h_abs_CF0c_norm);
543 xlabel('d');
544 ylabel('Normalized magnitude');
545 legend('|S_{CF0,c}(d)| from training','|h_{composite,c}(t)|');
546
547 %symbol-spaced
548 d_sync_CF0c=d_hat_CF0c_energy;
549
550 Y_sym_CF0c =Y_mf_CF0c(d_sync_CF0c+1:over :d_sync_CF0c+1 +(N-1)*over);
551
552 h0_index_CF0c = 2*B*over + 1;
553 h0_CF0c = h_composite_CF0c(h0_index_CF0c);
554
555 Y_sym_CF0c_n = Y_sym_CF0c/h0_CF0c;
556
557 figure;
558 plot(real(Y_sym_CF0c_n), imag(Y_sym_CF0c_n), 'o');
559 axis equal; grid on;
560 xlabel('In-phase');
561 ylabel('Quadrature');
562
563 %2.5
564 M = 10; %
565
566 % training symbols :
567 Y_tr_sym_CF0 = Y_sym_CF02(1:N_tr);
568
569 % training matrix Toeplitz
570 train_matr_CF0=toeplitz( ...
571     train_data_CF0(M:N_tr), ...
572     train_data_CF0(M:-1:1).');
573
574 % B
575 B_proj =train_matr_CF0*( (train_matr_CF0' * train_matr_CF0)^(-1) )*
    train_matr_CF0';
576
577 % CFO
578 v_axis= -0.03:1e-5:0.03;
579
580 f_stat=zeros(size(v_axis));
581
582 for i_v=1:length(v_axis)
583     n_vec = (1:N_tr-M+1).';
584     exp_i_v =exp(1j * 2*pi*v_axis(i_v)*n_vec);
585     G_i_v=diag(exp_i_v);
586
587     %  $\sum f(v) (Im)$ 
588     f_stat(i_v) = Y_tr_sym_CF0(M:N_tr)' * G_i_v * B_proj * G_i_v' *
        Y_tr_sym_CF0(M:N_tr);
589 end
590

```

```

591 %       $Re\{f(v)\}$ 
592 figure;
593 plot(v_axis, real(f_stat));
594 xlabel('v');
595 ylabel('Re\{f(v)\}');
596
597 %      CFO
598 [~, idx_max_v]=max(real(f_stat));
599 v_est = v_axis(idx_max_v);
600
601 DeltaFs_true =DeltaFs;
602 DeltaFs_est = v_est/over;
603
604 disp(['True  $\Delta F_s$  = ' num2str(DeltaFs_true)]);
605 disp(['Estimated  $\Delta F_s$  ' num2str(DeltaFs_est)]);
606 disp(['Estimated  $v = \Delta F * T$  ' num2str(v_est)]);
607
608 %2.6
609
610 n_sym = (1:N).'; %
611
612 cfo_corr = exp(-1j*2*pi*v_est*n_sym); %  $e^{-j 2 v_{est} n}$ 
613
614 Y_sym_CF02_corr = Y_sym_CF02 .* cfo_corr; %      CFO      symbol-spaced
615
616 %      /
617 Z_sym=Y_sym_CF02_corr/h0_CF0;
618
619 % $\Delta$       CFO
620 figure;
621 plot(real(Z_sym), imag(Z_sym), 'o');
622 axis equal; grid on;
623 xlabel('In-phase');
624 ylabel('Quadrature');
625
626 % 2.7
627
628
629 A_tr=train_data_CF0; %      training symbols
630
631 Y_tr_corr=Y_sym_CF02_corr(1:N_tr); %  $\Lambda$       training symbols      timing sync
        & CFO correction
632
633 % $Y_{tr\_corr} = h_{eq} * A_{tr}$ 
634 h_LS =(A_tr' * Y_tr_corr)/(A_tr' * A_tr);
635
636 %  $\Theta$ 
637 h_eq_true = h0_CF0;
638
639 % E
640 disp(['h_LS = ' num2str(h_LS)]);

```

