Ψηφιακές Τηλεπικοινωνίες

1ο σετ Εργαστηριακών Ασκήσεων

Λουδάρος Ιωάννης (1067400)

Μπορείτε να δείτε την τελευταία έκδοση του Project εδώ ή σκανάροντας τον κωδικό QR που βρίσκεται στην επικεφαλίδα.

Περιγραφή Αναφοράς

Παρακάτω παραθέτω τις απαντήσεις μου στην "1ο σετ Εργαστηριακών Ασκήσεων" του μαθήματος "Ψηφιακές Τηλεπικοινωνίες" καθώς και σχόλια τα οποία προέκυψαν κατά την εκπόνηση του.

Σύντομη Παρουσίαση του σετ

Για την διευκόλυνση σας, έχει προετοιμαστεί ένα Live Script αρχείο που παρουσιάζει όλα τα ερωτήματα, μαζί με τους υπολογισμούς που χρειάστηκαν. Για την εγκατάσταση του ακολουθήστε τα παρακάτω βήματα.

1. Πατήστε το κουμπί "Live Script" που βρίσκεται παρακάτω.
2. Θα οδηγηθείτε στα releases του Github Repository που χρησιμοποιείται για αυτό το σετ.
3. Κατεβάστε την τελευταία έκδοση του "`submission.zip`".
4. Ανοίξτε το αρχείο `Set1.mlx`.
5. Πατήστε "Run".

Live Script

Σε περίπτωση που θέλετε απλά να δείτε το περιεχόμενο, δίχως να αλληλεπιδράσετε με το Live Script, μπορείτε να ανοίξετε το αρχείο scripts_pdf/Set1.pdf.
Αντίστοιχα αρχεία pdf υπάρχουν για κάθε συνάρτηση η οποία αναπτύχθηκε στα πλαίσια της άσκησης.

