Κωδικοποίηση Φωνής: Πρότυπο ETSI GSM 06.10

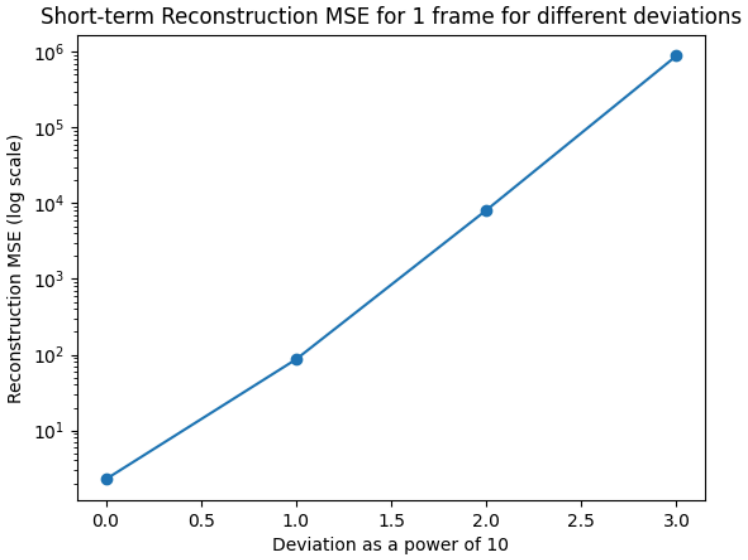
Λάτσκου Αφροδίτη 10433 afrolats@ece.auth.gr
Παλάσκα Αιμιλία 10453 aimiliapm@ece.auth.gr

Το πρότυπο ETSI GSM 06.10 είναι ένας αλγόριθμος κωδικοποίησης φωνής που χρησιμοποιείται για τη συμπίεση και αποκωδικοποίηση ηχητικών δεδομένων. Στην παρούσα εργασία, υλοποιήθηκαν και δοκιμάστηκαν demo εφαρμογές για την κωδικοποίηση και αποκωδικοποίηση φωνής σύμφωνα με το πρότυπο **και για τα τρία παραδοτέα**. Τα demo μπορούν να κληθούν τοπικά (στον φάκελο που ήδη βρίσκεται ο κώδικας) είτε μέσω ενός IDE είτε από την κονσόλα, ενώ όταν πρόκειται να αποθηκεύσουν δεδομένα (διαγράμματα, αρχεία ήχου) αυτό γίνεται τοπικά. Να σημειωθεί ότι όλες οι γραφικές παράχθηκαν με την βιβλιοθήκη matplotlib της python και επίσης ότι τα αρχεία **hw_utils.py** και **utils.py** περιέχουν γενικές βοηθητικές συναρτήσεις. Ο πηγαίος κώδικας του project, τα διαγράμματα και τα αρχεία ήχου είναι διαθέσιμα και στο GitHub repository.

