



Εργασία 2020-2021

ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ ΚΑΙ ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ

Αντώνιος Σκούρτης | 9142 | 07/02/2021

Περιεχόμενα

Γενικά 1

Level 1 1

 Περιγραφή 1

 SSC..... 1

 Filterbank 1

 AACoder1 2

 demoAACoder1 2

 Αποτελέσματα..... 3

Level 2..... 4

 Περιγραφή 4

 TNS 4

 AACoder2..... 5

 demoAAC2..... 5

 Αποτελέσματα..... 6

Οδηγίες χρήσης..... 7

 Παράδειγμα..... 7

 Προσοχή 7

Κώδικας..... 8



Γενικά

Η παρούσα εργασία σκοπεύει στην υλοποίηση ενός κωδικοποιητή και αποκωδικοποιητή ήχου. Πιο συγκεκριμένα ο κωδικοποιητής/αποκωδικοποιητής ακολουθεί το πρότυπο AAC¹. Από τα τρία επίπεδα της εργασίας υλοποιήθηκαν τα δύο πρώτα, δηλαδή το Level 1 και Level 2.

Level 1

ΠΕΡΙΓΡΑΦΗ

Στο παρόν επίπεδο της εργασίας υλοποιήθηκαν οι βασικές συναρτήσεις για την κωδικοποίηση και αποκωδικοποίηση μίας κυματομορφής.

SSC

Η συνάρτηση SSC υπολογίζει τον τύπο του εκάστοτε παραθύρου, βάσει του τύπου του προηγούμενου παραθύρου και τα δεδομένα του επόμενου.

Παραδοχή 1

Δεδομένου πως και τα δύο κανάλια έχουν ίδιο κοινό προηγούμενο τύπο και βάσει του πίνακα απόφασης που δόθηκε στην εκφώνηση, παρατηρήθηκε πως η διαδικασία μπορεί να απλοποιηθεί υπολογίζοντας το αν είναι *ESH* τουλάχιστον ένα κανάλι από το επόμενο παράθυρο. Αν έστω ένα κανάλι είναι *ESH* τότε και τα δύο θεωρούνται *ESH* και απλώς υπολογίζεται ο εκάστοτε τύπος βάσει του κοινού προηγούμενου καναλιού και του κοινού επόμενου.

Filterbank

Η συνάρτηση filterbank μετατρέπει ένα frame από το πεδίο του χρόνου στο πεδίο της συχνότητας μέσω μετασχηματισμού MDCT.

Ομοίως η αντίστροφη συνάρτηση iFilterbank μετατρέπει ένα frame από το πεδίο της συχνότητας στο πεδίο του χρόνου μέσω αντίστροφου μετασχηματισμού MDCT, iMDCT.

Παραδοχή 1

Οι πίνακες του αποτελέσματος της filterbank φαίνεται να αλλάζουν μορφή αναλόγως με τον τύπο του παραθύρου στην είσοδο. Δηλαδή, αν το παράθυρο δεν είναι *ESH* τότε η επιστροφή είναι της μορφής 1024×2 , αλλιώς είναι της μορφής 128×16 με κάθε ζευγάρι να είναι το αποτέλεσμα του μετασχηματισμού MDCT για το κάθε sub-frame του παραθύρου. Αντί για το παραπάνω αποφασίστηκε τα αποτελέσματα του μετασχηματισμού MDCT για κάθε sub-frame να αποθηκεύονται σε διαδοχικές ομάδες γραμμών καταλήγοντας σε αποτέλεσμα με 8 διαδοχικές ομάδες 128 γραμμών, δηλαδή σε πίνακα της μορφής 1024×2 .

Ομοίως η iFilterbank δέχεται πίνακες 1024×2 τους οποίους μεταφράζει ανάλογα με το αν το παράθυρο ήταν *ESH* ή όχι.

¹ AAC — Advanced Audio Coding ([link](#))



Παραδοχή 2

Ο υπολογισμός του παραθύρου για τον MDCT αλλά και για τον iMDCT μπορεί να γίνει μία φορά, καθώς δεν εξαρτάται από τα δεδομένα της κυματομορφής αλλά μόνο με το μέγεθος του παραθύρου (το οποίο θεωρείται σταθερό). Επομένως το παράθυρο υπολογίζεται μία φορά (την πρώτη φορά που χρησιμοποιείται). Μόλις υπολογιστεί την πρώτη φορά, αποθηκεύεται σε ένα Dictionary² για μελλοντική χρήση. Επομένως ο αλγόριθμος μπορεί να υπολογίσει παράθυρο όποιου μεγέθους και να το επαναχρησιμοποιεί όποτε χρειαστεί.