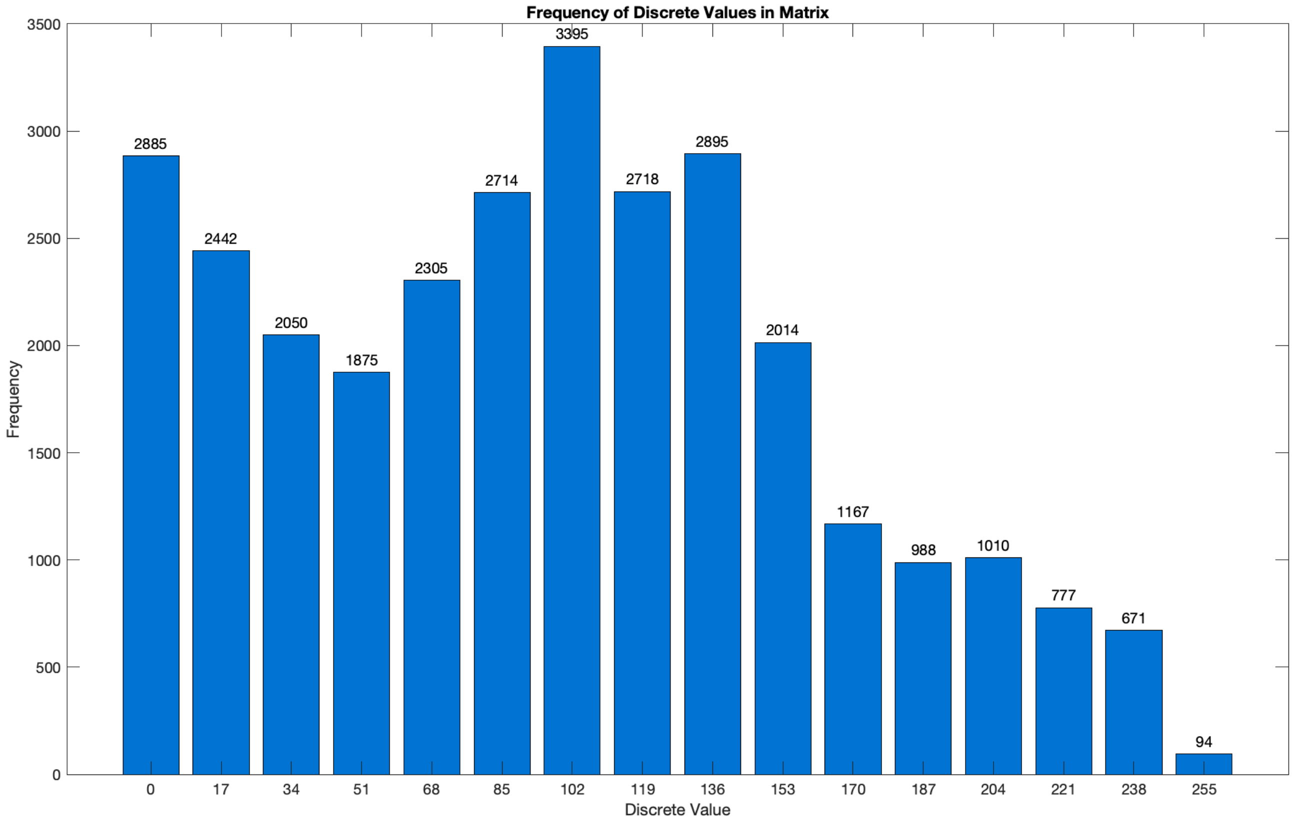
Περιεχόμενα

Απαντήσεις

Μέρος Α

1.a. Συχνότητα Εμφάνισης τιμών στο Μητρώο

Αφού διαπεράσαμε το μητρώο, εξάγαμε τις εξής συχνότητες εμφάνισης.



1.b. Κωδικοποίηση Huffman

Symbol	Encoding	Probability
0	110	9.6%
17	0010	8.1%
34	0100	6.8%
51	0111	6.3%
68	0011	7.7%
85	0000	9.0%
102	100	11.3%
119	111	9.1%
136	101	9.7%
153	0101	6.7%
170	00011	3.9%
187	01101	3.3%
204	01100	3.4%
221	000100	2.6%
238	0001010	2.2%
255	0001011	0.3%

Χρησιμοποιώντας την συνάρτηση `huffmandict`, παίρνουμε την κωδικοποίηση huffman για κάθε σύμβολο που περιέχει το μητρώο μας. Ύστερα, τροφοδοτούμε με αυτές τις κωδικοποιήσεις τη συνάρτηση `huffmanenco` και το σήμα της πηγή μας κωδικοποιείται επιτυχώς. Αριστερά μπορείτε να δείτε την κωδικοποίηση των συμβόλων.

Αξίζει να σημειωθεί ότι δεν κωδικοποιήσαμε ολόκληρο το εύρος τιμών που μπορεί να πάρει η φωτεινότητα ενός pixel, αλλά μόνο τις τιμές που όντως παίρνουν τα πίξελ στην εικόνα μας, συμβουλευόμενοι το προηγούμενο ερώτημα. Κάτι τέτοιο προφανώς δεν θα ήταν εφικτό αν δεν είχαμε την εικόνα γνωστή από πριν.

Εντροπία Κωδικοποίησης : 3.7831

Μέσο Μήκος Κώδικα : 3.8374

Αποδοτικότητα Κώδικα : 0.9859

2.a. Σύμβολα Δεύτερης Τάξης Επέκτασης Πηγής

Όπως καταλαβαίνουμε, η Δεύτερης Τάξης Επέκτασης Πηγή, θα έχει όλες τις δυνατές μεταθέσεις των συμβόλων του προηγούμενου ερωτήματος. Για ευνόητους λόγους δεν παραθέτουμε όλες τις 256 μεταθέσεις, αλλά αν το ποθείτε, είναι διαθέσιμες να τις θαυμάσετε μέσα στο 256x2 matrix με όνομα `new_symbols`. Οι αντίστοιχες πιθανότητες εμφάνισης βρίσκονται στο array `new_symbol_prob`.

Μπορούμε να υποθέσουμε από τώρα ότι οι πιθανότητες εμφάνισης που υπολογίσαμε, θα εμφανίζονταν μονάχα αν είχαμε μια πηγή η οποία γεννάει πολλές ασπρόμαυρες εικόνες (που επιπλέον χρωματίζονται μόνο από τις 16 φωτεινότητες στις οποίες έχει συμπιεστεί η εικόνα μας). Τώρα που έχουμε μονάχα αυτή τη συγκεκριμένη εικόνα, είναι τουλάχιστον αμφίβολο το αν τελικά θα εμφανιστούν και τα 256 σύνθετα σύμβολα μας.

2.b. Κωδικοποίηση Φωτογραφίας

Για να πραγματοποιηθεί αυτό το ερώτημα πρέπει να εντοπίσουμε ποια ζευγάρια εντοπίζονται όντως μέσα στην φωτογραφία. Στο 178x2 μητρώο `unique_pairs` μπορείτε να δείτε τα 178 ζεύγη που όντως συναντάμε στην εικόνα μας. Αντίστοιχα στο `unique_pair_frequency` μπορείτε να βρείτε τις αντίστοιχες συχνότητες, από τις οποίες υπολογίζουμε τις πιθανότητες εμφάνισης στο `unique_pair_ptobability`.

Πριν κωδικοποιήσουμε την φωτογραφία μας, μετατρέπουμε τα ζευγάρια σε καινούργια σύμβολα ώστε να διευκολύνουμε την διαδικασία. Μπορείτε να δείτε την εικόνα μετατρεμμένη σε σύμβολα αντρέχοντας στο `image_stream_converted`. Να σημειωθεί ότι το όνομα κάθε συμβόλου είναι ο αριθμός της σειράς στην οποία εντοπίζεται το αντίστοιχο ζευγάρι στο `unique_pairs`.

