

ΕΜΠ

ΣΗΜΜΥ

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΩΝ

2^η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΑΚ.ΕΤΟΣ : 2014-2015

ΠΑΡΑΔΟΤΕΑ: 7-3-15

ΟΝΟΜΑ:

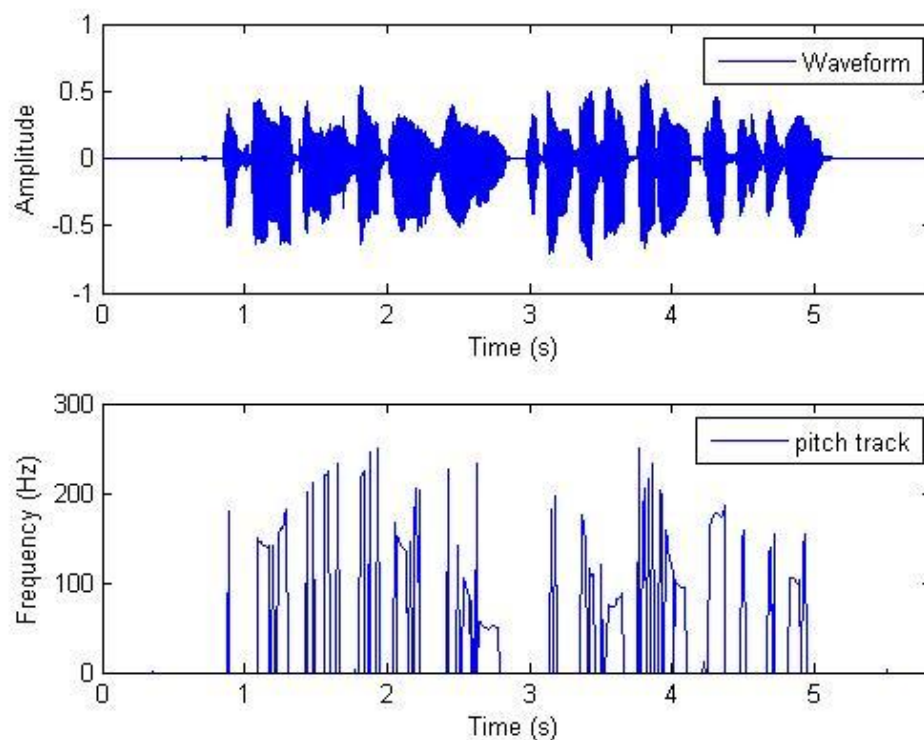
ΚΑΛΑΒΡΥΤΙΝΟΣ ΧΑΡΑΛΑΜΠΟΣ ΑΜ: 3112024

ΘΕΜΑ:

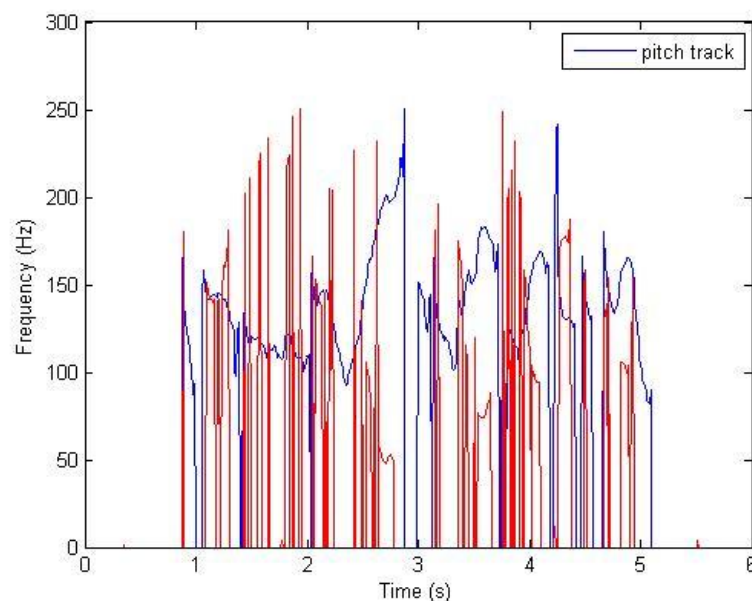
«Γραμμική πρόβλεψη (LPC) και ομομορφική (Cepstrum) επεξεργασία σημάτων με Matlab και εφαρμογές στη Σύνθεση και Συμπίεση φωνής»

ΜΕΡΟΣ 1. Εισαγωγή Pitch φωνής με χρήση Cepstrum.

Στο πρώτο μέρος στόχος είναι να εξαχθεί η θεμελιώδης συχνότητα φωνής με χρήση ομοιορφικής επεξεργασίας σημάτων. Παρακάτω φαίνεται το pitch που εξάχθηκε από το αρχείο εισόδου χρησιμοποιώντας την τεχνική **Cepstrum**, και συγκρίνεται με το pitch που δόθηκε στο αρχείο `pitch.mat` που εξάχθηκε χρησιμοποιώντας αλγόριθμο αυτοσυσχέτισης.



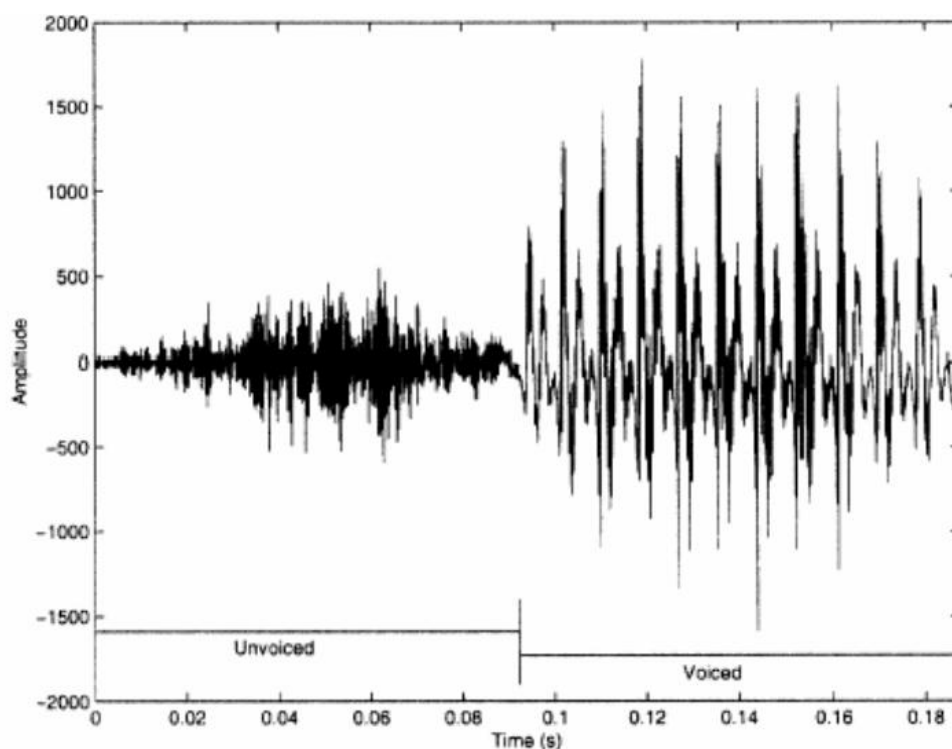
Σχ.1.1 Κυματομορφή φωνής εισόδου και pitch όπως εξήχθε από τον αλγόριθμο Cepstrum



