

Συστήματα Πολυμέσων και Εικονική Πραγματικότητα

Εργασίες 2024-25

A. Ντελόπουλος

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

1 Εισαγωγικές Παρατηρήσεις

Η παρακάτω εργασία αποτελεί *προαιρετικό* μέρος του μαθήματος Συστήματα Πολυμέσων και η εκτέλεσή της συνεισφέρει 1, 3 ή 4 επιπλέον μονάδες στην τελική βαθμολογία.

Η εργασία θα πρέπει να εκτελεστεί σε ομάδες των δύο ατόμων.

Η εργασία αποτελείται από 3 ενότητες. Η υλοποίηση της εργασίας μπορεί κατ' επιλογή να περιλάβει την πρώτη ενότητα, την πρώτη και τη δεύτερη, κ.ο.κ. συνεισφέροντας τις αντίστοιχες μονάδες για κάθε ενότητα. Δεν μπορεί όμως να εκτελεστεί μία ενότητα χωρίς να έχει ορθά εκτελεστεί η προηγούμενή της.

Η προτεινόμενη εργασία στοχεύει στην υλοποίηση ενός κωδικοποιητή / αποκωδικοποιητή φωνής κατά το πρότυπο ETSI GSM 06.10 που χρησιμοποιεί την τεχνολογία Regular Pulse Excitation.

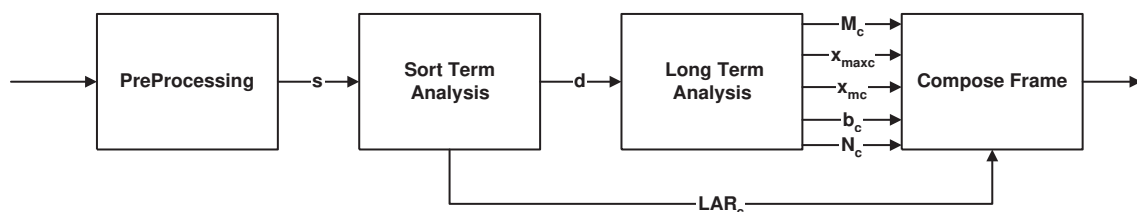
2 Κωδικοποίηση Φωνής σύμφωνα με το πρότυπο ETSI GSM 06.10

Στην ενότητα αυτή παρουσιάζονται οι βασικές λειτουργίες του κωδικοποιητή/αποκωδικοποιητή που θα υλοποιηθεί. Όποτε είναι εφικτό δίνεται παραπομπή στο κείμενο του αντίστοιχου προτύπου (ETSI EN 300 961 V8.1.1. 2000. "GSM 6.10 - Digital cellular telecommunications system (Phase 2+); Full rate speech; Transcoding." *Sophia Antipolis Cedex-France*.)

2.1 Ο Κωδικοποιητής

Σύμφωνα με το πρότυπο ο κωδικοποιητής επεξεργάζεται το σήμα φωνής (δειγματοληπτημένο στα 8000 δείγματα/sec) κατά τμήματα (frames) μήκους 160 δειγμάτων. Επιπλέον κάθε frame υποδιαιρείται σε 4 subframes μήκους 40 δειγμάτων. Όπως αναλύεται στη συνέχεια κάποια στάδια της κωδικοποίησης εκτελούνται σε επίπεδο frame και κάποια άλλα σε επίπεδο subframe.

Στην Εικόνα 1 παρουσιάζονται οι υψηλού επιπέδου λειτουργίες του κωδικοποιητή.



Σχήμα 1: Υψηλού επιπέδου διάγραμμα λειτουργίας του κωδικοποιητή

2.1.1 Preprocessing

Τα αρχικά δείγματα φωνής ενός frame περνούν από δύο διαδοχικά φίλτρα που υλοποιούν τις διαδικασίες Offset compensation και Pre-emphasis όπως προδιαγράφονται στις Ενότητες 3.1.1 και 3.1.2 του προτύπου. Για το επεξεργασμένο σήμα φωνής θα χρησιμοποιούμε το σύμβολο $s(n)$.

2.1.2 Short Term Analysis

Η διαδικασία αυτή εκτελείται στο επίπεδο του frame. Στόχοι της είναι: (α) Να υπολογιστούν οι συντελεστές του βέλτιστου γραμμικού short term προβλέπτη τάξης 8, αυτού δηλαδή του προβλέπτη που μπορεί να εξουδετερώσει την ημιτονοειδή εξάρτηση των φωνημάτων. (β) Να χρησιμοποιήσει τον προβλέπτη αυτόν ώστε να μειώσει την short term συσχέτιση των δειγμάτων φωνής παράγοντας ένα σήμα “σφαλμάτων πρόβλεψης” $d(n)$.