```

641 disp(['h_eq_true = ' num2str(h_eq_true)]);
642 disp(['|h_LS - h_eq_true| = ' num2str(abs(h_LS - h_eq_true))]);
643
644 % 2.8
645
646 w_eq = 1/h_LS; % 1 / h_LS
647
648 % Y_sym_CF02_corr
649 X_hat_eq=w_eq*Y_sym_CF02_corr;
650
651 % Δ
652 figure;
653 plot(real(X_hat_eq), imag(X_hat_eq), 'o');
654 axis equal; grid on;
655 xlabel('In-phase');
656 ylabel('Quadrature');
657
658 % 3.1
659
660 %  $4\text{-QAM}$ 
661 A_seq_3 = qam_norm*((2*randi([0 1], N, 1)-1)+1j*(2*randi([0 1], N, 1)-1) );
662
663 train_data_3 = A_seq_3(1:N_tr); % training symbols
664 data_3 = A_seq_3(N_tr+1:end); % data symbols
665
666 %  $SRRC: A_3(t) \Sigma = A_n g_T(t-nT)$ 
667 x_upsampled_3 = upsample(A_seq_3, over);
668 A_t_3 = conv(x_upsampled_3, g_T);
669 t_A_3 = (0:length(A_t_3)-1).' * Ts;
670
671 % Π  $A_3(t)$  ( )
672 figure;
673 plot(t_A_3, real(A_t_3)); hold on;
674 plot(t_A_3, imag(A_t_3));
675 xlabel('t');
676 ylabel('Amplitude');
677 legend('Re\{A_3(t)\}', 'Im\{A_3(t)\}');
678
679 % 3.2
680
681 %  $c(t) = c_0 * (t) + c_1 * (t - K * T_s)$ 
682
683 M = 10; %
684
685 K_3 = randi([0, (M-3)*over]); % K
686
687 c0=(randn+1j*randn)/sqrt(2); % c0
688 alpha=0.4;
689 c1=alpha*(randn+1j*randn)/sqrt(2); % c1
690
691 %  $c(t) = (t) (c_0, c_1)$ 

```

```

692 c_disc_3 = zeros(K_3+1,1);
693 c_disc_3(1) = c0;
694 c_disc_3(K_3+1)=c1;
695
696 %A_3(t)
697 Y_ch_3 =conv(A_t_3, c_disc_3);
698 t_Y_3=(0:length(Y_ch_3)-1).' * Ts;
699
700
701 % 3.3
702
703 %Matched filter
704 g_R_3 = g_T;
705
706 Y_mf_3=conv(Y_ch_3, g_R_3) * Ts; %
707 t_mf_3=(0:length(Y_mf_3)-1).' * Ts; %           t=0
708
709 figure;
710 plot(t_mf_3, real(Y_mf_3)); hold on;
711 plot(t_mf_3, imag(Y_mf_3));
712 xlabel('t');
713 ylabel('Amplitude');
714 legend('Re\{y_3(t)\}','Im\{y_3(t)\}');
715
716 %           -> g_T * c(t) * g_R
717 h_tx_3 = conv(g_T, c_disc_3); % g_T*c(t)
718 h_comp_3 = conv(h_tx_3, g_R_3)*Ts; % (g_T*c*g_R)*Ts
719 t_h3 = (0:length(h_comp_3)-1).' * Ts;
720
721 figure;
722 plot(t_h3, abs(h_comp_3));
723 xlabel('t');
724 ylabel('|h_{composite,3}(t)|');
725
726 %3.4
727
728 A_half = B; %           SRRC
729 L_mf=length(Y_mf_3);
730
731 %           d
732 d_max_theory = 4*A_half*over + K_3 + 1;
733
734 %\Gamma           N_tr training samples           Y_mf_3
735 %           composite
736 d_max_corr=min(d_max_theory, L_mf-1-(N_tr-1)*over);
737 d_max_corr=min(d_max_corr, length(h_comp_3)-1-A_half*over);
738
739 d_range_corr=0:d_max_corr;
740
741 corr_d=zeros(length(d_range_corr),1);
742

```