Εντροπία, Μέσο Μήκος Κώδικα και Αποδοτικότητα Κώδικα για τις δύο Πηγές

	Εντροπία της κωδικοποίησης	Μέσο Μήκος Κώδικα	Αποδοτικότητα
Πηγή ερωτήματος 1	3.7831	3.8374	98.59%
Πηγή ερωτήματος 2	5.6147	5.6412	99.53%

2.c. Σύγκριση με Προηγούμενο Ερώτημα

Όπως βλέπουμε στον προηγούμενο πίνακα, η εντροπία της κωδικοποίησης μας και η αποδοτικότητα της αυξήθηκαν. Αυτό όμως, έγινε με ένα κόστος. Το μέσο μήκος του κώδικα μας (και ακόμη περισσότερο το πλήθος των λέξεων του) αυξήθηκαν.

3.a. Υπολογισμός Εντροπίας

Όπως θίξαμε ήδη από την δεύτερη παράγραφο του ερωτήματος 2.a, η διαδικασία που ακολουθήσαμε δεν δημιούργησε ακριβώς την δεύτερης τάξης επέκταση πηγής του ερωτήματος 1, αλλά μια μικρότερη που περιέχει ένα υποσύνολο των συμβόλων της. Έτσι από τα 256 που θα έπρεπε να είναι κανονικά τα σύμβολα, συναντήσαμε μόνο 178.

3.b. Φράγμα Μέσου Μήκους Κώδικα

Το φράγμα για το μέσο μήκος όπως διατυπώνεται στις σημειώσεις του μαθήματος ισχύει και στις δύο πηγές.

$$H(X) \leq \bar{L} < H(X) + 1$$

Δηλαδή :

- $3.7831 \leq 3.8374 < 4.7831$, που είναι αληθές, και
- $5.6147 \leq 5.6412 < 6.6147$, που είναι επίσης αληθές.

4. Επαλήθευση Κωδικοποίησης και Λόγος Συμπίεσης

Χρησιμοποιούμε την huffmandeco για να δούμε αν θα καταφέρουμε να αποκωδικοποιήσουμε το σήμα μας και να το επιστρέψουμε στην αρχική του μορφή. Πράγματι, συγκρίνωντας το αποκωδικοποίημενο, με το αρχικό stream, βλέπουμε ότι είναι ίδια.

Μπορούμε επίσης να δούμε ότι το compression rate είναι **95.93%**

Αξίζει εδώ να σημειώσουμε ότι η εκφώνηση μας προτείνει να χρησιμοποιήσουμε uint8 (και άρα να σπαταλίσουμε επιπλέον 4 bit που δεν χρειάζονται στην αναπαράσταση κάθε pixel). Στην δική μας υλοποίηση όμως διακρίνουμε ότι μπορούμε να χρησιμοποιήσουμε μόλις 4, αφού η φωτεινότητα παίρνει μόλις 16 διακριτές τιμές. Παρόλα αυτά, αν το αποτέλεσμα πρέπει να δωθεί με προϋπόθεση της χρήσης uint8, το compression rate γίνεται **47.97%.**

5. Μετάδοση Σήματος μέσα από Κανάλι

Χρησιμοποιούμε τη δοσμένη συνάρτηση και παίρνουμε το σήμα `received`. Διατρέχοντας μια φορά το `encoded_image_stream` (δηλαδή αυτό που στείλαμε) και το received, μπορούμε εύκολα να εξάγουμε τα $p(x_j)$, $p(y_k)$, $p(x_j, y_k)$ καθώς και την παράμετρο p. Έτσι έχουμε όσα χρειαζόμαστε για να καταλήξουμε στα παρακάτω αποτελέσματα:

Παράμετρος p, Χωρητικότητα Καναλιού, Αμοιβαία Πληροφορία

Ζητούμενο	Αποτέλεσμα
Παράμετρος p	p = 0.8818
Χωρητικότητα Καναλιού	C = 0.4758
Εντροπία	H = 0.5242
Αμοιβαία Πληροφορία	MI = 0.3705

Μέρος Β

Ερώτημα 1

Βοηθητικές Συναρτήσεις

Για τον κβαντιστή δημιουργήθηκε η συνάρτηση που βλέπεται στο πράσινο πλαίσιο που ακολουθεί.
Μπορείτε να δείτε την αναλυτική της υλοποίηση στο `functions/iquantizer.mlx` που συνοδεύει την αναφορά.

```
[quantized, centers] = iquantizer(y, N, min_value, max_value)
```

Είσοδοι

`y` : Το τρέχον δείγμα του σφάλματος πρόβλεψης ως είσοδος του κβαντιστή.

`N` : Ο αριθμός των bits/sample που θα χρησιμοποιηθούν.