Σχ.1.2 Σύγκριση μεταξύ των δύο pitch . Με κόκκινο είναι το pitch που εξαγάμουμε με τεχνική cepstrum, και με μπλε με τεχνική αυτοσυσχέτισης.

Στα παραπάνω σχήματα βλέπουμε τη θεμελιώδη συχνότητα **pitch** όπως εξάχθηκε από τον αλγόριθμο σε σύγκριση με την κυματομορφή του σήματος φωνής, και με το **pitch** όπως δόθηκε στην άσκηση. Παρατηρούμε ότι σαφώς και υπάρχουν αρκετές διαφορές στις δύο κυματομορφές, αλλά σε γενικές γραμμές βρίσκονται κοντά. Σίγουρα, δε θα μπορούσαμε να φτάσουμε σε απόλυτη ταύτιση των δύο κυματομορφών, αλλά βάζοντας ειδικό **threshold** και όρια για το **pitch** που γνωρίζουμε ότι πρέπει να είναι ανδρικής φωνής, καταλήγουμε στην κυματομορφή που φαίνεται πιο πάνω. Είναι γνωστό ότι το **pitch** για τους άνδρες ξεκινά από τα 50Hz και φτάνει κοντά στα 250Hz.

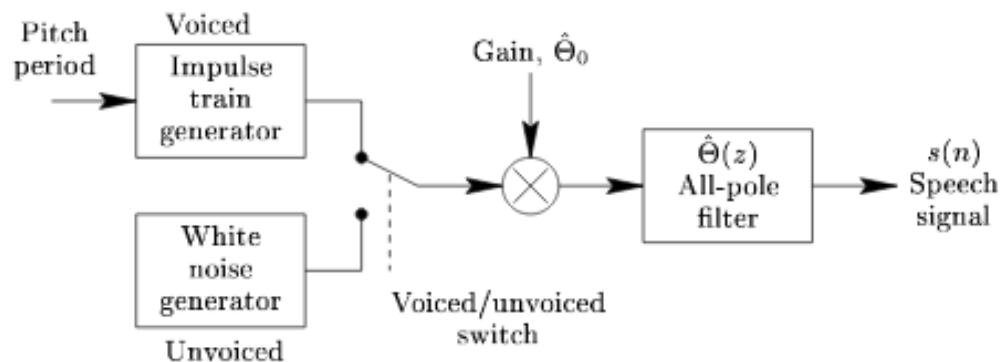
Κοιτώντας το Σχ.1 βλέπουμε ότι διακρίνουμε εύκολα τις έμφωνες από τις άφωνες περιοχές ομιλίας. Για την υλοποίηση των υπόλοιπων τμημάτων της άσκησης χρησιμοποιήθηκε το **pitch** όπως δόθηκε, και όχι αυτό που βρήκαμε στο πρώτο μέρος.



Σχ.2 Διαφορά ανάμεσα σε **voiced** και **unvoiced** περιοχές ομιλίας. Παρατηρούμε ότι οι άφωνες περιοχές δεν έχουν καμία περιοδικότητα, ενώ οι έμφωνες εμφανίζουν περιοδική συμπεριφορά.

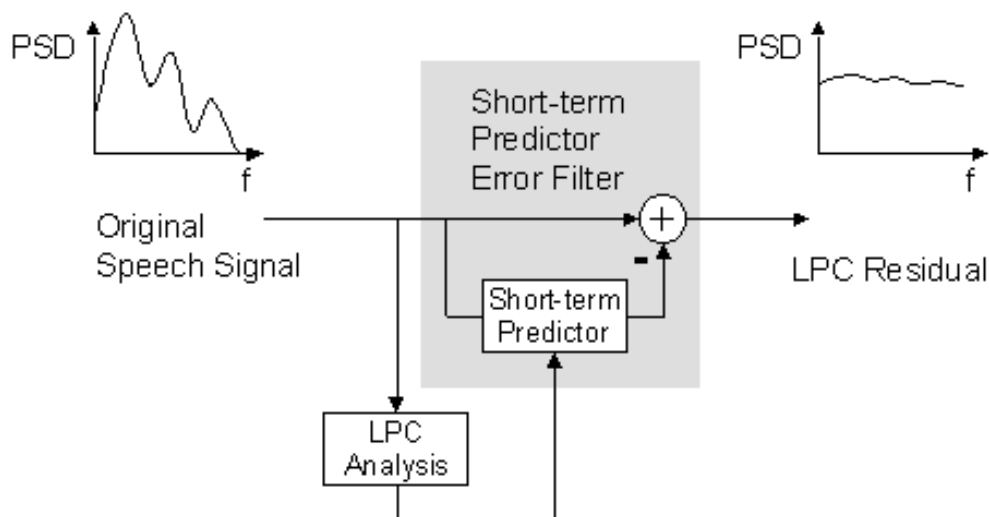
ΜΕΡΟΣ 2. Ψηφιακή σύνθεση φωνής με μέθοδο γραμμικής πρόβλεψης

Πρώτη μέθοδος ψηφιακής σύνθεσης που χρησιμοποιούμε είναι αυτή της **γραμμικής πρόβλεψης**. Η μέθοδος είναι αρκετά απλή στην κατανόηση και στην εφαρμογή της.



$$S(z) = E(z) \cdot \Theta(z) \quad \hat{\Theta}(z) = \frac{1}{\hat{A}(z)} = \frac{1}{1 - \sum_{i=1}^M \hat{a}(i)z^{-i}}$$