```

743 for ii=1:length(d_range_corr)
744     d=d_range_corr(ii);
745     y_tr_d = Y_mf_3(d+1 : over : d+1 + (N_tr-1)*over);
746     corr_d(ii) = sum( conj(train_data_3) .* y_tr_d );
747 end
748
749 %
750 corr_abs_norm = abs(corr_d)/max(abs(corr_d));
751
752 %  $d$ 
753 idx_h = A_half*over+d_range_corr+1;
754 h_sub=h_comp_3(idx_h);
755 h_sub_norm=abs(h_sub)/max(abs(h_sub));
756
757 %  $\Sigma$   $|corr_d|$   $|h_{composite,3}|$ 
758 figure;
759 plot(d_range_corr, corr_abs_norm); hold on;
760 plot(d_range_corr, h_sub_norm);
761 xlabel('d');
762 ylabel('Normalized magnitude');
763 legend('|corr_d|', '|h_{composite,3}|');
764
765 % 3.5
766
767 M_eq = M; %
768 L_sym = N+M_eq-1; % symbol-spaced
769
770 L_mf = length(Y_mf_3);
771
772 %  $d$   $L_{sym}$ 
773 d_max_E = L_mf - 1 - (L_sym-1)*over;
774 d_range_E = 0:d_max_E;
775
776 E_d_3 = zeros(length(d_range_E),1);
777
778 for ii = 1:length(d_range_E)
779     d=d_range_E(ii);
780     y_cand=Y_mf_3(d+1:over:d+1+(L_sym-1)*over);
781     E_d_3(ii)=sum(abs(y_cand).^2);
782 end
783
784 %  $d_{hat}$   $E_d$ 
785 [~, idx_max_E] = max(E_d_3);
786 d_hat_E =d_range_E(idx_max_E);
787 disp(['d_hat_energy = ' num2str(d_hat_E)]);
788
789 %  $\Sigma$ 
790 figure;
791 plot(d_range_E, E_d_3);
792 xlabel('d');
793 ylabel('Energy metric');

```

```

794
795 % Symbol-spaced  $N+M-1$ 
796 Y_sym_3_E = Y_mf_3(d_hat_E+1:over:d_hat_E+1+(L_sym-1)*over);
797 disp(['length(Y_sym_3_E) = ' num2str(length(Y_sym_3_E))]);
798
799 % 3.6
800 M_eq=M;%
801
802 % X  $symbol-spaced$ 
803 %  $N_{tr}$  training symbols
804 Y_tr_3=Y_sym_3_E(1:N_tr);
805
806 Y_tr_eff=Y_tr_3(M_eq:N_tr);
807
808
809 X_tr_3 = toeplitz( ...
810     train_data_3(M_eq:N_tr), ...
811     train_data_3(M_eq:-1:1).' );
812
813 % LS  $h_{eq}$  ( $M_{eq}$ )
814 h_eq_LS_3=(X_tr_3' * X_tr_3)\(X_tr_3' * Y_tr_eff);
815
816 figure;
817 stem(0:M_eq-1, abs(h_eq_LS_3), 'filled');
818 xlabel('Tap index');
819 ylabel('|h_{eq}[k]|');
820
821 %3.7
822 M_eq = M; %
823 L_ZF=5*M_eq; % ZF M 5
824 delta_ZF=M_eq; % = M
825
826 h_eq=h_eq_LS_3(:); %
827
828 % K  $H_{ZF}$   $h_{eq} * w_{ZF}$ 
829 h_col=[h_eq; zeros(L_ZF-1,1)];
830 h_row=[h_eq(1); zeros(L_ZF-1,1)];
831 H_ZF=toeplitz(h_col, h_row);
832
833 P_len = M_eq + L_ZF - 1;
834 e_delta = zeros(P_len,1);
835 e_delta(delta_ZF + 1)=1;
836
837 % ZF
838 w_ZF=H_ZF\e_delta;
839
840 %
841 g_eff = conv(h_eq, w_ZF);
842 k_g = 0:length(g_eff)-1;
843
844 figure;

```