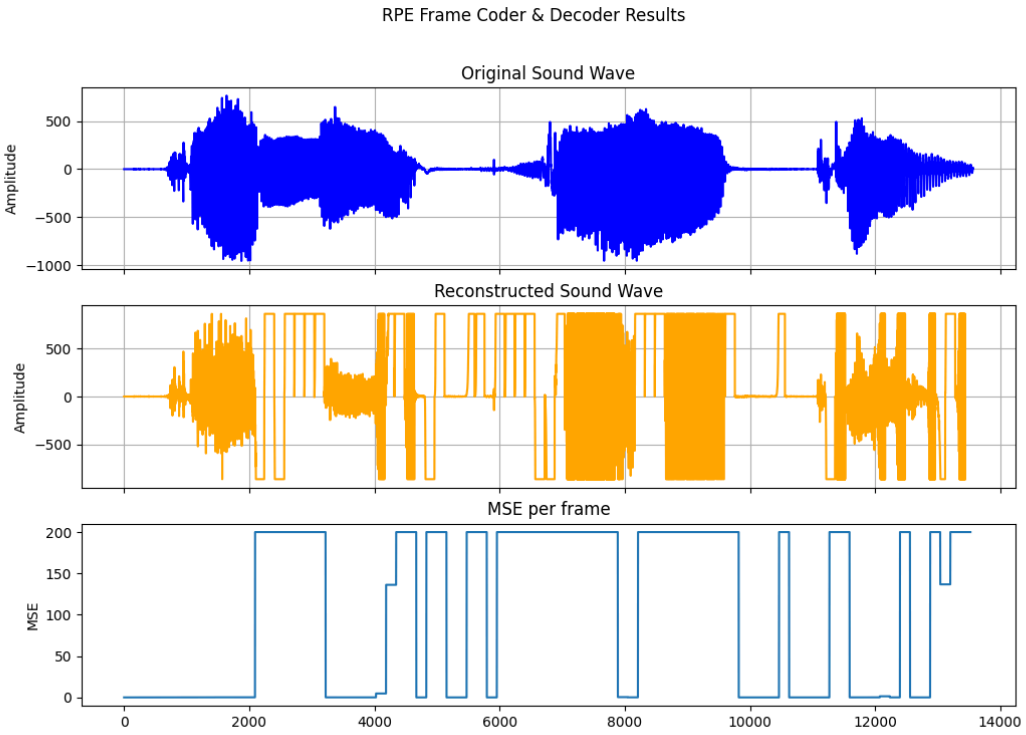
Pre-processing & Short-Term Analysis

Τα αρχεία **encoder.py** και **decoder.py** περιλαμβάνουν την κύρια υλοποίηση των παραδοτέων, τα οποία έχουν χωριστεί σε μικρότερες συναρτήσεις για τη βελτίωση της αναγνωσιμότητας και την ευκολότερη διαδικασία debugging. Συγκεκριμένα, τα αρχεία **utils_decoder.py** και **utils_encoder.py** περιέχουν βοηθητικές συναρτήσεις και παραμέτρους, οι οποίες συνοδεύονται από κατάλληλα σχόλια για τη διευκόλυνση της κατανόησης και της επεκτασιμότητας του κώδικα.

Στο πλαίσιο του unit testing, χρησιμοποιήθηκαν τα **demo1.py** και **demo2.py** για τη δοκιμή των συναρτήσεων *RPE_frame_st_coder* και *RPE_frame_st_decoder*, με την εφαρμογή κατάλληλων assert statements για την επαλήθευση του μεγέθους εξόδου σε ένα τυχαία παραγόμενο frame. Στο **demo3.py**, εξετάστηκε η επίδραση της απόκλισης (deviation) του σήματος εισόδου στο MSE της ανακατασκευής για ένα frame. Αυξήθηκε επαναληπτικά η απόκλιση σε λογαριθμική κλίμακα, αποδεικνύοντας εκθετική αύξηση του μέσου τετραγωνικού σφάλματος ανακατασκευής, όπως καταδεικνύεται στο διάγραμμα.



Στο **demo4.py**, πραγματοποιήθηκε ανακατασκευή του ηχητικού κομματιού **ena_dio_tria.wav**, εφαρμόζοντας ολόκληρο το pipeline κωδικοποίησης και ανακατασκευής. Προστέθηκε απλοϊκή κανονικοποίηση του εύρους του ηχητικού σήματος, χωρίς να αλλάζει η κατανομή των τιμών, και εφαρμόστηκε κλιπάρισμα για την αποφυγή ακραίων τιμών. Εκτός από την αποθήκευση του ανακατασκευασμένου ήχου στο αρχείο **short_term_reconstructed.wav**, δημιουργήσαμε και ένα διάγραμμα που απεικονίζει τα δύο σήματα και το MSE ανά frame.Σημειώνεται ότι υπάρχει σημαντικός θόρυβος στα σημεία της ομιλίας, κάτι αναμενόμενο λόγω του short-term analysis, όπου οι απότομες αλλαγές της φωνής επηρεάζουν αρνητικά την απόδοση της κωδικοποίησης και ανακατασκευής.



