

# Συστήματα Πολυμέσων

Υλοποίηση Κωδικοποιητή/Αποκωδικοποιητή AAC



7 Φεβρουαρίου 2021

Δάμη Υπατία 8606

[ypatiapd@ece.auth.gr](mailto:ypatiapd@ece.auth.gr)

# Περιεχόμενα

## 0. Εισαγωγή

### 1. Level 1

1.1 Ανάλυση κώδικα. . . . .	3
1.1.1 Συνάρτηση SSC . . . . .	3
1.1.2 Συνάρτηση Filterbank. . . . .	4
1.1.3 Συνάρτηση AACoder1. . . . .	4
1.1.4 Συνάρτηση iFilterbank . . . . .	4
1.1.4 Συνάρτηση iAACoder. . . . .	4
1.1.4 Συνάρτηση demoAAC1. . . . .	5

### 2. Level 2

2.1 Ανάλυση κώδικα. . . . .	5
2.1.1 TNS. . . . .	5
2.1.2 AACoder2. . . . .	6
2.1.1 iTNS . . . . .	6
2.1.2 iAACoder2 . . . . .	7
2.1.3 demoAAC2 . . . . .	7

### 3. Level 3

3.1 Ανάλυση κώδικα . . . . .	7
3.1.1 psycho . . . . .	8
3.1.2 AACquantizer. . . . .	8
3.1.3 AACoder3. . . . .	8
3.1.2 iAACquantizer . . . . .	9
3.1.3 iAACoder3 . . . . .	9
3.1.4 demoAAC3 . . . . .	9

### 4 .SNR

### 5. Παρατηρήσεις

## 0. Εισαγωγή

Στα πλαίσια αυτής της εργασίας υλοποιήθηκε στο περιβάλλον της Matlab ένας απλουστευμένος Κωδικοποιητής/Αποκωδικοποιητής ήχου , σύμφωνα με τις προδιαγραφές του προτύπου AAC. Η υλοποίηση πραγματοποιήθηκε σε τρία βήματα, προσθέτοντας σε κάθε βήμα μία επιπλέον βαθμίδα του κωδικοποιητή. Στο πρώτο βήμα τα δείγματα του stream αποσυσχετίστηκαν με χρήση του μετασχηματισμού MDCT, στο δεύτερο βήμα προστέθηκε η βαθμίδα TNS, η οποία αφαιρεί τις περιοδικότητες από τους συντελεστές DCT και στο τρίτο βήμα, υλοποιήθηκαν επιπλέον η βαθμίδα εντοπισμού των ισχυρών τόνων του σήματος μέσω του ψυχοακουστικού μοντέλου, ο κβαντιστής και η κωδικοποίηση του σήματος κατά τον κώδικα Huffman.

## 1.Step1

### 1.1 Ανάλυση κώδικα

#### 1.1.1.Συνάρτηση SSC

Η συνάρτηση SSC δέχεται ως ορίσματα το τρέχον frame, το επόμενο frame και τον τύπο του προηγούμενου frame .Μέσα από διαδοχικούς ελέγχους αποφασίζεται ο τύπος του τρέχοντος frame σύμφωνα με τη λογική της εκφώνησης. Εάν υπάρχει περίπτωση το επόμενο frame να είναι τύπου eight short, τότε καλείται η συνάρτηση `check_eight_small()` για να ελεγχθεί η ενέργειά του. Η συνάρτηση αυτή αρχικά εφαρμόζει στο επόμενο frame ένα υπεραπλοποιημένο φίλτρο μέσω της συνάρτησης `filter` της matlab .Έπειτα το frame χωρίζεται σε επικαλυπτόμενα subframes μήκους 128 δειγμάτων και ελέγχεται για το κάθε κανάλι και για το κάθε subframe ξεχωριστά η εκτίμηση ενέργειας του. Υπολογίζονται τα attack values και αν ικανοποιούνται οι συνθήκες το κανάλι του τρέχοντος frame κατηγοριοποιείται ως eight short. Τέλος καλείται η συνάρτηση `frameTypeMutual()` για να αποφασιστεί μέσω της χρήσης του look up table της εκφώνησης ο τύπος του τρέχοντος frame.