Βήμα (α)- Εκτίμηση βέλτιστου short term προβλέπτη: Για λόγους απλούστευσης ο τροποποιημένος κωδικοποιητής της παρούσας εργασίας υλοποιεί το βήμα (α) με τρόπο διαφορετικό από το πρότυπο. Συγκεκριμένα οι συντελεστές a_k ($k = 1 \dots 8$) του προβλέπτη:

$$\hat{s}(n) = \sum_{k=1}^8 a_k s(n-k), \quad (1)$$

που χρησιμοποιούνται για να παράγουν το “σφάλμα πρόβλεψης” (residual)

$$d(n) = s(n) - \hat{s}(n), \quad (2)$$

υπολογίζονται από την ελαχιστοποίηση του κριτηρίου

$$J_w = E\{d(n)^2\} = E\{(s(n) - \hat{s}(n))^2\} \quad (3)$$

ως προς $\mathbf{w} = [a_1, \dots, a_8]^T$. Η βέλτιστη τιμή των συντελεστών προκύπτει ως λύση των Κανονικών Εξισώσεων,

$$\mathbf{R}\mathbf{w} = \mathbf{r} \quad (4)$$

όπου ο 8×8 πίνακας \mathbf{R} και το 8×1 διάνυσμα στήλης \mathbf{r} αποτελούνται από συντελεστές της συνάρτησης αυτοσυσχέτισης του σήματος $s(n)$. Συγκεκριμένα,

$$\mathbf{R} = \begin{bmatrix} r_s(0) & r_s(1) & \dots & r_s(6) & r_s(7) \\ r_s(1) & r_s(0) & \dots & r_s(5) & r_s(6) \\ \vdots & & \dots & & \vdots \\ r_s(7) & r_s(6) & \dots & r_s(1) & r_s(0) \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} r_s(1) \\ r_s(2) \\ \vdots \\ r_s(8) \end{bmatrix}. \quad (5)$$

Στην πράξη δεν είναι διαθέσιμες οι τιμές της αυτοσυσχέτισης $r(k)$ αλλά χρησιμοποιούνται εκτιμήτριες όπως αυτή που περιγράφεται στην Ενότητα 3.1.4 του προτύπου. Θέτουμε συνεπώς $r_s(k) \simeq ACF(k)$.

Μετά τον υπολογισμό του \mathbf{w} και συνεπώς των συντελεστών πρόβλεψης που αυτό περιέχει, οι συντελεστές a_k κβαντίζονται και κωδικοποιούνται.

Για λόγους συμβατότητας με το πρότυπο αντί των συντελεστών πρόβλεψης ο κβαντισμός και η κωδικοποίηση θα εφαρμοστούν σε ένα σύνολο (1-1) ισοδύναμων ποσοτήτων που ονομάζονται Log-Area Ratios (LAR). Στην πραγματικότητα οι τελευταίες παράγονται ως εκφράσεις των Συντελεστών Ανάκλασης (Reflection Coefficients) $r(i)$ οι οποίοι με τη σειρά τους εκφράζονται σα συνάρτηση των a_k . Ισχύει δηλαδή το σχήμα:

$$a_k, k = 1 \dots 8 \quad < \text{---} > \quad r(i), i = 1 \dots 8 \quad < \text{---} > \quad LAR(i), i = 1 \dots 8 \quad (6)$$

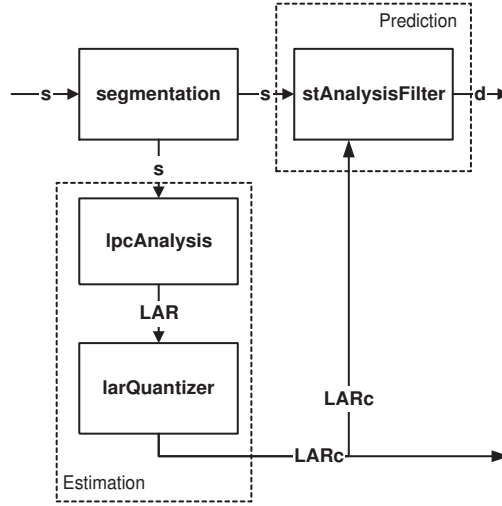
Ο αλγόριθμος μετατροπής $\{a_k, k = 1 \dots 8\} \quad < \text{---} > \quad \{r(i), i = 1 \dots 8\}$ θα σας δοθεί έτοιμος. Η μετατροπή $\{r(i), i = 1 \dots 8\} \quad < \text{---} > \quad \{LAR(i), i = 1 \dots 8\}$ υλοποιείται σύμφωνα με τις Ενότητες 3.1.6 και 3.1.10 του προτύπου.