`min_value` : Η ελάχιστη αποδεκτή τιμή του σφάλματος πρόβλεψης.

`max_value` : Η μέγιστη αποδεκτή τιμή του σφάλματος πρόβλεψης.

Έξοδοι

`quantized` : Το κβαντισμένο δείγμα του τρέχοντος δείγματος του σφάλματος πρόβλεψης.

`centers` : Διάνυσμα με τα κέντρα των περιοχών κβάντισης

Για την εξαγωγή των στοχαστικών ποσοτήτων δημιουργήθηκε η συνάρτηση που βλέπεται στο πράσινο πλαίσιο που ακολουθεί.
Μπορείτε να δείτε την αναλυτική της υλοποίηση στο `functions/Rx.mlx` που συνοδεύει την αναφορά.

```
[R, r] = Rx(p, x)
```

Είσοδοι

`p` : Το πλήθος των παρελθοντικών τιμών του δείγματος που χρησιμοποιούνται στην πρόβλεψη.

`x` : Το προς κωδικοποίηση σήμα.

Έξοδοι

`R` : Πίνακας αυτοσυσχέτισης διάστασης $\boldsymbol{pxp}$.

`r` : Διάνυσμα αυτοσυσχέτισης διάστασης $\boldsymbol{px1}$.

Για την εξαγωγή της πρόβλεψης δημιουργήθηκε η συνάρτηση που βλέπεται στο πράσινο πλαίσιο που ακολουθεί.
Μπορείτε να δείτε την αναλυτική της υλοποίηση στο `functions/ipredictor.mlx` που συνοδεύει την αναφορά.

```
prediction = ipredictor(a, memory)
```

Είσοδοι

`a` : Οι συντελεστές που θα χρησιμοποιηθούν για την πρόβλεψη.

`memory` : Οι προηγούμενες τιμές του σήματος που θα χρησιμοποιηθούν για την πρόβλεψη.

Έξοδοι

`prediction` : Η πρόβλεψη.

Οι παραπάνω συναρτήσεις τελικά χρησιμοποιήθηκαν για να συνθέσουν τις συναρτήσεις για την κωδικοποίηση και την αποκωδικοποίηση που φαίνονται στην συνέχεια.

Κωδικοποίηση

Παρατίθεται η συνάρτηση που αναλαμβάνει την κωδικοποίηση του σήματος σε σειρά κβαντισμένων σφαλμάτων.

Μπορείτε να δείτε την αναλυτική της υλοποίηση στο `functions/idpcm_enco.mlx` που συνοδεύει την αναφορά.

```
[y_error_quantised, centers, a_quantised]  = idpcm_enco(x, p, N, min_value, max_value)
```

Είσοδοι

`x` : Το προς κωδικοποίηση σήμα.

`p` : Το πλήθος των παρελθοντικών τιμών του δείγματος που χρησιμοποιούνται στην πρόβλεψη.

`N` : Ο αριθμός των bits/sample που θα χρησιμοποιηθούν.

`min_value` και `max_value` : Οι τιμές που θα περαστούν στον κβαντιστή μέσα στον κωδικοποιητή.

Έξοδοι

`y_error_quantised` : Το κβαντισμένο σφάλμα που θα σταλεί.

`centers` : Τα κέντρα που παρήγαγε ο κβαντιστής.

`a_quantised` : Οι συντελεστές που παρήγαγε η Rx του κωδικοποιητή.

Αποκωδικοποίηση

Παρατίθεται η συνάρτηση που αναλαμβάνει την ανακατασκευή του σήματος από την σειρά κβαντισμένων σφαλμάτων.

Μπορείτε να δείτε την αναλυτική της υλοποίηση στο `functions/idpcm_deco.mlx` που συνοδεύει την αναφορά.

```
reconstructed = idpcm_deco(encoded, a, centers)
```

Είσοδοι

`encoded` : Το προς αποκωδικοποίηση σήμα.

a : Οι συντελεστές που παρήγαγε η Rx του κωδικοποιητή.