### 1.1.2 Συνάρτηση Filterbank

Η συνάρτηση Filterbank εφαρμόζει στα δείγματα του κάθε frame τον μετασχηματισμό MDCT, με σκοπό την παραγωγή αποσυσχετισμένων συντελεστών DCT. Σε περίπτωση που πρόκειται για eight short frame, η διαδικασία εφαρμόζεται ξεχωριστά σε καθένα από τα 8 subframes του. Επίσης η διαδικασία εφαρμόζεται σε κάθε κανάλι ξεχωριστά. Αρχικά, ανάλογα με το είδος του frame δημιουργείται το αντίστοιχο παράθυρο, σύμφωνα με τις οδηγίες της εκφώνησης μέσω της συνάρτησης window(). Για SIN παράθυρα χρησιμοποιήθηκε η συνάρτησης sin\_window(), και το kbdwin η συνάρτηση bdwin() της matlab. Έπειτα το κάθε frame, ένα για long frame και 8 για short, πολλαπλασιάζεται με το παράθυρο που δημιουργήθηκε προηγουμένως. Στο προϊόν του πολλαπλασιασμού εφαρμόζουμε τον μετασχηματισμό mdct μέσω της συνάρτησης mdct() του Μάριου Αθηναίου που υπάρχει στο github. Οι συντελεστές mdct που επιστρέφονται από τη συνάρτηση έχουν το μισό πλήθος από τα δείγματα του frame στο πεδίο του χρόνου.

### 1.1.3 Συνάρτηση AACoder1

Η συνάρτηση AACoder1 αποτελεί τον κωδικοποιητή του πρώτου βήματος. Αρχικά διαιρεί το σήμα εισόδου σε frames και εφαρμόζει zero padding στην αρχή και στο τέλος του σήματος έτσι ώστε να είναι πολλαπλάσιο του 1024 και να μην χαθεί πληροφορία. Στη συνέχεια μέσα σε μια επαναληπτική διαδικασία με επαναλήψεις όσες και ο αριθμός των προς κωδικοποίηση frames, εφαρμόζεται σε κάθε frame η επιλογή του τύπου του και ο μετασχηματισμός MDCT και τα αποτελέσματα αποθηκεύονται στην δομή AACSeq1 για να αποκτήσει πρόσβαση σε αυτά στη συνέχεια ο αποκωδικοποιητής.

### 1.1.4 Συνάρτηση iFilterbank

Η συνάρτηση iFilterbank αντιστρέφει την διαδικασία του μετασχηματισμού mdct, σε κάθε frame ξεχωριστά. Χρησιμοποιήθηκε η συνάρτηση imdct() του Μάριου Αθηναίου, που βρίσκεται στο github. Το κάθε frame πολλαπλασιάζεται με το ίδιο παράθυρο που χρησιμοποιήθηκε πριν τον μετασχηματισμό. Σε περίπτωση που το frame είναι eight short, σχηματίζουμε από τα subframes το αρχικό frame. Το frame που επιστρέφεται έχει τιμές στο πεδίο του χρόνου και μήκος 2048 δείγματα.

### 1.1.5 Συνάρτηση iAACoder

Η συνάρτηση iAACoder αποτελεί τον αποκωδικοποιητή του σήματος. Αρχικά, μέσα σε μια επαναληπτική διαδικασία με αριθμό επαναλήψεων ίσο με τον αριθμό των προς αποκωδικοποίηση frames, εφαρμόζει σε κάθε frame τη συνάρτηση iFilterbank(), και έπειτα ανασηματίζει το αρχικό

stream προσθέτοντας τα επικαλυπτόμενα frames(add and overlap) .Η διαδικασία αποκωδικοποίησης φαίνεται στον παρακάτω κώδικα

```
index=0;
for i= 1:frames
    frameF(N/2,2)=zeros();
    frameF(:,1)= AACSeq1(i).chl.frameF;
    frameF(:,2)=AACSeq1(i).chr.frameF;
    frameType=AACSeq1(i).frameType;
    winType=AACSeq1(i).winType;
    frameT=iFilterbank(frameF,frameType,winType);
    decoded(index+1:index+N,1)=decoded(index+1:index+N,1)+frameT(1:N,1);
    decoded(index+1:index+N,2)= decoded(index+1:index+N,2)+frameT(1:N,2);
    index=index+N/2;
end
```

Τέλος αφαιρούνται τα πρόσθετα μηδενικά που είχαν προστεθεί αρχικά στο σήμα , και το αποκωδικοποιημένο stream επιστρέφεται από τη συνάρτηση στο demo.

### 1.1.6 Συνάρτηση demoAAC1

Η συνάρτηση demoAAC1() καλείται από το χρήστη έτσι ώστε να κωδικοποιηθεί/αποκωδικοποιηθεί το stream της επιλογής του. Δέχεται ως είσοδο το όνομα του αρχείου προς κωδικοποίηση και το όνομα του αρχείου που θα αποθηκευτεί το αποκωδικοποιημένο stream.Η συνάρτηση όταν τελειώσει η αποκωδικοποίηση επιστρέφει το SNR της κωδικοποίησης.