Παραδοχή 3

Ομοίως οι πίνακες των συνημίτονων που χρησιμοποιεί ο MDCT και ο iMDCT μπορούν να υπολογιστούν μόνο μία φορά για κάθε μέγεθος και απλώς να επαναχρησιμοποιούνται όποτε χρειαστεί. Αυτό επιτυγχάνεται ομοίως με μία δομή Dictionary.

Οι παραπάνω δύο παραδοχές επιταχύνουν τον αλγόριθμο αισθητά.

Παραδοχή 4

Υπάρχει τρόπος να υπολογιστεί ο MDCT μέσω του FFT. Αν είχε υλοποιηθεί ο MDCT βάσει του FFT ο χρόνος εκτέλεσης θα μειωνόταν δραστικά. Ομοίως για τον iMDCT.

AACoder1

Η συνάρτηση AACoder1 δέχεται ολόκληρο το σήμα και το μετατρέπει στην τελική κωδικοποιημένη μορφή του. Ομοίως η αντίστροφη συνάρτηση iAACoder1 μετατρέπει το κωδικοποιημένο σήμα στην αρχική μορφή του (όσο είναι αυτό εφικτό).

Παραδοχή 1

Το σήμα στην AACoder1 πρέπει να χωριστεί σε ακέραιο αριθμό από frames. Επομένως το τελικό frame γεμίζει με μηδενικά αν δεν είναι ολόκληρο.

Παραδοχή 2

Στο σήμα στην AACoder1 πρέπει να προστεθούν στην αρχή και στο τέλος μισά frames γεμάτα μηδενικά, ώστε μετά τον MDCT να μην υπάρχει απώλεια πληροφορίας. Επίσης στην iAACoder1 πρέπει να φύγουν τα μηδενικά από την αρχή και από το τέλος ώστε το σήμα να είναι ίδιο σε μέγεθος με το αρχικό.

Παραδοχή 3

Στην AACoder1 το προηγούμενο frame θεωρείται OLS, και ο τρόπος κωδικοποίησης είναι KBD.

demoAACoder1

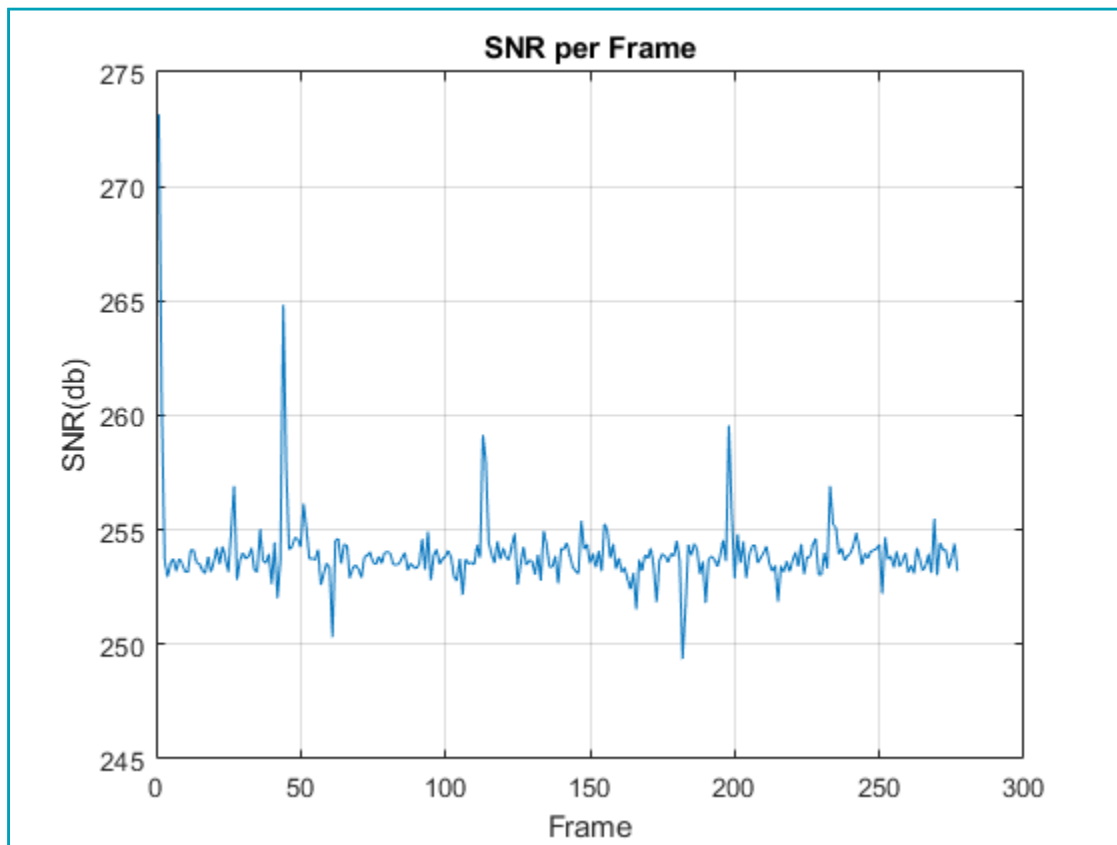
Η συνάρτηση αυτή απλώς συνδυάζει τις υπόλοιπες. Δέχεται δύο αλφαριθμητικά για είσοδο, το όνομα του αρχείου εισόδου και το όνομα του αρχείου εξόδου.