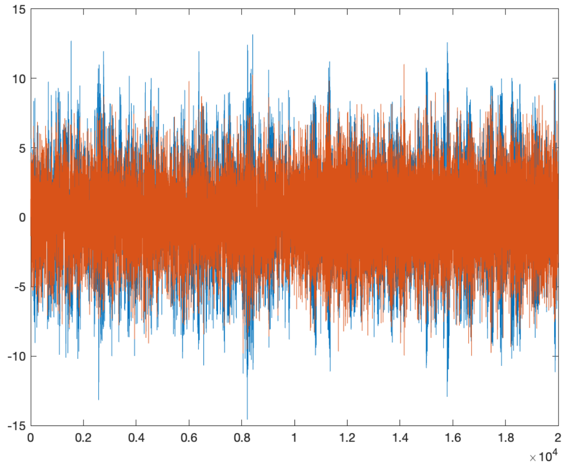
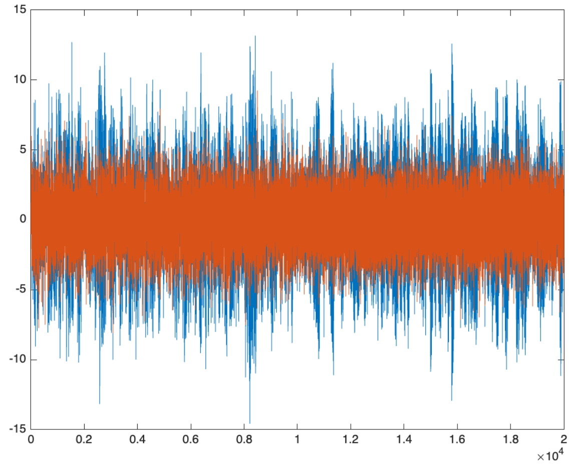
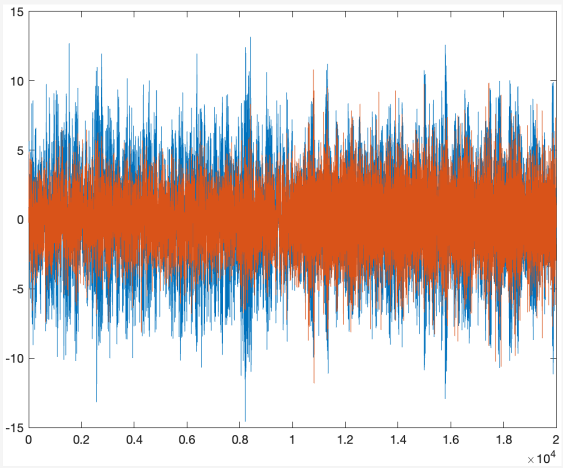
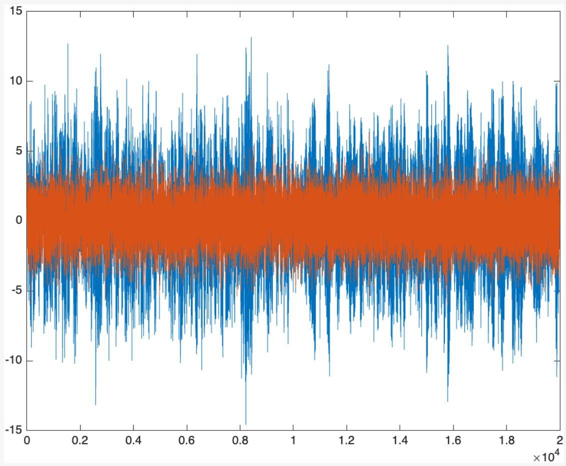
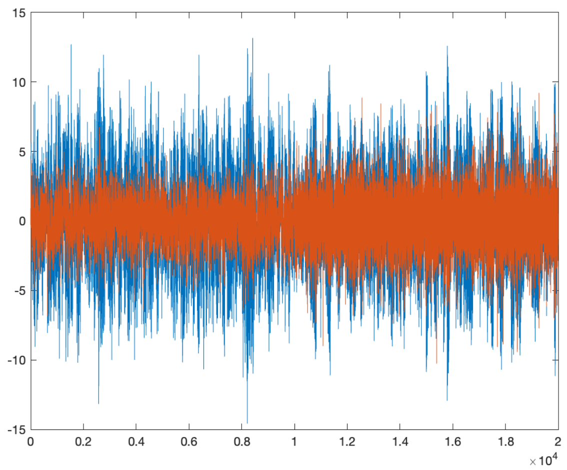
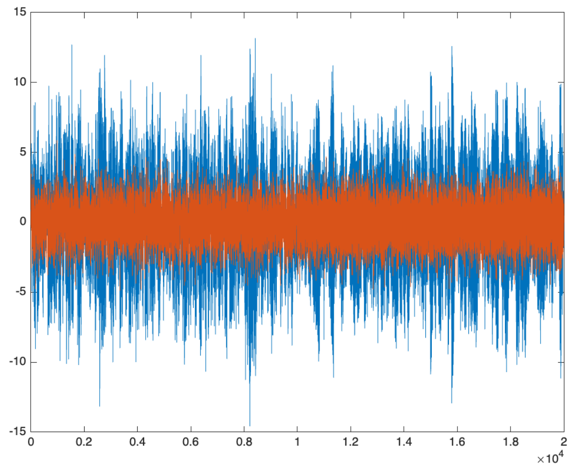
`centers` : Τα κέντρα που παρήγαγε ο κβαντιστής.