Η βασική ιδέα σύνθεσης φωνής χρησιμοποιώντας τη μέθοδο LPC φαίνεται πιο πάνω. Στόχος είναι κατά την ανάλυση να βρεθούν οι **συντελεστές γραμμικής πρόβλεψης** ακ, χρησιμοποιώντας τον αλγόριθμο **Levinson-Durbin**, και το κέρδος **G**. Αργότερα, οι συντελεστές αυτοί χρησιμοποιούνται για τη σύνθεση της φωνής από μια impulse generator βασισμένη στο pitch period. Στο φίλτρο Θ μπορούμε να αναφερόμαστε και ως *short-term predictor error filter*. Η LPC ανάλυση βασίζεται στον υπολογισμό των παραμέτρων για το **φίλτρο Θ** που είναι ένα **all-pole filter**. Για την ανάλυση και τη σύνθεση χωρίζουμε το αρχικό μας σήμα σε overlapping frames, και βρίσκουμε τις παραμέτρους του προβλέπτη σε κάθε frame. Ως τάξη προβλέπτη θέτουμε **$p=Fs(kHz)+4$** , που ισοδυναμεί σε τάξη 52, αφού η συχνότητα δειγματοληψίας είναι 48.000Hz. Η μεγαλύτερη τάξη προβλέπτη προσφέρει καλύτερη ακρίβεια στην ανάλυση και σύνθεση φωνής, αν και πολλά βιβλία αναφέρουν πως ένας προβλέπτης της τάξης $p=12-16$ τις περισσότερες φορές αρκεί. Τα παράθυρα (frames) που χωρίζουμε το σήμα μας έχουν διάρκεια 30ms (1440 samples) και τα overlapping frames έχουν μήκος 20ms (960 samples). Το χρονικό βήμα είναι λοιπόν 10ms. Για τη σωστή σύνθεση χρησιμοποιείται η μέθοδος overlap-add analysis στα παραθυροποιημένα frames.



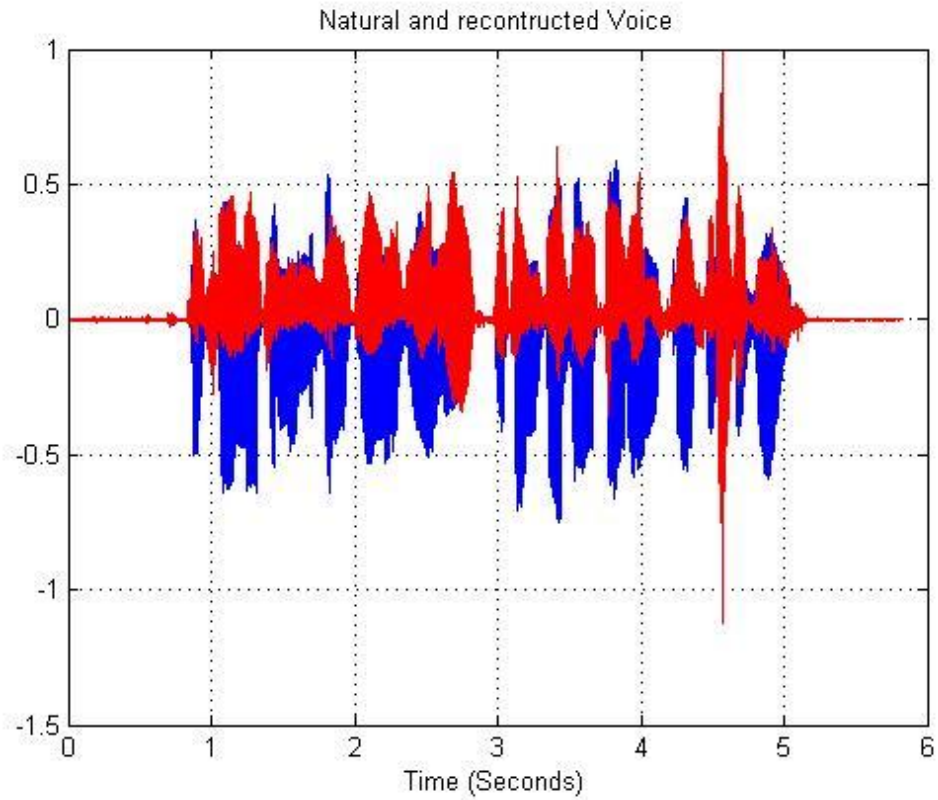
LPC Filter.

Σχ.3 Ως LPC residual αναφερόμαστε στο λάθος της πρόβλεψης από το αρχικό σήμα

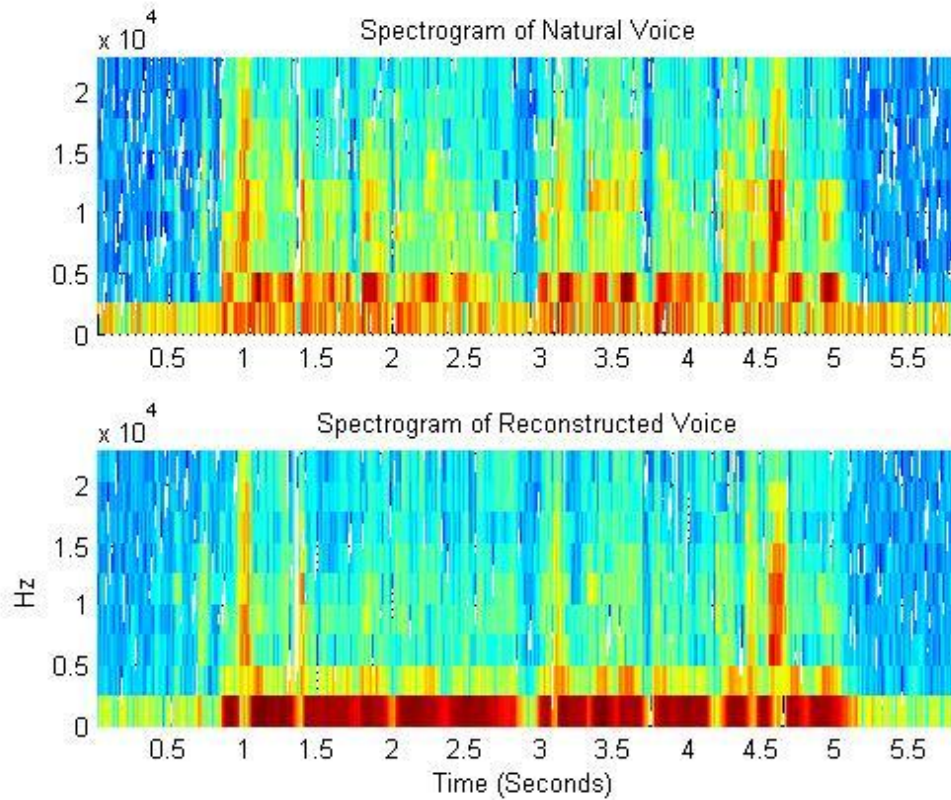
Χρησιμοποιώντας τις συναρτήσεις που μας δόθηκαν έτοιμες για αυτό το κομμάτι της άσκησης, προέκυψαν αρκετά προβλήματα. Συνεπώς, οι συναρτήσεις χρησιμοποιήθηκαν τροποποιημένες εν μέρει ή εξ ολοκλήρου για να εκτελεστεί επιτυχώς το 2^ο μέρος. Η ανάλυση και η σύνθεση γίνεται στην ίδια συνάρτηση. Για τη σύνθεση χρησιμοποιήθηκε το αντίστροφο φίλτρο **lpc (vocal tract filter)**.