² Dictionary — Γνωστό και ως Map, είναι μία δομή δεδομένων που αντιστοιχίζει ένα κλειδί σε μία τιμή ([link](#))



ΑΠΟΤΕΛΕΣΜΑΤΑ

Τα αποτελέσματα του πρώτου επιπέδου μπορούν να «μετρηθούν» με δύο τρόπους. Πρώτον το αποτέλεσμα αρχείο ακούγεται εξίσου καλά όσο με το εισαγόμενο αρχείο, οπότε η κωδικοποίηση και αποκωδικοποίηση δεν προσθέτουν μεγάλο σφάλμα στο σήμα, τουλάχιστον όχι αρκετά μεγάλο για να ακουστεί. Δεύτερον μπορεί να υπολογιστεί το SNR των δύο σημάτων θεωρώντας το αρχικό σήμα ως είσοδο και το αποτέλεσμα της αποκωδικοποίησης ως έξοδο. Δεδομένων αυτών, η παραπάνω διαδικασία έχει $SNR = 253.9652 \text{ db}$ δηλαδή η διαδικασία επιφέρει σφάλματα της τάξης 10^{-24} σε σχέση με το σήμα.





Level 2

ΠΕΡΙΓΡΑΦΗ

Στο παρόν επίπεδο της εργασία υλοποιήθηκε η συνάρτηση TNS καθώς και η αντίστροφή της iTNS. Οι παραπάνω συναρτήσεις είναι υπεύθυνες για την, όσο το δυνατό, αφαίρεση των μεταβατικών φαινομένων από τα σφάλματα ενός frame.

TNS

Όπως αναφέρθηκε η συνάρτηση TNS προσπαθεί να μειώσει τον θόρυβο των μεταβατικών δεδομένων ώστε το σήμα να θεωρείτε στάσιμο.

Παραδοχή 1

Το σήμα εισόδου είναι πάντα της μορφής 1024×2 . Καθώς η συνάρτηση μπορεί να υπολογίζει την συνάρτηση TNS και για τα δύο κανάλια μαζί. Στην πραγματικότητα η συνάρτηση δέχεται $1024 \times N$ με $N \in \mathbb{N}$.

Παραδοχή 2

Οι συντελεστές στην έξοδο θα είναι της μορφής 4×2 αν το σήμα δεν είναι *ESH*, αλλιώς θα είναι της μορφής 32×2 με κάθε διαδοχική τετράδα γραμμών να είναι οι συντελεστές για το κάθε sub-frame. Κάθε στήλη των συντελεστών αναφέρεται στο ανάλογο κανάλι εισόδου.

Παραδοχή 3

Ο κβαντιστής 4-bit δεν μπορεί να είναι συμμετρικός και νεκρής ζώνης, επομένως μία τιμή του απορρίπτεται και αντί για 16 επίπεδα, έχει 15.