```

845 stem(k_g, abs(g_eff), 'filled');
846 xlabel('Tap index');
847 ylabel('|h_{eq} * w_{ZF}|');
848
849 %3.8
850 %
851 Y_eq_3 = conv(Y_sym_3_E, w_ZF);
852
853 %
854 Y_eq_3_cut=Y_eq_3(delta_ZF+1:end);
855
856 %          N
857 X_hat_3=Y_eq_3_cut(1:N);
858
859 % Δ          N
860 figure;
861 plot(real(X_hat_3), imag(X_hat_3), 'o');
862 axis equal; grid on;
863 xlabel('In-phase');
864 ylabel('Quadrature');
865
866 %4.1
867
868 DeltaF_s_4 = 1e-3;
869 phi_4=0;
870
871 n_ch_4=(0:length(Y_ch_3)-1).';
872 CFO_term_4=exp(1j*(2*pi*DeltaF_s_4*n_ch_4+phi_4));
873
874 % Σ          CFO
875 Y_ch_4 =Y_ch_3 .* CFO_term_4;
876
877 % Matched filter
878 g_R_4=g_T;
879 Y_mf_4=conv(Y_ch_4, g_R_4) * Ts;
880 t_mf_4=(0:length(Y_mf_4)-1).'* Ts;
881
882 figure;
883 plot(t_mf_4, real(Y_mf_4)); hold on;
884 plot(t_mf_4, imag(Y_mf_4));
885 xlabel('t');
886 ylabel('Amplitude');
887 legend('Re\{y_4(t)\}', 'Im\{y_4(t)\}');
888
889 h_comp_4=h_comp_3;
890 t_h4=t_h3;
891
892 figure;
893 plot(t_h4, abs(h_comp_4));
894 xlabel('t');
895 ylabel('|h_{composite,4}(t)|');

```

```

896
897 %4.2
898
899 A_half = B;
900 L_mf_4 = length(Y_mf_4);
901 d_max_theo4 = 4*A_half*over+K_3+1;
902
903 d_max_corr4=min(d_max_theo4, L_mf_4-1-(N_tr-1)*over);
904 d_max_corr4=min(d_max_corr4, length(h_comp_4)-1- A_half*over);
905
906 d_range_corr4 =0:d_max_corr4;
907
908 corr_d_4=zeros(length(d_range_corr4),1);
909
910 for ii=1:length(d_range_corr4)
911     d=d_range_corr4(ii);
912     y_tr_d_4=Y_mf_4(d+1 : over : d+1 + (N_tr-1)*over);
913     corr_d_4(ii)=sum( conj(train_data_3) .* y_tr_d_4 );
914 end
915
916 corr_abs_norm_4 =abs(corr_d_4) / max(abs(corr_d_4));
917
918 idx_h_4=A_half*over + d_range_corr4 + 1;
919 h_sub_4 =h_comp_4(idx_h_4);
920 h_sub_norm_4=abs(h_sub_4) / max(abs(h_sub_4));
921
922 figure;
923 plot(d_range_corr4, corr_abs_norm_4); hold on;
924 plot(d_range_corr4, h_sub_norm_4);
925 xlabel('d');
926 ylabel('Normalized magnitude');
927 legend('|corr_d| (CF0)', '|h_{composite,4}|');
928
929 %4.3
930
931 M_eq_4 = M;
932 L_sym_4 = N+M_eq_4-1;
933 L_mf_4 = length(Y_mf_4);
934
935 d_max_E4=L_mf_4-1-(L_sym_4-1)*over;
936 d_range_E4=0:d_max_E4;
937
938 E_d_4 = zeros(length(d_range_E4),1);
939
940 for ii = 1:length(d_range_E4)
941     d=d_range_E4(ii);
942     y_cand_4=Y_mf_4(d+1 : over : d+1+(L_sym_4-1)*over);
943     E_d_4(ii)=sum(abs(y_cand_4).^2);
944 end
945
946 [~, idx_max_E4]=max(E_d_4);

```