Παρατηρούμε ότι η φωνή που προκύπτει από **LPC synthesis** αν και μονότονη και όχι τόσο καθαρή, είναι ικανοποιητικής ποιότητας για την αναγνώριση της φράσης. Μειώνοντας την τάξη πρόβλεψης παρατηρούμε ότι δεν υπάρχει μεγάλη διαφορά, αφού η τάξη είναι ήδη αρκετά μεγάλη. Μόνο αν φτάσει χαμηλότερα από τα 10 βλέπουμε διαφορά στα residuals και στην σύνθεση του σήματος φωνής.

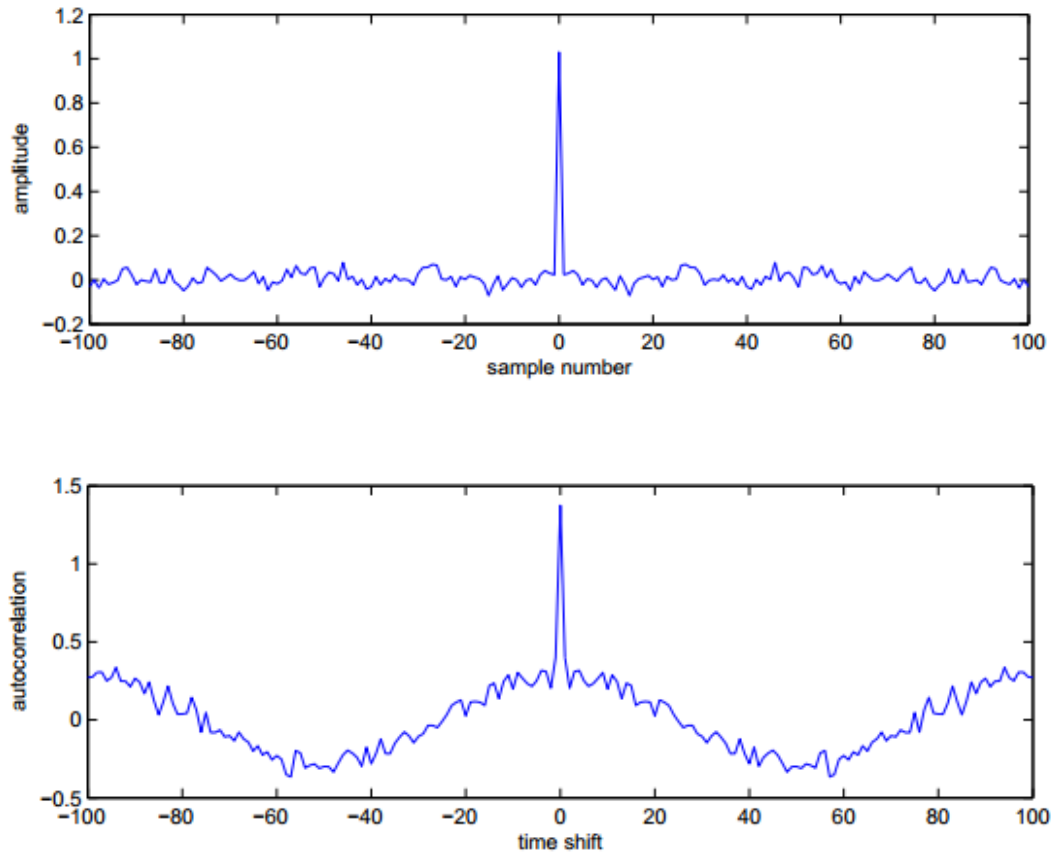
Ακούγοντας με τη ρουτίνα `sound()` του **matlab** το λάθος e1, παρατηρούμε ότι διακρίνουμε τη φράση του σήματος εισόδου, αν και αρκετά διαφορετική. Αυτό σημαίνει ότι το λάθος δεν είναι **λευκός θόρυβος**. Για να διακρίνουμε αν είναι ή όχι λευκός θόρυβος, μπορούμε να χρησιμοποιήσουμε την αυτοσυσχέτιση του λάθους, και από τη μορφή της να αποφασίσουμε. Παρακάτω φαίνονται οι αυτοσυσχετίσεις δύο διαφορετικών σημάτων. Το πρώτο είναι καθαρός λευκός θόρυβος, ενώ το 2^ο είναι ένα σήμα που περιέχει θόρυβο.



Σχ.4.1 Αρχικό σήμα φωνής (μπλε χρώμα) και σήμα μετά από σύνθεση LPC (κοκκινο χρώμα)



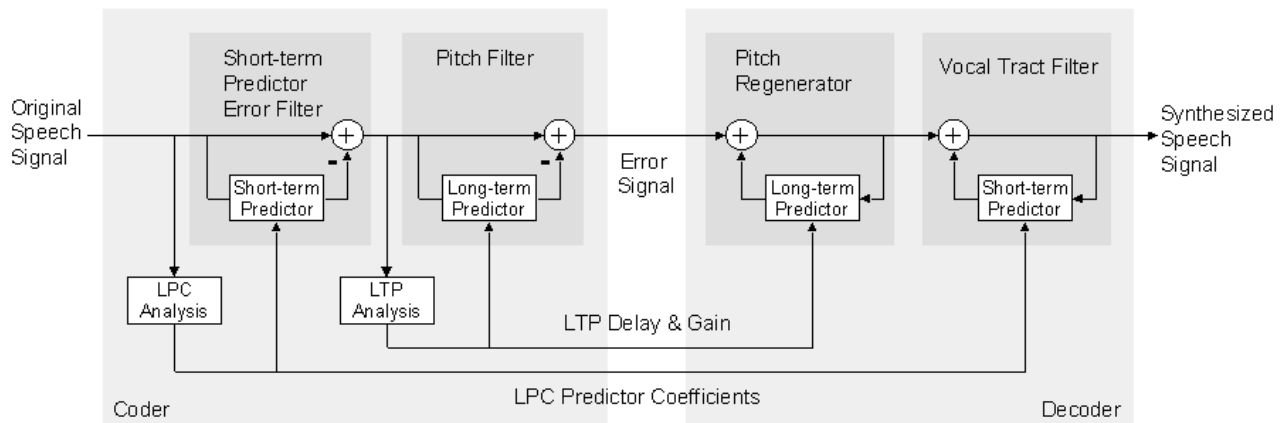
Σχ.4.2 Spectrogram of original and reconstructed voice using LPC



Σχ.4.3 Αυτοσυσχέτιση λευκού θορύβου (πάνω) και αυτοσυσχέτιση ημιτονοειδούς σήματος με λευκό θόρυβο (κάτω). Παρατηρούμε ότι ο λευκός θόρυβος επηρεάζει τη μορφή της αυτοσυσχέτισης, αλλά διακρίνεται και το κανονικό σήμα.