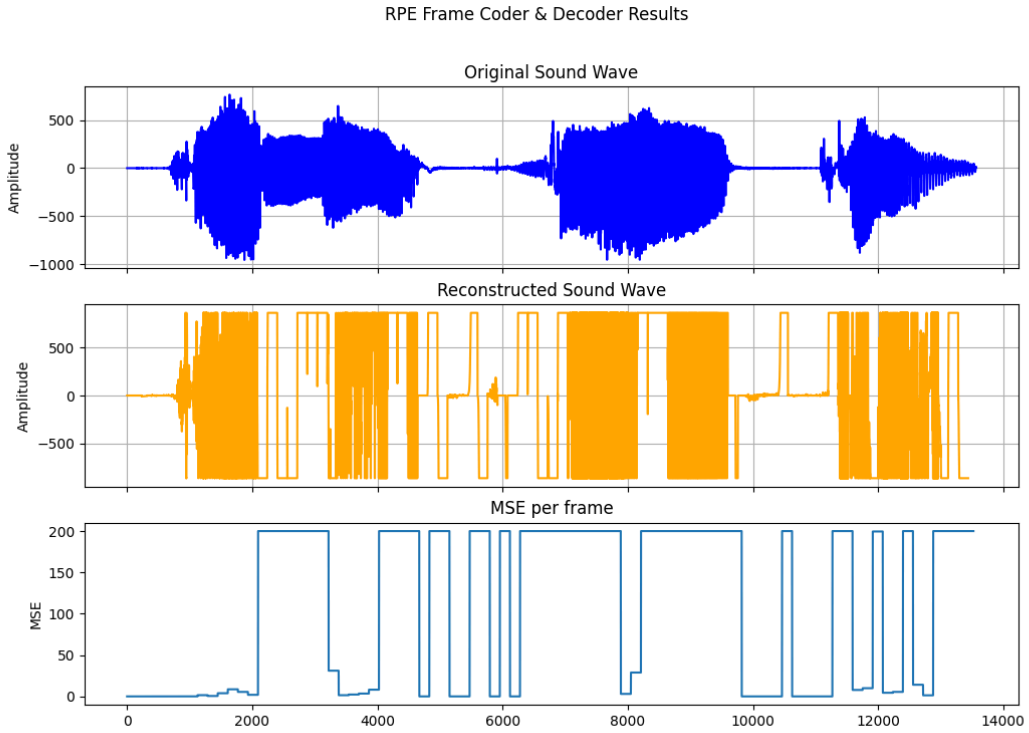
Long-Term Analysis

Στο πλαίσιο της διαδικασίας (απο)κωδικοποίησης, το Long-Term Analysis χρησιμοποιείται στην εκτίμηση και μοντελοποίηση της περιοδικότητας του ηχητικού σήματος. Η υλοποίηση του βασίζεται στις συναρτήσεις *RPE_frame_slt_coder*, *RPE_frame_slt_decoder* και *RPE_subframe_slt_lte*, οι οποίες επιτελούν τον καθορισμό του pitch (N) και της παραμέτρου gain (b). Η διαδικασία αυτή περιλαμβάνει την κβάντιση των παραμέτρων και, τέλος, την ανακατασκευή του residual για την αποστολή των δεδομένων.

Η συνάρτηση *RPE_subframe_slt_lte* λειτουργεί βοηθητικά στον κωδικοποιητή, προσδιορίζοντας το pitch και το gain σε επίπεδο subframe (40 δείγματα). Αυτό βασίζεται στη σύγκριση των τιμών αυτοσυσχέτισης του subframe με τα 120 προηγούμενα δείγματά του. Η ορθότητα της συνάρτησης αυτής επαληθεύεται μέσω του **demo5.py**. Σε κατώτερο επίπεδο κλήσης και διαστάσεων, η συνάρτηση *RPE_frame_slt_coder* ελέγχεται μέσω του **demo6.py**, ενώ η συνάρτηση *RPE_frame_slt_decoder* στο **demo7.py**.