Έξοδοι

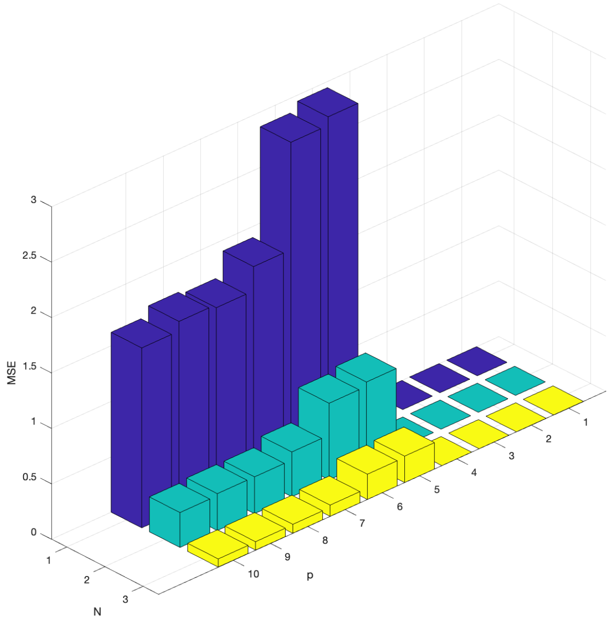
`reconstructed` : Το ανακατασκευασμένο σήμα.

Ερώτημα 2

Μας γίνεται ξεκάθαρο ότι όσο μεγαλύτερο το p, τόσο καλύτερη η πρόβλεψη και όσο μεγαλύτερο το N, τόσο μικρότερο σφάλμα εισάγει η κβάντιση. Και οι δύο παράμετροι, όσο ανεβαίνουν, μειώνουν το σφάλμα y.

	p=5	**p=10**
N=1		
N=2		
N=3		

Ερώτημα 3



Στο γράφημα αριστερά μπορείτε να δείτε πως μεταβάλλεται το μέσο τετραγωνικό σφάλμα σε σχέση με τις παραμέτρους N και p.

Όπως είχαμε ήδη υποψιαστεί από το προηγούμενο ερώτημα, υψηλότερες τιμές στο N και στο p, επιφέρουν ένα καλύτερα ανακατασκευασμένο σήμα (με μικρότερο MSE).

Είναι επίσης ενδιαφέρον να παρατηρήσουμε ότι από ένα σημείο και μετά υπάρχει ένας κορεσμός στο πόσο μπορεί να συνεισφέρει το p στην μείωση του MSE.