Για τον κβαντισμό και την κωδικοποίηση των παραγόμενων $LAR(i)$ θα χρησιμοποιηθούν οι προδιαγραφές της Ενότητας 3.1.7 του προτύπου.

Βήμα (β)- short term πρόβλεψη: Οι συντελεστές a_k που υπολογίστηκαν στο Βήμα (α) χρησιμοποιούνται για να παραχθεί το residual $d(n)$ σύμφωνα με την εξίσωση (2).

Ορισμένες παρατηρήσεις διαφωτίζουν τη διαδικασία:

1. Ο συνδυασμός των εξισώσεων (2) και (1) επιτρέπει τον υπολογισμό του $d(n)$ από το $s(n)$ με τη χρήση ενός FIR φίλτρου μήκους 9 με συντελεστές $[1, -a'_1, \dots, -a'_8]$.



Σχήμα 2: Διάγραμμα λειτουργίας της διαδικασίας short term analysis στον κωδικοποιητή. Το Βήμα (α) αντιστοιχεί στις βαθμίδες που χαρακτηρίζονται Estimation και το Βήμα (β) στη βαθμίδα που χαρακτηρίζεται prediction.

2. Οι ανωτέρω συντελεστές a'_k δεν είναι ακριβώς αυτοί που πρωτογενώς προκύπτουν από τη λύση της Εξίσωσης (5). Αντί αυτών, ο κωδικοποιητής μιμούμενος τον αποκωδικοποιητή χρησιμοποιεί τις αντίστοιχες αλλά ελαφρώς τροποποιημένες τιμές που προκύπτουν από την αποκωδικοποίηση των *κβαντισμένων* $LAR(i)$.
3. Τέλος, σύμφωνα με την Ενότητα 3.1.9 του προτύπου προτείνεται η χρήση διαφορετικών τιμών των a'_k για 4 διαφορετικές περιοχές κάθε frame. Οι τιμές αυτές προκύπτουν από παρεμβολή μεταξύ των συντελεστών του τρέχοντος και του προηγούμενου frame. Στόχος αυτής της παρεμβολής είναι να αποφευχθούν απότομες αλλαγές των formants στα όρια των frames. Πάντως για λόγους απλούστευσης η μέριμνα αυτή μπορεί να παραληφθεί.

Στο Σχήμα 2 απεικονίζονται σχηματικά οι παραπάνω διεργασίες.

2.1.3 Long Term Analysis

Η διαδικασία αυτή εκτελείται μια φορά για κάθε subframe και μπορεί να χωριστεί σε 4 διαφορετικές υπολειτουργίες: (α) Εκτίμηση (Estimation) των παραμέτρων long term πρόβλεψης. (β) Πρόβλεψη (Prediction) του (τελικού) σφάλματος πρόβλεψης (long term prediction residual). (γ) Σχηματισμό της ακολουθίας διέγερσης που θα σταλεί στον αποκωδικοποιητή (excitation computation). (δ) Σύνθεση (Synthesis) των short term residual *προηγούμενων* subframes με σκοπό να χρησιμοποιηθούν στη διαδικασία εκτίμησης (των υπολειτουργιών (α) και (β)).

Υπολειτουργία (α) - Εκτίμηση (Estimation): Στην ενότητα αυτή εκτιμώνται οι παράμετροι long term πρόβλεψης δηλαδή το pitch period, N και ο παράγοντας ενίσχυσης/απόσβεσης, b , που θα χρησιμοποιηθούν κατά την πρόβλεψη ενός δείγματος της ακολουθίας $d(n)$ από ένα απομακρυσμένο προηγούμενο δείγμα:

$$\hat{d}(n) = b d(n - N) . \quad (7)$$

Η διαδικασία εκτίμησης στηρίζεται στην παρατήρηση ότι το σήμα φωνής (όταν είναι έμφωνη) παρουσιάζει μία περιοδικότητα (pitch) που αντιστοιχεί στην περιοδικότητα της ταλάντωσης των φωνητικών χορδών. Η αντίστοιχη περίοδος είναι κατά πολύ μεγαλύτερη από την περίοδο της ημιτονοειδούς ταλάντωσης με την οποία καταπιάνεται η προηγούμενη διαδικασία short term prediction και ως εκ τούτου συνεχίζει να υπάρχει στο σήμα $d(n)$. Για το δεδομένο ρυθμό δειγματοληψίας (8000 δείγματα/sec) το πρότυπο θεωρεί ότι η περίοδος αυτή (δηλαδή το N) κυμαίνεται στο διάστημα $[40, \dots, 120]$ δειγμάτων. Για την εκτίμηση του επιχειρείται το ταίριασμα των δειγμάτων $d(n)$ του τρέχοντος subframe με μια ισομήκη περιοχή δειγμάτων των προηγούμενων subframes. Ακριβέστερα αντί να χρησιμοποιηθούν τα παλαιότερα δείγματα $d(n)$ όπως πρωτογενώς είχαν παραχθεί από τη βαθμίδα του short term prediction χρησιμοποιούνται ανακατασκευασμένες εκδοχές τους $d'(m)$ που δημιουργούνται από τη λειτουργία Σύνθεσης που θα περιγραφεί παρακάτω.

Έτσι αν επεξεργαζόμαστε το $j - th$ subframe του τρέχοντος frame και k_0 είναι ο δείκτης του πρώτου δείγματος αυτού του frame, θα χρειαστεί να ταιριάζουμε τα δείγματα

$$[d(k_0 + 40j), \dots, d(k_0 + 40j + 39)]$$

σε μία περιοχή 40 δειγμάτων από το παρελθόν, της μορφής

$$[d'(k_0 + 40j - \lambda), \dots, d'(k_0 + 40j + 39 - \lambda)], \lambda \in [40, \dots, 120]$$

Το N θα πάρει τιμή ίση προς εκείνο το λ για το οποίο παρουσιάστηκε το καλύτερο ταίριασμα. Ως κριτήριο ταιριάσματος χρησιμοποιείται η εταιροσυσχέτιση των δύο σημάτων:

$$R(\lambda) = \sum_{i=0}^{39} d(k_0 + 40j + i) d'(k_0 + 40j + i - \lambda) \quad (8)$$

και τότε

$$N = \underset{\lambda \in [40, \dots, 120]}{\operatorname{maximizer}} R(\lambda) \quad (9)$$

Έχοντας εκτιμήσει το N η παράμετρος ενίσχυσης/απόσβεσης b υπολογίζεται ως

$$b = \frac{\sum_{i=0}^{39} d(k_0 + 40j + i) d'(k_0 + 40j + i - N)}{\sum_{i=0}^{39} d'(k_0 + 40j + i - N)^2} \quad (10)$$

Η διαδικασία αυτή επαναλαμβάνεται για κάθε subframe δηλαδή για $j = 0, \dots, 3$.

Μετά την εκτίμησή τους, οι τιμές των a και b κβαντίζονται και κωδικοποιούνται όπως περιγράφεται στις Ενότητες 3.1.14 και 3.1.15 του προτύπου.

Υπολειτουργία (β) - Πρόβλεψη (Prediction): Η λειτουργία αυτή χρησιμοποιεί τις παραμέτρους N και b (για την ακρίβεια τις κβαντισμένες εκδοχές τους N_c και b_c αντίστοιχα για να σχηματίσει το σφάλμα (residual, $e(n)$ της long term πρόβλεψης:

$$e(n) = d(n) - \hat{d}(n) \quad (11)$$

όπου η πρόβλεψη $\hat{d}(n)$ έχει περιγραφεί στην Εξίσωση (7). Για την ακρίβεια αντί του προβλέπτη της Εξίσωσης (7) χρησιμοποιείται ο

$$d''(n) = b_c d'(n - N_c). \quad (12)$$

ο οποίος χρησιμοποιεί τις κβαντισμένες παραμέτρους N_c και b_c και δείγματα της ανακατασκευασμένης ακολουθίας $d'(n)$ αντί δειγμάτων της πρωτογενούς $d(n)$. Κατ' αυτό τον τρόπο η πρόβλεψη στηρίζεται στην ίδια πληροφορία με αυτή που θα είναι διαθέσιμη στον αποκωδικοποιητή. Δικαιολογείται έτσι η κατηγοριοποίηση της μεθόδου στην οικογένεια των κωδικοποιητών Analysis by Synthesis.

Και στην περίπτωση αυτή τα δείγματα της $d'(n)$ προκύπτουν από αποκωδικοποίηση παλαιότερων subframes που υλοποιείται στην Υπολειτουργία της Σύνθεσης (βλ. (δ)).