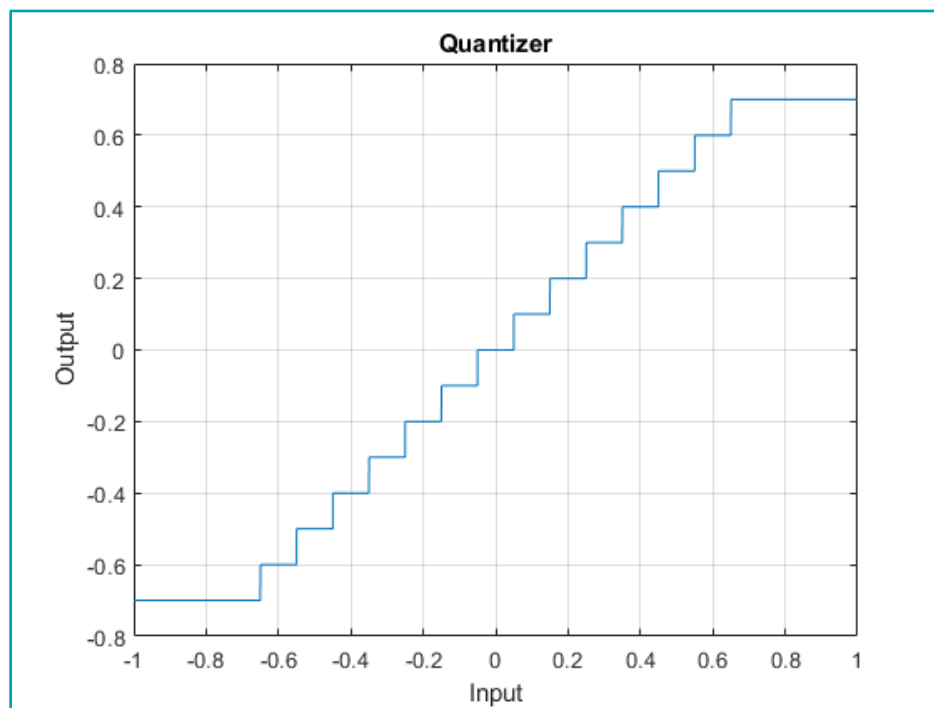


Figure 1: 4-bit κβαντιστής, ομοιόμορφος, συμμετρικός και νεκρής ζώνης



Παραδοχή 4

Θεωρείται πως πάντα οι πόλοι του φίλτρου οδηγούν σε ευστάθεια, δηλαδή πως πάντα $|r_i| < 1$. Προφανώς κάτι τέτοιο δεν είναι αληθές σε κάθε περίπτωση. Ένας τρόπος λύσης του προβλήματος αυτού είναι να εξαναγκαστούν οι πόλοι να βρίσκονται εντός του μοναδιαίου κύκλου.

```
function b = forceInvertible(a)
% Calculate roots
r = roots(a);
% Approach zero and one
z = 0.0001;
u = 1-z;
% Avoid division by real zero
r(r == 0) = z;
% Force roots inside unit circle, approach unit circle boundary
r(r > +1) = +u;
r(r < -1) = -u;
% Retrieve coefficients
b = poly(r);
end
```

Code-Snippet 1: Κώδικας εξαναγκασμού ευστάθειας

Ο παραπάνω αλγόριθμος δεν χρησιμοποιήθηκε καθώς μετά την εφαρμογή του οι συντελεστές δεν είναι κβαντισμένοι. (Αν κβαντιστούν εκ νέου μπορεί να μην είναι ευσταθές το φίλτρο?)

AACoder2

Η συνάρτηση AACoder2 ακολουθεί όμοια λογική με τη συνάρτηση AACoder1, ομοίως και η αντίστροφη iAACoder2. Η συνάρτηση δέχεται το όνομα του αρχείου προς κωδικοποίηση και επιστρέφει το κωδικοποιημένο αποτέλεσμα. Αντιστρόφως, η iAACoder2 δέχεται το κωδικοποιημένο αποτέλεσμα και το αποθηκεύει σε μορφή wav σε ένα αρχείο αφού το αποκωδικοποιήσει.