Γίνεται επίσης έλεγχος της αλληλεπίδρασης των δύο συναρτήσεων (end-to-end) μέσα από το **demo8.py** και τέλος, το στάδιο της κατασκευής του ηχητικού κομματιού που δίνεται πραγματοποιείται μέσω του **demo9.py**, το οποίο παράγει το αρχείο

long_term_reconstructed.wav και εφαρμόζει, αντίστοιχα με το Short-Term Analysis, κανονικοποίηση και κλιπάρισμα. Οι αντίστοιχες κυματομορφές συγκρίνονται μέσω του παρακάτω διαγράμματος, ενώ παρουσιάζεται και το μέσο τετραγωνικό σφάλμα (MSE) ανά frame.



Αξίζει να σημειωθεί ότι η συγκεκριμένη ανακατασκευή διαφοροποιείται από το πλήρες σύστημα κωδικοποίησης και αποκωδικοποίησης, καθώς δεν περιλαμβάνει τη δημιουργία και μεταφορά των δεδομένων σε μορφή bitstream, διαδικασία που εξετάζεται στο τρίτο παραδοτέο. Παρατηρήθηκε ότι η μέθοδος Long-Term Analysis εισάγει μεγαλύτερο θόρυβο σε σύγκριση με το Short-Term Analysis, με αποτέλεσμα την αυξημένη παραμόρφωση του ηχητικού σήματος. Υποθέτουμε ότι η χρήση λεπτομερέστερων επιπέδων κβάντισης θα μπορούσε να μετριάσει το φαινόμενο, ωστόσο, δεν προβλέπεται στο πρότυπο που υλοποιήσαμε.