Υπολειτουργία (γ) - Σχηματισμός της ακολουθίας διέγερσης (excitation computation): Τα δείγματα των σφαλμάτων πρόβλεψης, $e(n)$, που δημιουργήθηκαν για κάθε subframe από την προηγούμενη διαδικασία υποδειματοποιούνται και κβαντίζονται. Η παραγόμενη ακολουθία αποτελεί το σήμα διέγερσης που θα σταλεί στο αποκωδικοποιητή. Αναλυτικότερα τα 40 δείγματα του σφάλματος πρόβλεψης $e(n)$ που αντιστοιχούν στο $j - th$ subframe φιλτράρονται από ένα FIR φίλτρο με κρουστική απόκριση $H(k)$ (weighting filter) που προσεγγίζει αδρά τη συμπεριφορά του ανθρώπινου αυτιού. Οι τιμές της κρουστικής απόκρισης και η διαδικασία φιλτραρίσματος περιγράφεται στην Ενότητα 3.1.18 του προτύπου.

Στη συνέχεια, από τα 40 δείγματα του παραγόμενου σήματος $x(n)$ εξετάζονται οι ακόλουθες υπακολουθίες μήκους

13

$$\begin{aligned} &x(k_0 + 40j + 0), x(k_0 + 40j + 3), \dots, x(k_0 + 40j + 36) \\ &x(k_0 + 40j + 1), x(k_0 + 40j + 4), \dots, x(k_0 + 40j + 37) \\ &x(k_0 + 40j + 2), x(k_0 + 40j + 5), \dots, x(k_0 + 40j + 38) \\ &x(k_0 + 40j + 3), x(k_0 + 40j + 6), \dots, x(k_0 + 40j + 39) \end{aligned}$$

και επιλέγεται αυτή με τη μεγαλύτερη ισχύ (έστω $x_M(i)$) σύμφωνα με την περιγραφή της Ενότητας 3.1.19 του ίδιου προτύπου. Η εν λόγω επιλογή καταγράφεται στην παράμετρο M_c με προφανείς τιμές 0, 1, 2 ή 3.

Στη συνέχεια υπολογίζεται η μέγιστη τιμή x_{max} της επιλεχθείσας υπακολουθίας και κβαντίζονται (i) η x_{max} και (ii) οι 13 λόγοι $x'(i) = x_M(i)/x'_{max}$ όπου x'_{max} είναι η αποκβαντισμένη εκδοχή της x_{max} . Η διαδικασία κβαντισμού προδιαγράφεται λεπτομερώς στην Ενότητα 3.1.20 του προτύπου.

Υπολειτουργία (δ) - Σύνθεση (Synthesis): Η υπολειτουργία αυτή είναι υπεύθυνη για την (ανα)σύνθεση των ακολουθιών $d'(n)$ που χρησιμοποιούνται στις υπολειτουργίες (α) και (β). Μιμείται απολύτως τη λειτουργία του αποκωδικοποιητή και παράγει την $d'(n)$ ως ακολούθως:

1. Υπολογίζει προσεγγίσεις των $x_M(i)$ από τη σχέση $x'_M(i) = x'_c(i) * x'_{max}$ όπου $x'_c(i)$ είναι η αποκβαντισμένη εκδοχή των $x'(i)$
2. Παρεμβάλλει μηδενικά σε κατάλληλες θέσεις της υπακολουθίας $x'_M(i)$ ανάλογα με την τιμή του M_c . Η ακολουθία, $e'(n)$, 40 δειγμάτων που προκύπτει θεωρείται η διέγερση του φίλτρου long term σύνθεσης.
3. Τα 40 δείγματα της ακολουθίας $d'(n)$ του τρέχοντος subframe υπολογίζονται μέσω της σχέσης:

$$d'(n) = e'(n) + b'd''(n) \quad (13)$$

$$d''(n) \equiv d'(n - N'), \quad (14)$$

όπου οι παράμετροι b' και N' είναι οι αποκβαντισμένες εκδοχές των b και N του τρέχοντος subframe. Σημειώστε ότι τα δείγματα $d'(n - N')$ που χρησιμοποιούνται στον αναδρομικό υπολογισμό ανήκουν με βεβαιότητα στα 3 προηγούμενα subframes καθώς ισχύει ότι $40 \leq N' \leq 120$. Συνεπώς είναι ήδη διαθέσιμα. (Τί γίνεται με τα τρία πρώτα subframes?).

2.1.4 Compose Frame

Η διαδικασία αυτή είναι υπεύθυνη για τη σύνθεση ενός binary block 260 bits που θα περιέχει το σύνολο της κωδικοποιημένης πληροφορίας ενός frame. Το πλήθος και η διάταξη των bits ανα παράμετρο φαίνεται στο Table 6.1b της σελίδας 47 του προτύπου.