```

947 d_hat_E4=d_range_E4(idx_max_E4);
948 disp(['d_hat_energy = ' num2str(d_hat_E4)]);
949
950 figure;
951 plot(d_range_E4, E_d_4);
952 xlabel('d');
953 ylabel('Energy metric');
954
955 % symbol-spaced  $N+M-1$ 
956 Y_sym_4_E = Y_mf_4(d_hat_E4+1 : over : d_hat_E4+1 + (L_sym_4-1)*over);
957 disp(['length(Y_sym_4_E) = ' num2str(length(Y_sym_4_E))]);
958
959 % --- 4.4 / LS ( 3.6) ---
960
961 % Training symbol-spaced
962 Y_tr_4 = Y_sym_4_E(1:N_tr);
963 Y_tr_eff4 = Y_tr_4(M_eq_4:N_tr);
964
965 X_tr_4 =toeplitz( ...
966     train_data_3(M_eq_4:N_tr), ...
967     train_data_3(M_eq_4:-1:1).' );
968
969 h_eq_LS_4=(X_tr_4' * X_tr_4) \ (X_tr_4' * Y_tr_eff4);
970
971 figure;
972 stem(0:M_eq_4-1, abs(h_eq_LS_4), 'filled');
973 xlabel('Tap index');
974 ylabel('|h_{eq,4}[k]|');
975
976 %4.5
977
978 L_ZF_4 = 5*M_eq_4;
979 delta_ZF4 =M_eq_4;
980
981 h_eq_4=h_eq_LS_4(:);
982
983 h_col_4=[h_eq_4; zeros(L_ZF_4-1,1)];
984 h_row_4=[h_eq_4(1); zeros(L_ZF_4-1,1)];
985 H_ZF_4 =toeplitz(h_col_4, h_row_4);
986
987 P_len_4=M_eq_4+L_ZF_4-1;
988 e_delta_4=zeros(P_len_4,1);
989 e_delta_4(delta_ZF4+1)=1;
990
991 w_ZF_4=H_ZF_4\e_delta_4;
992
993 g_eff_4=conv(h_eq_4, w_ZF_4);
994 k_g_4=0:length(g_eff_4)-1;
995
996 figure;
997 stem(k_g_4, abs(g_eff_4), 'filled');

```

```

998 xlabel('Tap index');
999 ylabel('|h_{eq,4} * w_{ZF,4}|');
1000
1001 %4.6
1002
1003 Y_eq_4=conv(Y_sym_4_E, w_ZF_4);
1004
1005 Y_eq_4_cut=Y_eq_4(delta_ZF4+1:end);
1006 X_hat_4 =Y_eq_4_cut(1:N);
1007
1008
1009 figure;
1010 plot(real(X_hat_4), imag(X_hat_4), 'o');
1011 axis equal; grid on;
1012 xlabel('In-phase');
1013 ylabel('Quadrature');
1014
1015
1016
1017
1018
1019 function [phi, t] = srrc_pulse(T, Ts, A, a)
1020 %
1021 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1022 % phi = srrc_pulse(T, Ts, A, a)
1023 %
1024 % OUTPUT
1025 %
1026 % phi: truncated SRRC pulse, with parameter T,
1027 % roll-off factor a, and duration 2*A*T
1028 % t: time axis of the truncated pulse
1029 % INPUT
1030 %
1031 % T: Nyquist parameter or symbol period (real number)
1032 % Ts: sampling period (Ts=T/over)
1033 % where over is a positive INTEGER called oversampling factor
1034 % A: half duration of the pulse in symbol periods (positive INTEGER)
1035 % a: roll-off factor (real number between 0 and 1)
1036 %
1037 %
1038 % A. P. Liavas, Nov. 2011

```

```

1034 %
1035 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1036 t = [-A*T:Ts:A*T] + 10-8; % in order to avoid division by zero problems at
1037 t=0.
1038 if (a>0 && a<=1)
1039     num = cos((1+a)*pi*t/T) + sin((1-a)*pi*t/T) ./ (4*a*t/T);
1040     denom = 1-(4*a*t./T).^2;
1041     phi = 4*a/(pi*sqrt(T)) * num ./ denom;
1042 elseif (a==0)
1043     phi = 1/(sqrt(T)) * sin(pi*t/T)./(pi*t/T);
1044 else
1045     phi = zeros(length(t),1);
1046     disp('Illegal value of roll-off factor')
1047     return
1048 end
1049 end

```