## 2.Step 2

### 2.1 Ανάλυση κώδικα

#### 2.1.1 Συνάρτηση TNS

Η συνάρτηση TNS() δέχεται ως είσοδο τους συντελεστές DCT που προέκυψαν για κάθε frame και αφαιρεί υπάρχουσες περιοδικότητες με την παρακάτω διαδικασία. Αρχικά , κανονικοποιεί τους συντελεστές DCT καλώντας τη συνάρτηση normalize(), με όρισμα τις μπάντες συχνοτήτων του ψυχοακουστικού μοντέλου, με την χρήση των οποίων θα γίνει η κανονικοποίηση. Έπειτα στη συνάρτηση normalize() , για κάθε μπάνα υπολογίζεται η ενέργεια των δειγμάτων του frame που ανήκουν σε αυτή, και στη συνέχεια οι συντελεστές MDCT που αντιστοιχούν στην εκάστοτε μπάνα κανονικοποιούνται με βάση την ενέργεια που υπολογίστηκε .Η διαδικασία αυτή φαίνεται στον παρακάτω κώδικα

```
for i =1:size(bands,2)-1
    k=bands(i)+1:bands(i+1);
```

```

P(i)=sum(frameFin(k).^2);
Sw(k)=sqrt(P(i));
k=K1;
while k>=1
    Sw(k)=(Sw(k)+Sw(k+1))/2;
    k=k-1;
end
k=2;
while k<=K2
    Sw(k)=(Sw(k)+Sw(k-1))/2;
    k=k+1;
end
end
Xw=frameFin ./ Sw;

```

Έπειτα, οι κανονικοποιημένοι συντελεστές κβαντίζονται από έναν ομοιόμορφο και συμμετρικό κβαντιστή μέσω της συνάρτησης `quantise()`. Η διαδικασία κβαντισμού για frames τύπου `eight short` φαίνεται στον παρακάτω κώδικα

```

for j=1:8
    for i=1:4
        quantised_coeffs(i,j) = round(coeffs(i,j)*10)/10 ;
        if quantised_coeffs(i,j)~=0
            if quantised_coeffs(i,j)>0
                quantised_coeffs(i,j)=quantised_coeffs(i,j)+0.05;
            else
                quantised_coeffs(i,j)=quantised_coeffs(i,j)-0.05;
            end
        end
        if quantised_coeffs(i,j)<(-0.75)
            quantised_coeffs(i,j)=-0.75;
        end
        if quantised_coeffs(i,j)>(0.75)
            quantised_coeffs(i,j)=0.75;
        end
    end
end
end

```

Τέλος στις κβαντισμένες τιμές εφαρμόζεται ένα φίλτρο, μέσω της συνάρτησης `FIR()`;

## 2.1.2 Συνάρτηση AACoder2

Η συνάρτηση `AACoder2()` αποτελεί τον κωδικοποιητή του δεύτερου βήματος . Η διαδικασία που ακολουθεί είναι η ίδια με τον κωδικοποιητή του πρώτου βήματος, με την επιπλέον εφαρμογή της συνάρτησης `TNS()` στους συντελεστές DCT που επέστρεψε η συνάρτηση `Filterbank`.

## 2.1.3 Συνάρτηση iTNS

Η συνάρτηση `iTNS` υλοποιεί την αντιστροφή της διαδικασίας που εφαρμόστηκε στην συνάρτηση `TNS()` . Η μόνη διαδικασία που απαιτείται για την αντιστροφή διαδικασία είναι η εφαρμογή του αντιστροφου φίλτρου στους συντελεστές DCT , μέσω της συνάρτησης `filter()` της `matlab`.

### 2.1.4 Συνάρτηση iAACoder2

Η συνάρτηση αυτή αποτελεί τον αποκωδικοποιητή του δεύτερου βήματος .. Η διαδικασία που ακολουθεί είναι η ίδια με τον αποκωδικοποιητή του πρώτου βήματος, με την επιπλέον εφαρμογή της συνάρτησης iTNS() στους συντελεστές DCT στην αρχή της διαδικασίας.

### 2.1.5 Συνάρτηση demoAAC2

Η λειτουργία της συνάρτησης αυτής είναι η ίδια με την λειτουργία της συνάρτησης demoAAC1 του πρώτου βήματος .

## 3. Step 3

### 3.1 Ανάλυση κώδικα

#### 3.1.1 Συνάρτηση Spreading\_function

Η συνάρτηση spreading\_function() υπολογίζει τη διασπορά μεταξύ των μπαντών των πινάκων B.2.1.9.a και B.2.1.9.b .Η συνάρτηση εκτελείται αυτόνομα πριν την κωδικοποίηση και τα αποτελέσματα αποθηκεύονται στους πίνακες spread\_a.mat και spread\_b.mat, έτσι ώστε να μην χρειάζεται να εκτελούνται οι υπολογισμοί κατά την ώρα της κωδικοποίησης.