Βρίσκοντας την αυτοσυσχέτιση του σήματος λάθους (residual) και σχεδιάζοντας την κυματομορφή της, παρατηρούμε ότι αν και υπάρχει η χαρακτηριστική κορυφή στο 0, διακρίνονται και χαρακτηριστικά του αρχικού σήματος σε αυτή. Οπότε το σήμα λάθους περιέχει λευκό θόρυβο αλλά και το αρχικό σήμα σε παραμορφωμένη μορφή.

ΜΕΡΟΣ 3. Βελτιωμένη σύνθεση φωνής με χρήση μακροπρόθεσμης πρόβλεψης και κατάλληλα σχεδιασμένης βάσης διεγέρσεων CELP.



Adaptive Predictive Coding

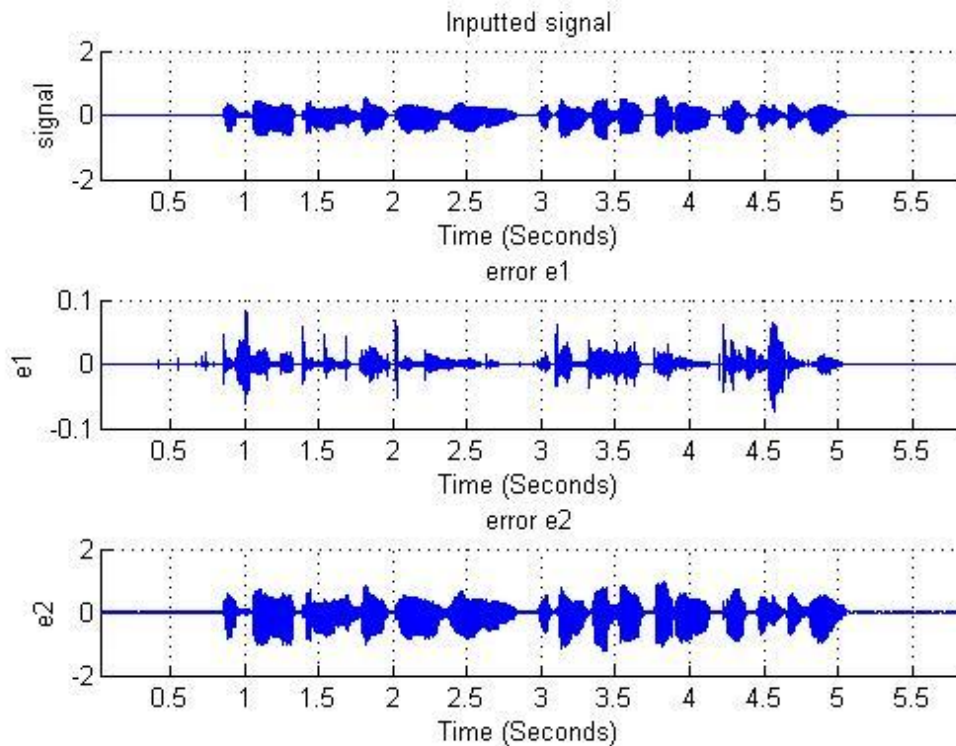
Adaptive Predictive coding using both short-term and long-term predictors

Στο 3^ο μέρος της άσκησης ζητείται να βελτιωθεί η προηγούμενη σύνθεση φωνής χρησιμοποιώντας ένα επιπλέον φίλτρο κατά την ανάλυση και το αντίστροφο κατά τη σύνθεση. Ο προβλέπτης βρίσκει την περιοδικότητα της φωνής, όπου αυτή υπάρχει, και ακολουθεί τον πρώτο προβλέπτη στην ανάλυση, όπως φαίνεται και στο προηγούμενο σχήμα. Το φίλτρο προβλέπτη μακράς καθυστέρησης μπαίνει, όπως φαίνεται στο σχήμα, μετά το **LPC φίλτρο** κατά την ανάλυση. Κατά τη σύνθεση, χρησιμοποιείται το αντίστροφο φίλτρο του **LTP πριν από το vocal tract filter**.

Συνεπώς, για την εύρεση του συντελεστή β και M , (LTP coefficients)

χρησιμοποιείται το λάθος του πρώτου προβλέπτη e_1 ενώ το λάθος του δεύτερου προβλέπτη το ονομάζουμε e_2 . Κατά τη σύνθεση εφαρμόζεται πρώτα το inverse LTP filter στα σήματα του impulse generator ($u_{generator}()$) πάνω σε κάθε παράθυρο του σήματος, και έπειτα ακολουθείται η ίδια διαδικασία που ακολουθήθηκε και προηγουμένως (**μέρος 2**).

Τα 2 λάθη e_1 , e_2 φαίνονται παρακάτω σε αντιπαράβολή με το αρχικό σήμα:

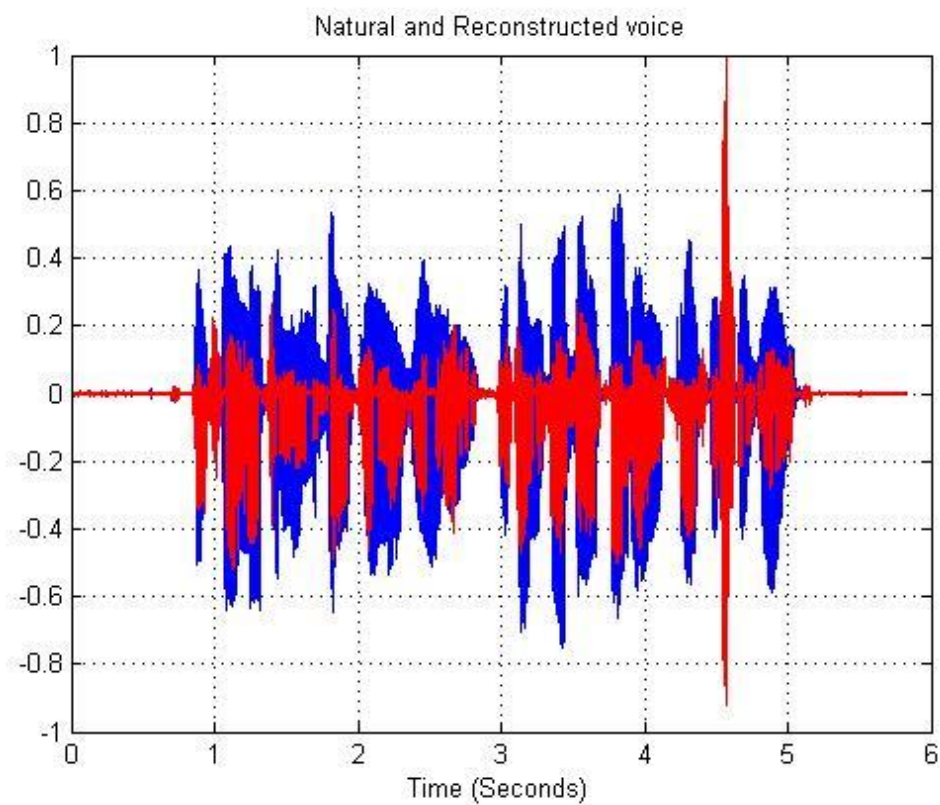


Σχ.5 Διάγραμμα σύγκρισης μεταξύ των predictors errors. Το e1 αντιστοιχεί στο λάθος του πρώτου προβλέπτη, ενώ το e2 στο λάθος της νέας πρόβλεψης έπειτα από την εφαρμογή του 2^{ου} φίλτρου

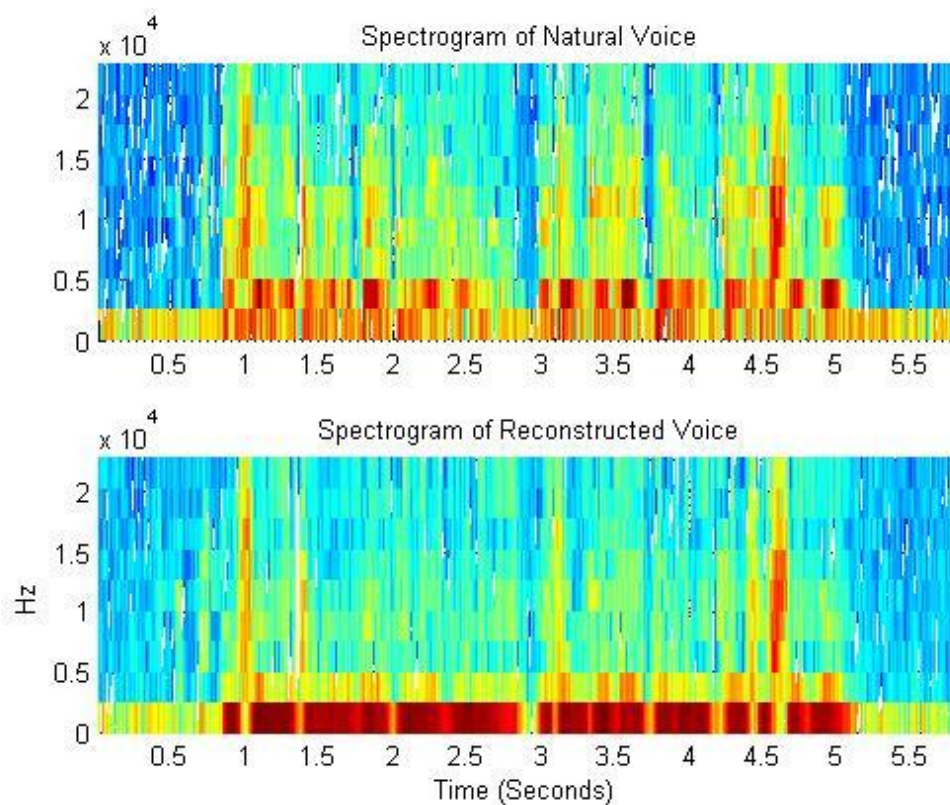
Παρατηρούμε πως το πρώτο λάθος πρόβλεψης είναι αρκετά μικρότερο από το 2^ο λάθος πρόβλεψης.

Παρακάτω φαίνεται και η σύνθεση φωνής χρησιμοποιώντας τον **long-term predictor**. Ακούγοντας τη σύνθεση φωνής, παρατηρούμε ότι υπάρχει μία βελτίωση όσον αφορά την ποιότητα της φωνής, που όμως δεν είναι αρκετά αισθητή. Στο πρώτο από τα επόμενα διαγράμματα (Σχ6.1) παρατηρούμε τη σύνθεση της φωνής σε αντιπαραβολή με το αρχικό σήμα. Το πλάτος της σύνθεσης είναι μικρότερα στα περισσότερα σημεία, με εξαίρεση μια ανωμαλία που προκύπτει στο 4,5 δευτερόλεπτο.

Η φωνή συνεχίζει να ακούγεται μονότονη και ηλεκτρονικά τροποποιημένη, χωρίς την αρχική της φυσικότητα.



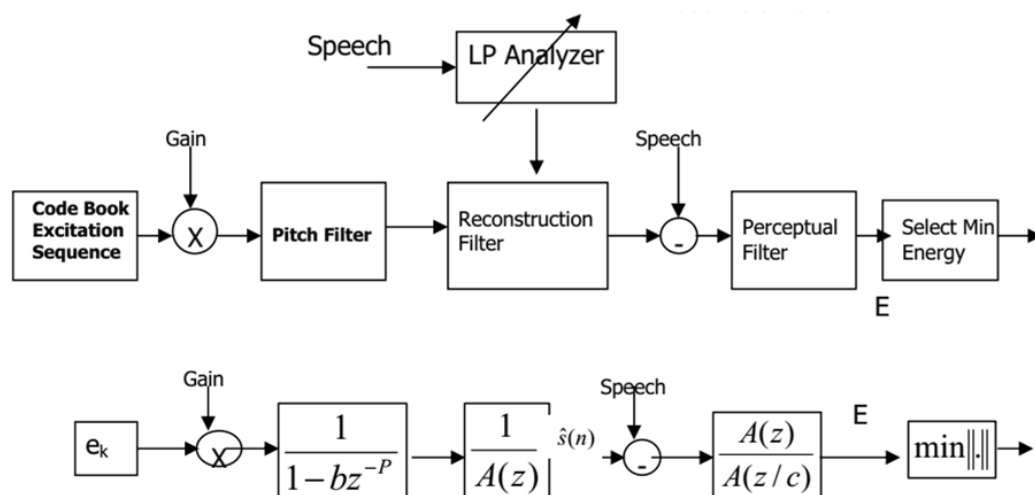
Σχ.6.1 Αρχικό σήμα φωνής (μπλε χρώμα) και σήμα μετά από σύνθεση LPC long (κοκκινο χρώμα)