2.2 Ο Αποκωδικοποιητής

Η διαδικασία αποκωδικοποίησης είναι αρκετά απλούστερη από αυτήν της κωδικοποίησης. Για κάθε frame επαναλαμβάνεται η ακόλουθη σειρά βημάτων:

1. Από τα 260 bits του αντίστοιχου binary block αναπαράγονται: (α) Μία φορά για ολόκληρο το frame οι αποκβαντισμένες τιμές των 8 συντελεστών LAR , (β) για κάθε subframe οι αποκβαντισμένες τιμές των παραμέτρων N' , b' , M' , x'_{max} και οι 13 τιμές των $x'_M(i)$.
2. Υπολογίζονται για κάθε subframe οι 40 τιμές των ακολουθιών $d'(n)$ ακριβώς όπως περιγράφηκε στην Υπολειτουργία Σύνθεσης του Κωδικοποιητή.
3. Υπολογίζονται οι αποκωδικοποιημένες τιμές των συντελεστών ανάκλασης από τις αντίστοιχες τιμές των $LAR(i)$ σύμφωνα με τις προδιαγραφές των Ενοτήτων 3.1.8 έως 3.1.10 του προτύπου (και εδώ η παρεμβολή που προδιαγράφεται στην Ενότητα 3.1.9 μπορεί να θεωρηθεί προαιρετική).
4. Από τους συντελεστές ανάκλασης υπολογίζονται οι συντελεστές πρόβλεψης a_k . Για λόγους ευκολίας θα σας δοθεί ο αντίστοιχος αλγόριθμος έτοιμος.
5. Σχηματίζεται η ακολουθία όλων των $d'(n)$ από τα 4 subframes (μήκους 160 δειγμάτων) και χρησιμοποιείται για να διεγείρει το φίλτρο short time synthesis

$$H_s(z) = \frac{1}{1 - \sum_{k=1}^8 a_k z^{-k}} \quad (15)$$

Η παραγόμενη έξοδος είναι η αποκωδικοποιημένη εκδοχή, $s'(n)$ του σήματος $s(n)$.

6. Ακολουθεί μετεπεξεργασία του σήματος $s'(n)$ αντίστροφη της πρώτης βαθμίδας του κωδικοποιητή που προδιαγράφεται στην Ενότητα 3.2.4 του προτύπου (εκεί χρησιμοποιείται το σύμβολο s_r αντί για s').

3 Διάρθρωση και Παραδοτέα της Εργασίας

Ο πυρήνας της εργασίας είναι η κατασκευή ενός κωδικοποιητή/αποκωδικοποιητή ενός frame φωνής της μορφής

```
frame_bit_stream: str, curr_frame_resd: np.ndarray = RPE_frame_coder(  
    s0: np.ndarray,  
    prev_frame_resd: np.ndarray  
)
```

και

```
s0: np.ndarray, curr_frame_resd: np.ndarray = RPE_frame_decoder(  
    frame_bit_stream: str,  
    prev_frame_resd: np.ndarray  
)
```

όπου:

- `s0`: ένα numpy array, που αναπαριστά ένα διάνυσμα από 160 δείγματα φωνής
- `prev_frame_resd`: οι τιμές της ακολουθίας $d'(n)$ μήκους 160 δειγμάτων, του προηγούμενου frame
- `frame_bit_stream`: τα 260 bits που αναπαριστούν το τρέχον frame.
Σημείωση: η αναπαράσταση του bitstream μπορεί για ευκολία να είναι ένα απλό string από τους χαρακτήρες '0', '1', είτε να χρησιμοποιεί τη βιβλιοθήκη `bitstream`¹, είτε κάποια άλλη υλοποίηση. Παρόλα αυτά, κάθε επιλογή πρέπει να περιγράφεται επαρκώς στην αναφορά σας.
- `curr_frame_resd`: οι τιμές της ακολουθίας $d'(n)$ μήκους 160 δειγμάτων, του τρέχοντος frame.

Για την επίδειξη της λειτουργίας των παραπάνω είναι αναγκαία η κατασκευή ενός προγράμματος-περίβλημα που διαβάζει/γράφει ανά τμήματα των 160 δειγμάτων αρχεία τύπου .wav.

Η κατασκευή των δύο παραπάνω συναρτήσεων πρέπει να γίνει με δομημένο τρόπο έτσι ώστε να είναι δυνατή και η αξιολόγηση των ενδιάμεσων βαθμίδων. Είναι μάλιστα δυνατή η τμηματική εκτέλεση της εργασίας εφόσον ακολουθηθούν οι παρακάτω σχεδιαστικές συμβάσεις.