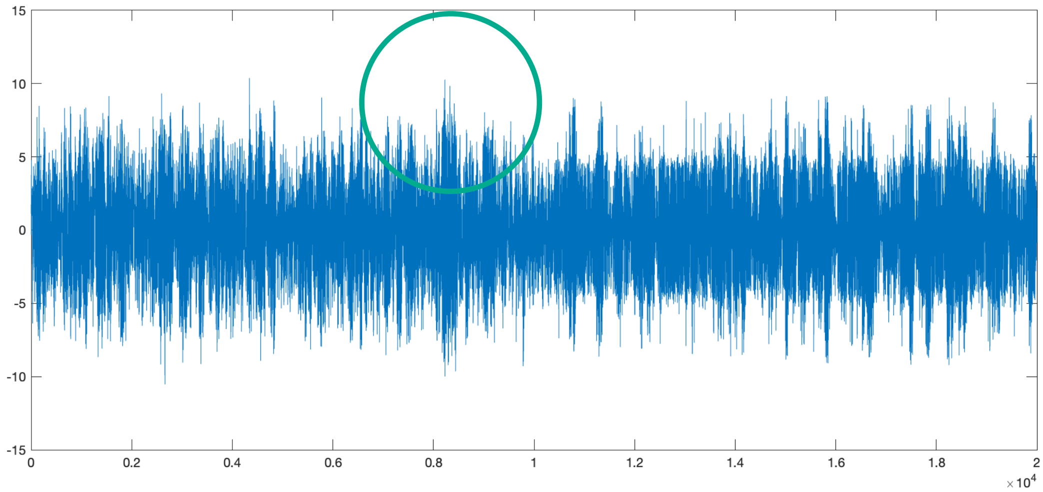
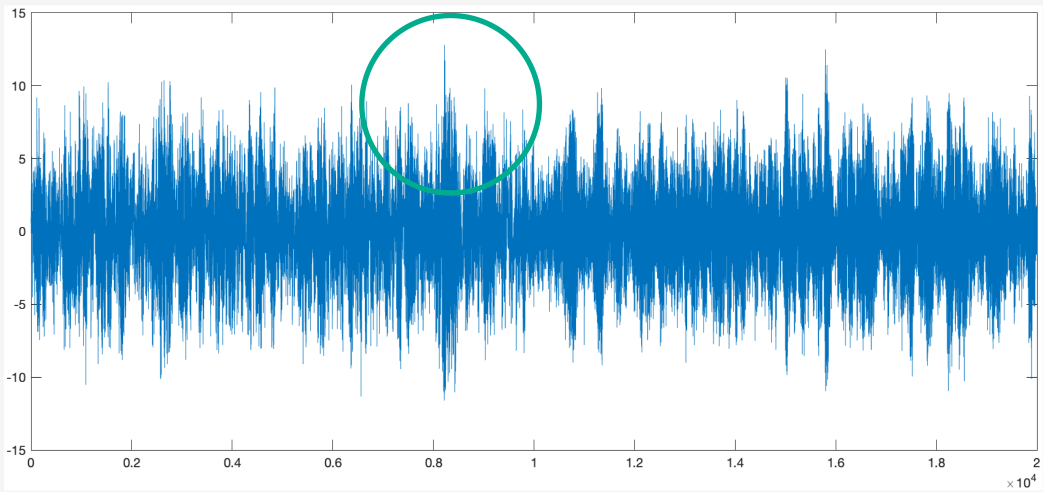
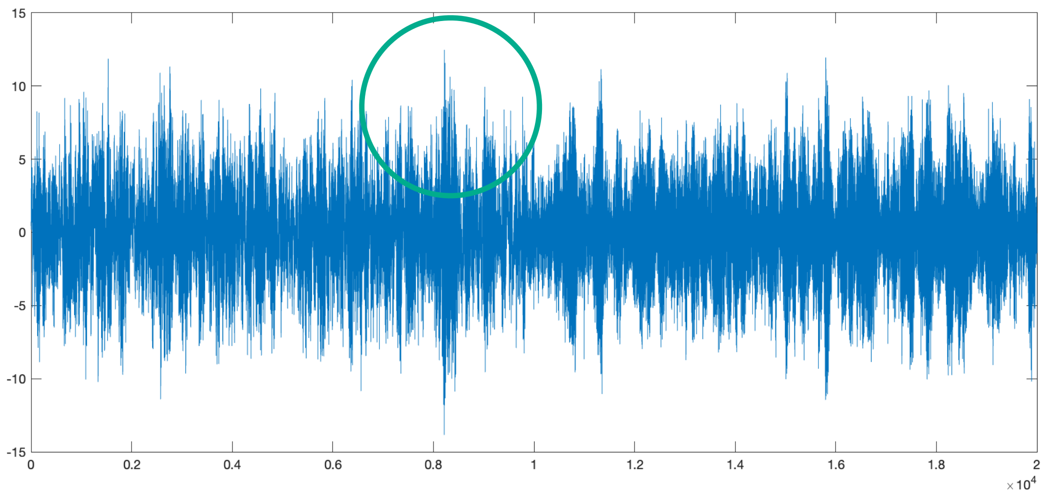
Επίσης άξιο λόγου είναι να αναφέρουμε ότι η αύξηση του N οδηγεί σε δραματική μείωση του MSE.