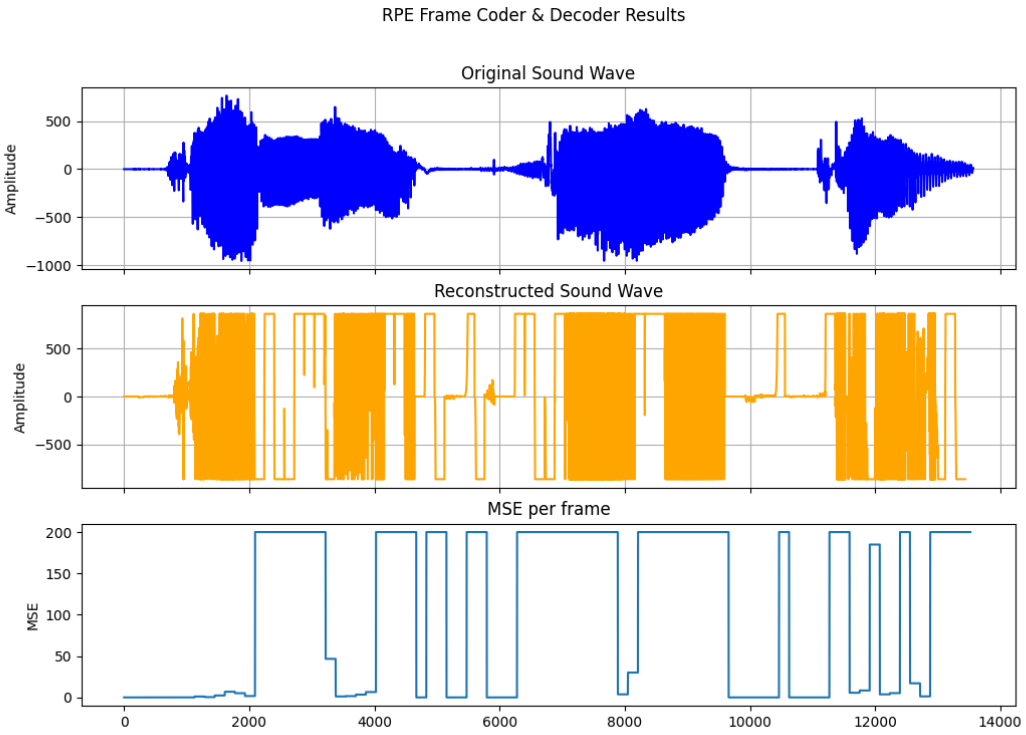
Πλήρης (Απο)Κωδικοποιητής

Για την ολοκλήρωση τόσο του κωδικοποιητή όσο και του αποκωδικοποιητή, υλοποιήθηκαν οι συναρτήσεις *RPE_frame_st_coder* και *RPE_frame_st_decoder*, στις οποίες χρησιμοποιούνται οι συναρτήσεις των προηγούμενων επιπέδων και επιπλέον συντίθεται και αναπαράγεται αντίστοιχα το binary block για την επικοινωνία μεταξύ coder και decoder σύμφωνα με τα standard του GSM 06.10. Για τη δημιουργία του bitstream χρησιμοποιήθηκε η βιβλιοθήκη της python bitstring.

Με δεδομένο ότι παραλείπεται η υπολειτουργία (γ) στο Long-Term Analysis, και δε δημιουργείται ποτέ η ακολουθία $x'_M(i)$, θεωρήσαμε καλύτερη η επιλογή της απευθείας μετάδοσης της ακολουθίας *curr_frame_ex_full* των 160 δειγμάτων, και επομένως το τελικό μέγεθος του bitstream ανέρχεται σε 552 bits. Αυτό δεν συνάδει πλήρως με τις προδιαγραφές του προτύπου GSM 06.10 καθώς αυτό απαιτεί bitstream μεγέθους 260 bits, αλλά το

θεωρούμε κατά σύμβαση με τις προδιαγραφές της εργασίας (λειτουργία χωρίς μείωση των bits κωδικοποίησης σφάλματος).

Στο **demo10.py** γίνεται έλεγχος της πλήρης λειτουργίας του κωδικοποιητή και αποκωδικοποιητή (end-to-end) και στο **demo11.py** γίνεται ανακατασκευή του ίδιο σήματος ήχου, χρησιμοποιώντας τις ίδιες τεχνικές pre- και post- processing (κανονικοποίηση και κατάλληλο clipping) για να έχουμε όσο το δυνατόν πιο ομαλά αποτελέσματα. Το ανακατασκευασμένο σήμα αποθηκεύεται με όνομα **full_codec_reconstructed.wav**. Παρακάτω φαίνονται οι κυματομορφές του πραγματικού και ανακατασκευασμένου σηματος. Επιπροσθέτως, δημιουργήθηκε και η γραφική παράσταση του MSE για κάθε frame



Παρατηρούμε ότι τα αποτελέσματα παρέμειναν σχεδόν αμετάβλητα μετά την υλοποίηση του Long-Term Analysis. Αυτό είναι επιθυμητό, καθώς η μόνη νέα προσθήκη αφορά την επικοινωνία μεταξύ κωδικοποιητή και αποκωδικοποιητή μέσω του παραγόμενου bitstream. Επομένως, θεωρούμε ότι αυτή η λειτουργία υλοποιήθηκε πλήρως.

Βιβλιογραφία

- European Telecommunications Standards Institute (ETSI). (1992). **GSM 06.10**: Full Rate Speech Transcoding. ETSI Technical Specification GSM 06.10 version 3.2.0. Retrieved from here.
- SciPy Community. (2024). **SciPy v1.11.4** Reference Guide. link
- Scott Griffiths. (2024). **Bitstring v4.0.1** Documentation. link
- Matplotlib Development Team. (2024). **Matplotlib v3.8**. link
- NumPy Developers. (2024). **NumPy v1.26.2** Manual. link
- Μέρη του κώδικα και της αναφοράς βελτιώθηκαν από **γλωσσικά μοντέλα** (ChatGPT 4, DeepSeek), αλλά η συνολική υλοποίηση παραμένει προϊόν προσωπικής πνευματικής δουλειάς.