#### 3.1.2 Συνάρτηση psycho

Η συνάρτηση psycho() υπολογίζει το masking ratio για την κάθε μπάντα .Φορτώνονται οι πίνακες spread\_a.mat και spread\_b.mat που υπολογίστηκαν από την spreading\_function() και εκτελούνται διαδοχικά τα βήματα της εκφώνησης για τον εντοπισμό των τόνων του σήματος .Η συνάρτηση sub\_frames() διαιρεί σε περίπτωση eight short to frame σε subframes για να εφαρμοστεί η διαδικασία σε κάθε ένα από αυτά. Για τον μετασχηματισμό fourier των frames χρησιμοποιήθηκε η συνάρτηση fft() της matlab, ενώ για τον υπολογισμό του πλάτους και της φάσης του μετασχηματισμού χρησιμοποιήθηκαν οι συναρτήσεις abs() και angle() της matlab.

### 3.1.3 Συνάρτηση AACquantiser

Η συνάρτηση αυτή υλοποιεί τον κβαντιστή των συντελεστών DCT στο τρίτο βήμα της εργασίας. Υπολογίζεται η αρχική τιμή της παραμέτρου  $a$  και έπειτα μέσω μιας επαναληπτικής διαδικασίας, αυξάνεται κατά 1 έως ότου φτάσει το κατώφλι ακουστότητας. Κατά την επανάληψη καλείται η συνάρτηση `quant_error()` που υπολογίζει το σφάλμα κβαντισμού για κάθε συντελεστή. Η διαδικασία φαίνεται στον παρακάτω κώδικα και εκτελείται ξεχωριστά για τα δείγματα που ανήκουν σε κάθε μπάντα, εφόσον θα έχουν κοινό συντελεστή  $a$

```
for j =2:size(matrix,1)
    k=matrix(j,2)+1:matrix(j,3)+1;
    P=sum(frameF(k,i).^2);
    T=P/SMR(j,i);
    while true
        a = a1(j);
        P = quant_error(frameF(k,i), a)
        if P<T
            a1(j) = a + 1;
        end
        if (a1(j) == a) || (abs(a1(j)-a1(j-1)) > 60)
            break
        end
    end
    S(k, i) = quantize(frameF(k,i),a1(j));
end
```

Αφού έχουν επιλεγεί οι κατάλληλοι συντελεστες  $a$  για όλες τις μπάντες, οι συντελεστές DCT ολου του frame κβαντίζονται μέσω της συνάρτησης `quantise()`, σύμφωνα με τις σχέσεις κβαντισμού της εκφώνησης.

### 3.2.4 Συνάρτηση AACCoder3

Η συνάρτηση `AACCoder3` αποτελεί τον κωδικοποιητή του βήματος. Η διαδικασία που ακολουθείται είναι ίδια με το δεύτερο βήμα, με επιπλέον τις συναρτήσεις `psycho()`, `AACquantiser()` και `encodeHuff()` που πραγματοποιούν την εύρεση των τόνων του stream, τον κβαντισμό των συντελεστών και την κωδικοποίησή τους κατά Huffman.

### 3.2.5 Συνάρτηση iAACquantiser

Υλοποιεί τον αποκβαντισμό των συμβόλων σύμφωνα με την σχέση της εκφώνησης, μέσω της συνάρτησης `de_quantise()`.



### 3.2.6 Συνάρτηση iAACoder3

Η συνάρτηση αυτή αποτελεί τον αποκωδικοποιητή του τρίτου βήματος . Η διαδικασία που ακολουθεί είναι η ίδια με τον αποκωδικοποιητή του δεύτερου βήματος, με την επιπλέον εφαρμογή της συνάρτησης decodeHuff() και iAACquantise()

### 3.2.7 Συνάρτηση demoAAC3

Η συνάρτηση demoAAC3 ,όπως και στα προηγούμενα βήματα εκτελεί το demo της κωδικοποίησης.

## 4 . SNR

Το SNR που υπολογίστηκε για το κάθε βήμα είναι

Step 1: 8.1582

Step 2: 7.1458

Step 3: 4.8094

## 5. Παρατηρήσεις

Το SNR των πρώτων δύο βημάτων δεν είναι το επιθυμητό , καθώς και ο χρόνος διεκπεραίωσης του τρίτου βήματος είναι μεγαλύτερος από τον προβλεπόμενο(περίπου 1,30 λεπτό) . Λόγω προθεσμίας δεν μπόρεσα να εντοπίσω την αιτία.