Παραδοχή

Οι παραδοχές είναι ίδιες με την AACoder1

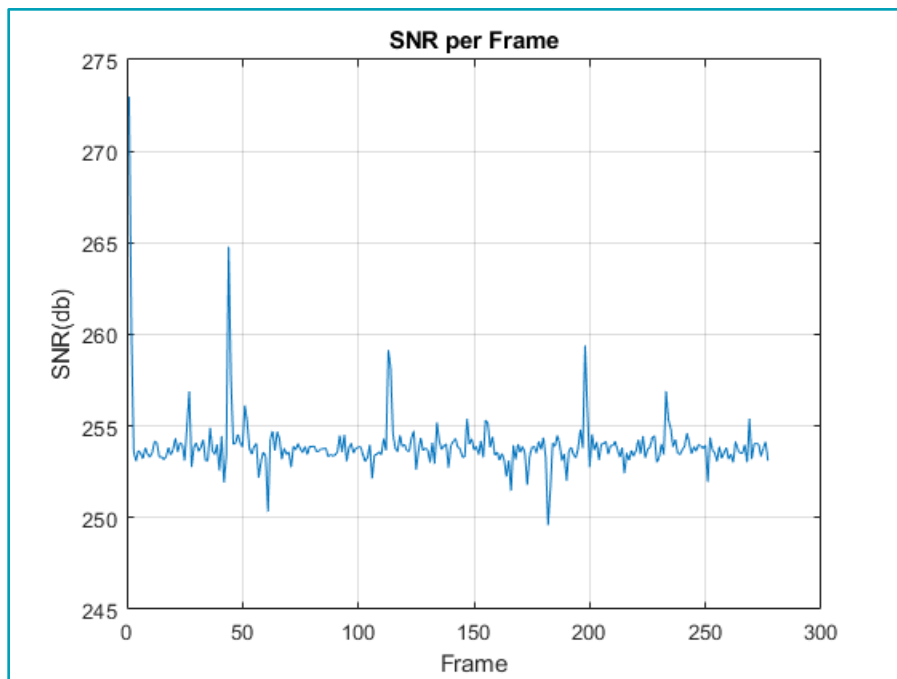
demoAAC2

Η συνάρτηση αυτή απλώς συνδυάζει τις υπόλοιπες. Δέχεται δύο αλφαριθμητικά για είσοδο, το όνομα του αρχείου εισόδου και το όνομα του αρχείου εξόδου.



ΑΠΟΤΕΛΕΣΜΑΤΑ

Ομοίως με το πρώτο επίπεδο, στο δεύτερο το συνολικό $\text{SNR} = 253.96 \text{ db}$ δηλαδή πολύ μικρό σφάλμα σε σχέση με το σήμα εισόδου. Επίσης ακουστικά το αρχικό και το τελικό αρχείο είναι ίδια.





Οδηγίες χρήσης

Τα δύο επίπεδα έχουν το κάθε ένα από μία συνάρτηση demoAAC. Οι συναρτήσεις demo δέχονται για ορίσματα δύο αλφαριθμητικά³, το πρώτο αλφαριθμητικό είναι η σχετική διαδρομή⁴ στο αρχείο εισόδου μορφής WAV με ρυθμό δειγματοληψίας 48kHz. Ομοίως το δεύτερο αλφαριθμητικό είναι η σχετική διαδρομή στο αρχείο εξόδου μορφής WAV με ρυθμό δειγματοληψίας 48kHz.

ΠΑΡΑΔΕΙΓΜΑ

Για να κωδικοποιηθεί μέσω του επιπέδου 1 το αρχείο *LicorDeCalandraca.wav* και να αποθηκευτεί στο αρχείο *result.wav* θα πρέπει να γραφτεί ο παρακάτω κώδικας.

```
% Call demoAAC1  
demoAAC1("LicorDeCalandraca.wav", "result.wav")
```

Code-Snippet 2: Κώδικας χρήσης του επιπέδου 1

Τα αρχεία αυτά είναι τοπικά στον φάκελο, δηλαδή υπάρχουν μέσα στον φάκελο Level1 αν τα αρχεία ήταν σε άλλον φάκελο τότε το αλφαριθμητικό εισόδου θα πρέπει να είναι η σωστή διαδρομή και όχι μόνο το όνομα. Ομοίως και για το αλφαριθμητικό εξόδου.

Επιπλέον για να εκτελεστεί οποιοδήποτε demoAAC πρέπει η MATLAB να γνωρίζει την διαδρομή προς την συνάρτηση demoACC. Το παραπάνω μπορεί να επιτευχθεί είτε πηγαίνοντας και εκτελώντας το αρχείο από τον εκάστοτε φάκελο Level είτε εκτελώντας την συνάρτηση της MATLAB [addpath](#) με κατάλληλα ορίσματα.

ΠΡΟΣΟΧΗ

Στο Level2 πρέπει να υπάρχει πάντα το αρχείο TableB219.mat ώστε να διαβαστούν οι ανάλογες μπάντες για το TNS

Επιπλέον το πρόγραμμα αποθηκεύει *persistent* μεταβλητές, καλό θα ήταν πριν και μετά την εκτέλεση της παραπάνω εργασίας να εκτελεστεί η εντολή **clear classes** για να καθαρίσει η MATLAB τυχών μεταβλητές από άλλες εργασίες.

³ Αλφαριθμητικό — String, όχι πίνακα χαρακτήρων!

⁴ Διαδρομή — Path



Κώδικας

Ο κώδικας μπορεί να βρεθεί στο αποθετήριο στο [github](https://github.com).