Σχ.6.2 Spectrogram of original and reconstructed voice using LPC long

3.2 Δημιουργία βάσης τυχαίων διεγέρσεων και επιλογή της βέλτιστης για σύνθεση (CELP)

Σε αυτό το βήμα, προσπαθούμε να βελτιώσουμε την προηγούμενη σύνθεση, χρησιμοποιώντας **ένα πλήθος διαφορετικών διεγέρσεων u** , και επιλέγοντας αυτή που δίνει το μικρότερο λάθος σύνθεσης. Αυτό γίνεται βρίσκοντας την ενέργεια λάθους κάθε διεγέρσης, και επιλέγοντας τη διεγέρση που έχει τη μικρότερη.



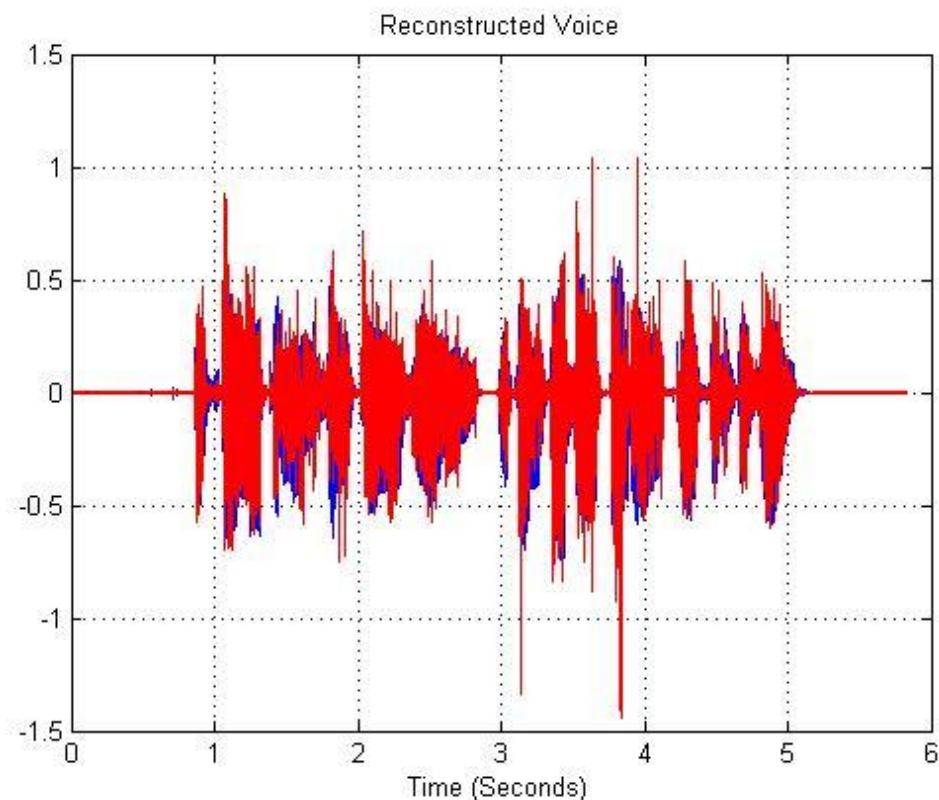
Σχ.7 Μοντέλο CELP που χρησιμοποιήθηκε. Στην περίπτωση μας το e_k ταυτίζεται με τις διεγέρσεις u που αποτελούν το codebook. Τα pitch και reconstruction filters είναι αυτά που υλοποιήσαμε προηγουμένως

Στο παραπάνω block διάγραμμα περιγράφεται η διαδικασία σύνθεσης **του CELP**, γνωστή και ως **analysis-by-synthesis** κωδικοποίηση. Παρατηρούμε ότι η διαφορά ανάμεσα στη συνθετική φωνή και στην αρχική περνά μέσα από **ένα perceptual weighting filter** της μορφής που φαίνεται παραπάνω. Το φίλτρο έχει σκοπό να δώσει έμφαση σε perceptual important συχνότητες και αργότερα ελαχιστοποιείται για να βελτιστοποιηθεί το **excitation signal** (διεγέρσεις). Για τον υπολογισμό όλων αυτών, χωρίζουμε κάθε frame στο οποίο κάνουμε ανάλυση σε blocks, ώστε να έχουμε περισσότερη πληροφορία για το excitation, και να μπορέσουμε να βρούμε έτσι το βέλτιστο για τη σύνθεση. Αυτό σημαίνει ότι το σήμα που χρησιμοποιείται για διεγερση του **LP synthesis filter** υπολογίζεται περισσότερες από μια φορές σε κάθε frame. Μια διεγέρση(excitation) επιλέγεται πρώτα από το codebook, το φίλτρο μακράς καθυστέρησης χρησιμοποιείται για εξαγωγή του pitch (γι αυτό αναφέρεται και ως *pitch regenerator* πολλές φορές), και έπειτα βρίσκουμε τις παραμέτρους που χρειαζόμαστε για να υπολογίσουμε το καλύτερο (βέλτιστο) σήμα διεγέρσης που θα χρησιμοποιήσουμε.