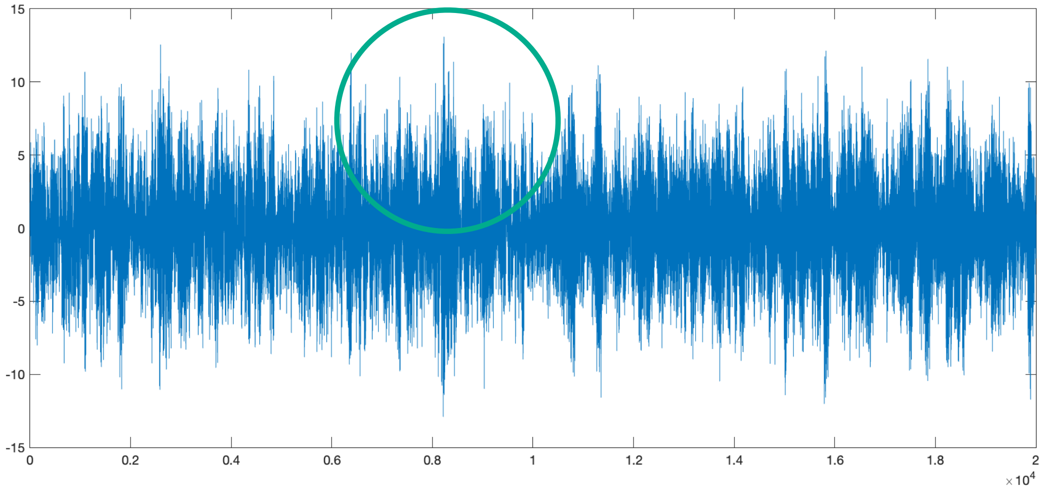
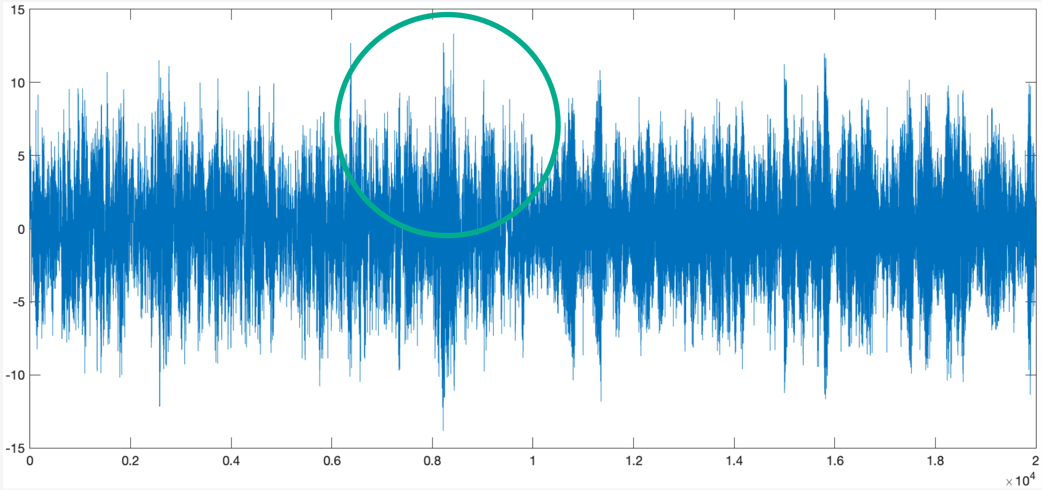
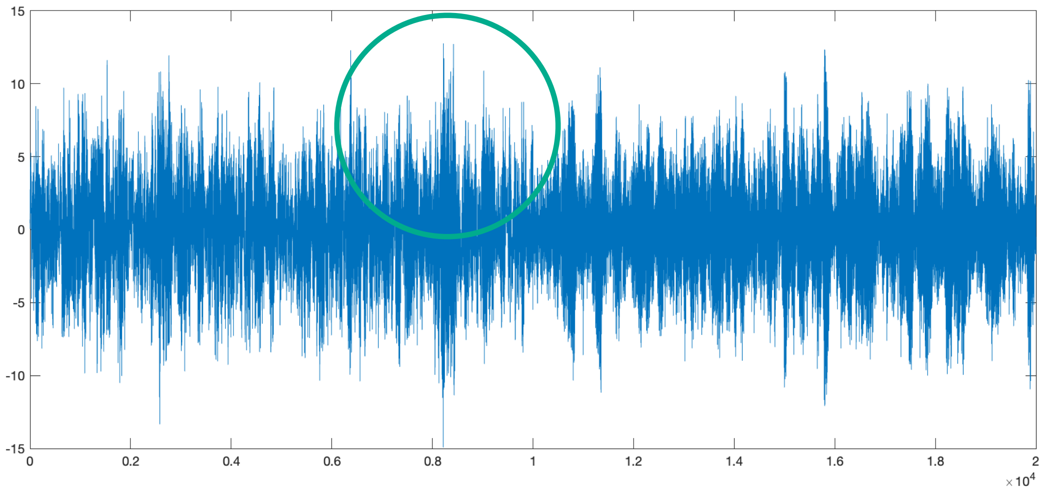
Ακολουθούν οι συντελεστές που παράγονται στα διαφορετικά p. Έχει ενδιαφέρον να σχολιάσουμε ότι γενικά, οι πιο πρόσφατες τιμές του σήματος μπορούν να προβλέψουν γενικά τις παροντικές καλύτερα από τις παλαιότερες.

p=5	p=6	p=7	p=8	p=9	p=10
1.2988	1.2988	1.2715	1.0801	1.1348	1.1074
-1.5723	-1.5723	-1.5176	-1.2988	-1.2988	-1.1895
0.9980	0.9434	1.1895	0.9434	0.9160	0.8887
-0.5332	-0.4785	-0.9434	-0.6152	-0.5605	-0.5605
-0.0410	-0.0957	0.6973	0.2871	0.2051	0.2324
	0.0410	-0.6152	-0.0684	0.0684	-0.0684
		0.5059	0.0684	-0.1504	0.0684
			0.3418	0.5332	0.2324
				-0.1504	0.0957
					-0.2324

Ερώτημα 4

Παρατηρώντας τις παρακάτω ανακατασκευές βλέπουμε ότι όσο μεγαλώνει το N, η ανακατασκευή μας μπορεί να ακολουθήσει όλο και πιο απότομες μεταβολές του αρχικού σήματος. Αυτό μπορεί να γίνει εμφανές στα σημεία για παράδειγμα που έχουν κυκλωθεί στα παρακάτω διαγράμματα.

	p=5
N=1	
N=2	
N=3	

	p=10
N=1	 15 10 5 0 -5 -10 -15 0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 ×10^4
N=2	 15 10 5 0 -5 -10 -15 0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 ×10^4
N=3	 15 10 5 0 -5 -10 -15 0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 ×10^4