3.1 Παραδοτέα

3.1.1 Παραδοτέο Επιπέδου 1: (1 μονάδα)

```
LARc: np.ndarray, curr_frame_st_resd: np.ndarray = RPE_frame_st_coder(  
    s0: np.ndarray,  
    prev_frame_st_resd: np.ndarray  
)
```

και

```
s0: np.ndarray = RPE_frame_st_decoder(  
    LARc: np.ndarray,  
    curr_frame_st_resd: np.ndarray  
)
```

¹<https://pypi.org/project/bitstream/>

όπου

- `s0`: ένα `numpy array`, που αναπαριστά ένα διάνυσμα από 160 δείγματα φωνής
- `prev_frame_st_resd`: οι τιμές της ακολουθίας $d'(n)$ μήκους 160 δειγμάτων, του προηγούμενου frame.
- `LARc`: Οι 8 κβαντισμένες-αποκβαντισμένες τιμές των συντελεστών LAR του τρέχοντος frame
- `curr_frame_st_resd`: οι τιμές της ακολουθίας $d'(n)$ μήκους 160 δειγμάτων, του τρέχοντος frame.

Η πρώτη από τις δύο συναρτήσεις υλοποιεί πλήρως τις διαδικασίες Preprocessing και Short Term Analysis όπως περιγράφηκαν στην προηγούμενη θεωρητική ενότητα για τον κωδικοποιητή. Η δεύτερη υλοποιεί την αντίστροφη ακριβώς διαδικασία όπως περιγράφηκε για τον αποκωδικοποιητή.

3.1.2 Παραδοτέο Επιπέδου 2: (+2 μονάδες)

```
LARc: np.ndarray,  
Nc: List[int],  
bc: List[int],  
curr_frame_ex_full: np.ndarray,  
curr_frame_st_resd: np.ndarray = RPE_frame_slt_coder(  
    s0: np.ndarray,  
    prev_frame_st_resd: np.ndarray  
)
```

και

```
s0: np.ndarray, curr_frame_st_resd: np.ndarray = RPE_frame_slt_decoder(  
    LARc: np.ndarray,  
    Nc: List[int],  
    bc: List[int],  
    curr_frame_ex_full: np.ndarray,  
    prev_frame_st_resd: np.ndarray  
)
```

όπου

- `curr_frame_st_resd`: ένα `numpy array` με τις τιμές της ακολουθίας $d'(n)$, μήκους 160 δειγμάτων, του τρέχοντος frame
- `curr_frame_ex_full`: ένα `numpy array` με τα 160 δείγματα του ολικού σφάλματος πρόβλεψης $e(n)$
- `prev_frame_st_resd`: οι τιμές της ακολουθίας $d'(n)$ μήκους 160 δειγμάτων, του προηγούμενου frame.
- `Nc`, `bc`: λίστες 4 στοιχείων, με την τιμή της εκάστοτε παραμέτρου για καθένα από τα 4 subframes του τρέχοντος frame.

και οι λοιπές μεταβλητές όπως αυτές του Επιπέδου 1.

Στην πρώτη από τις δύο παραπάνω υλοποιείται (σε εξέλιξη της αντίστοιχης του Επιπέδου 1) και μέρος της βαθμίδας Long Term Analysis και συγκεκριμένα οι Υπολειτουργίες (α), (β) και (δ). Παραλείπεται η Υπολειτουργία (γ) και για το λόγο αυτό θεωρούμε ότι η μεταβλητή `curr_frame_ex_full` περιέχει τις τιμές της ακολουθίας $e(n)$ και των 4 subframes χωρίς κβαντισμό και χωρίς κωδικοποίηση.

Η δεύτερη από τις παραπάνω συναρτήσεις υλοποιεί την αντίστροφη διαδικασία (αποκωδικοποίηση).

Απαραίτητο συστατικό του Παραδοτέου 2 είναι επίσης η συνάρτηση

```
N: int, b: float = RPE_subframe_slt_lte(
    d: np.ndarray,
    prev_d: np.ndarray
)
```

που δέχεται τις εισόδους:

- d: ένα numpy array διάνυσμα (τυπικά μήκους 40)
- prev_d: ένα numpy array διάνυσμα (τυπικά μήκους 120)

Η συνάρτηση χειρίζεται τα διανύσματα d, prev_d αντίστοιχα προς τα σήματα $d(n)$ και $d'(n)$ της Υπολειτουργίας (α), κι εκτιμά τις παραμέτρους:

- N: το pitch period
- b: ο συντελεστής ενίσχυσης/απόσβεσης

Η `RPE_frame_slt_coder(...)` πρέπει να χρησιμοποιεί προφανώς την `RPE_subframe_lte(...)`.

3.1.3 Παραδοτέο Επιπέδου 3: (+1 μονάδα)

Οι συνολικοί κωδικοποιητές - αποκωδικοποιητές όπως περιγράφηκαν στην αρχή της Ενότητας 3.

3.2 Μορφή Παραδοτέων - Υποδείξεις

Σε κάθε περίπτωση τα παραδοτέα πρέπει να συνοδεύονται (α) από ένα κείμενο επεξήγησης του τρόπου υλοποίησης, (β) κατάλληλο πρόγραμμα επίδειξης της διαδικασίας κωδικοποίησης / αποκωδικοποίησης ώστε να είναι άμεσα δυνατός ο έλεγχος ορθής λειτουργίας, (γ) οδηγίες για τον τρόπο κλήσης του προγράμματος επίδειξης.

Κάθε παραδιδόμενη εργασία θα πρέπει να δηλώνει ρητά το επίπεδο παραδοτέων που στοχεύει. Παραδοτέα του Επιπέδου 2 (3) δεν θα αξιολογηθούν αν δεν εκτελούνται με επιτυχία αυτά του Επιπέδου 1 (1 και 2).

Τα παραδοτέα προγράμματα θα πρέπει να είναι γραμμένα σε Python (οποιαδήποτε up-to-date version 3+ είναι αποδεκτή) προκειμένου να υποστηρίζονται με βοηθητικές συναρτήσεις που θα σας δοθούν στο αρχείο `hw_utils.py` (για τη μετατροπή των συντελεστών πρόβλεψης σε συντελεστές ανάκλασης και αντίστροφα).

Υπόδειξη: Τα `.py` αρχεία της υλοποίησής σας, μπορεί να είναι τα εξής:

- `encoder.py`: Όλες οι συναρτήσεις που σχετίζονται με διάφορα βήματα της κωδικοποίησης
- `decoder.py`: Όλες οι συναρτήσεις που σχετίζονται με διάφορα βήματα της αποκωδικοποίησης
- `utils.py`: Τυχόν επιπλέον helper συναρτήσεις που ενδεχομένως υλοποιήσετε και καλείτε σε διάφορα άλλα σημεία
- `demo1.py`, `demo2.py`, ...: Τα αρχεία επίδειξης της σωστής λειτουργίας των υλοποιήσεών σας.
- `hw_utils.py`: Το δοθέν αρχείο με τις βοηθητικές συναρτήσεις.

Υπόδειξη: για τη διαδικασία του FIR Filtering, μπορείτε να χρησιμοποιήσετε τις μεθόδους του module `scipy.signal` της βιβλιοθήκης `scipy`. Επίσης, για την ανάγνωση αρχείων `.wav` σε `numpy array`, μπορείτε να χρησιμοποιήσετε οποιαδήποτε έτοιμη υλοποίηση/βιβλιοθήκη επιθυμείτε (κάθε επιλογή όμως, πρέπει να τεκμηριώνεται στην αναφορά σας).

4 Για την υποβολή της εργασίας

Παραδώστε μία αναφορά με τις περιγραφές και τα συμπεράσματα που σας ζητούνται στην εκφώνηση.

Ο κώδικας θα πρέπει να είναι σχολιασμένος ώστε να είναι κατανοητό τι ακριβώς λειτουργία επιτελεί (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλίσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται και να υπολογίζει τα σωστά αποτελέσματα για *οποιαδήποτε* είσοδο πληροί τις υποθέσεις της εκφώνησης, και όχι μόνο για το συγκεκριμένο αρχείο που σας δίνεται.

Απαραίτητες προϋποθέσεις για την βαθμολόγηση της εργασίας σας είναι ο κώδικας να εκτελείται χωρίς σφάλμα (μόνο demos που εκτελούνται επιτυχώς θα βαθμολογηθούν), καθώς και να τηρούνται τα ακόλουθα:

- Υποβάλετε ένα και μόνο αρχείο, τύπου zip.
- Το όνομα του αρχείου πρέπει να είναι `AEM1_AEM2.zip`, όπου AEM είναι τα τέσσερα ψηφία του Α.Ε.Μ. του κάθε φοιτητή της ομάδας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα Python και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα `report.pdf`.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου UTF-8, και να έχουν κατάληξη `m`.
- Το αρχείο τύπου zip που θα υποβάλετε δεν πρέπει να περιέχει κανέναν φάκελο.
- Μην υποβάλετε το αρχείο ήχου που σας δίνεται για πειραματισμό.
- Μην υποβάλετε αρχεία που δεν χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ `"Thumbs.db"`, `".DS_Store"`, `".directory"`.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ `"#"`, `"$"`, `"%"` κλπ.