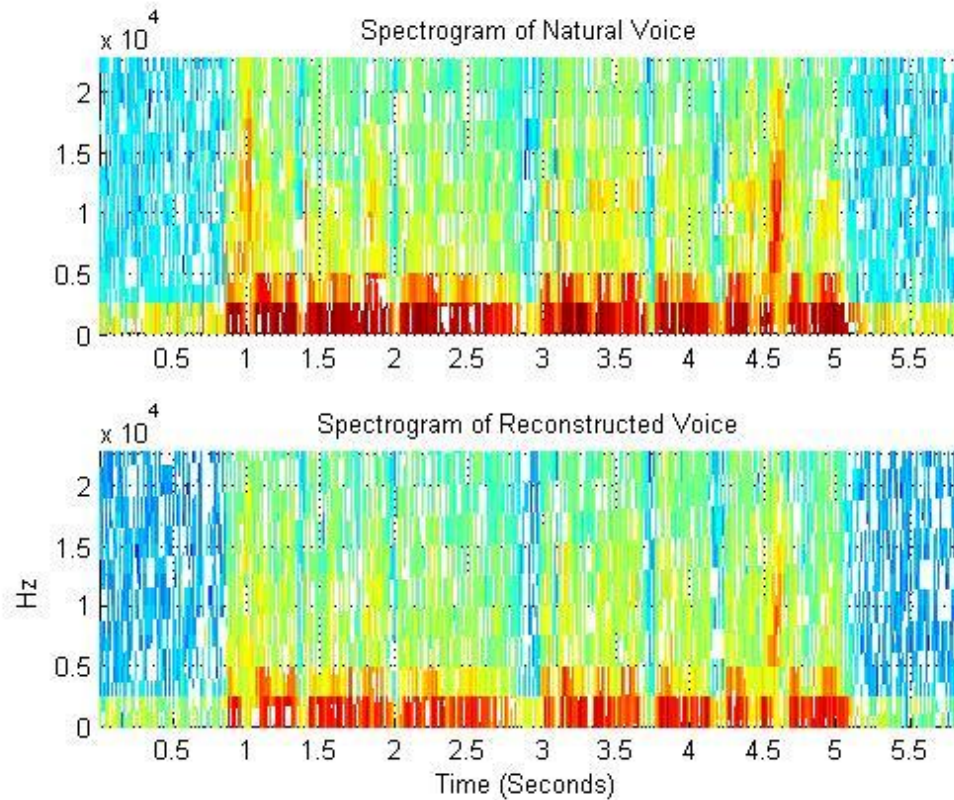
Οι παράμετροι που μας ενδιαφέρουν στη σύνθεση *celr* είναι όπως και πριν οι συντελεστές γραμμικής πρόβλεψης α , και το κέρδος G ή Θ_0 , οι παράμετροι β και P (ή M (καθυστέρηση *pitch*)) του φίλτρου πρόβλεψης μακράς καθυστέρησης και το k (*codebook index/vector*) που χρησιμοποιείται για την σύνθεση.

Οι διεγέρσεις είναι τις μορφές αθροίσματων συνημιτόνων, με πλάτος c_k που είναι Rayleigh καταναμημένο. Για τη δημιουργία των 1024 διεγέρσεων χρησιμοποιήσαμε τη συνάρτηση `raylrnd()` για παραγωγή των c_k και τυχαίες φάσεις ϕ_k με ευρος $[0, 2\pi]$.

Ο συντελεστής c του *perceptual weighting filter* έχεις εύρος $[0, 1]$. Στην περίπτωση αυτή επιλέγεται $c=0.8$. Όσον αφορά το *pitch filter* (ή *ltp filter*) για διευκόλυνση το περιορίζουμε σε ένα εύρος P_{idx} , που εξαρτάται από το πλάτος των *frames* που χρησιμοποιούμε. Επειδή το P μπορεί να είναι μεγαλύτερο από κάθε *block* (*sub frame*) που χρησιμοποιούμε, θα πρέπει να αποθηκεύουμε με κάποιον τρόπο προηγούμενες τιμές των *excitation signals* που υπολογίζουμε για τη σύνθεση της φωνής. Αυτός είναι ο λόγος που υπάρχουν κάποια *buffers* στο πρόγραμμα, που τα χρησιμοποιούμε σε περίπτωση που είναι $P > \text{frame_length}$.



Σχ.8.1 Αρχικό σήμα φωνής (μπλε χρώμα) και σήμα μετά από σύνθεση CELP (κόκκινο χρώμα)



Σχ.8.2 Spectrogram of original and reconstructed voice using CELP

Παρατηρούμε ότι το σήμα φωνής που προκύπτει από την μέθοδο CELP ακούγεται αρκετά καθαρά, σε σχέση με τα δύο προηγούμενα, αλλά δεν είναι απόλυτα φυσικό. Έχει χάσει όμως την μονοτονία των δύο προηγούμενων σημάτων («ρομποτική φωνή»).

ΜΕΡΟΣ 4. Κωδικοποίηση/ συμπίεση φωνής

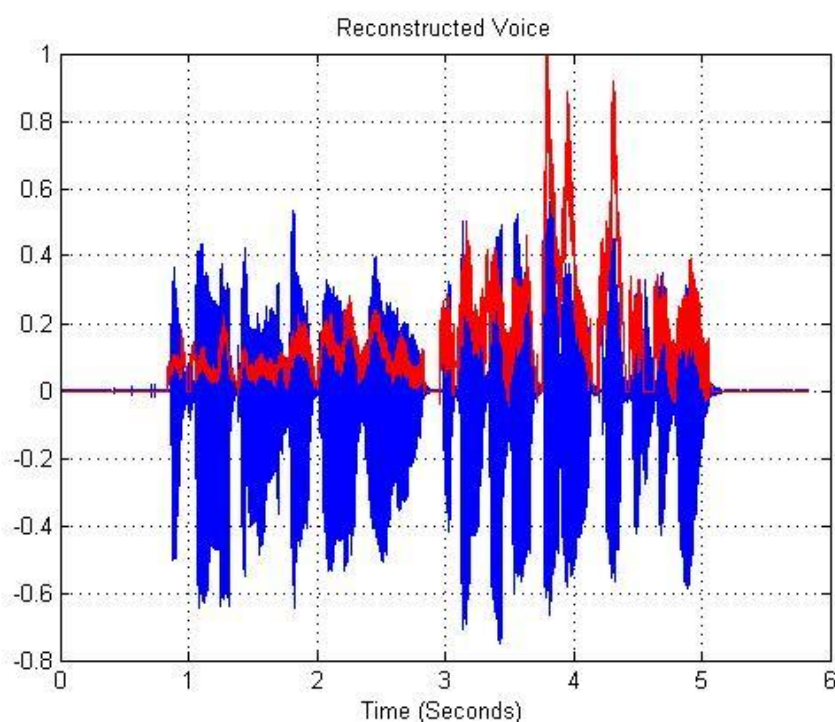
Στο τελευταίο μέρος της 2^{ης} άσκησης ασχολούμαστε με τη συμπίεση της φωνής. Για να γίνει επιτυχώς η συμπίεση φωνής, πρέπει να κβαντίσουμε τις παραμέτρους του *LPC synthesizer*. Σύμφωνα με την εκφώνηση για την κβάντιση των παραμέτρων χρησιμοποιούμε τα *log-area ratios* και όχι άμεσα τους συντελεστές γραμμικής πρόβλεψης α . Ο λόγος είναι ότι κατά την άμεση κβάντισή τους υπάρχει μεγάλη πιθανότητα μια μικρή μεταβολή στις τιμές, να αλλάξει πλήρως τη θέση των πόλων, με αποτέλεσμα να μη γίνει σωστά η σύνθεση.

$$g_i = \log \frac{1 - k_i}{1 + k_i}, i = 1, 2, \dots, M$$

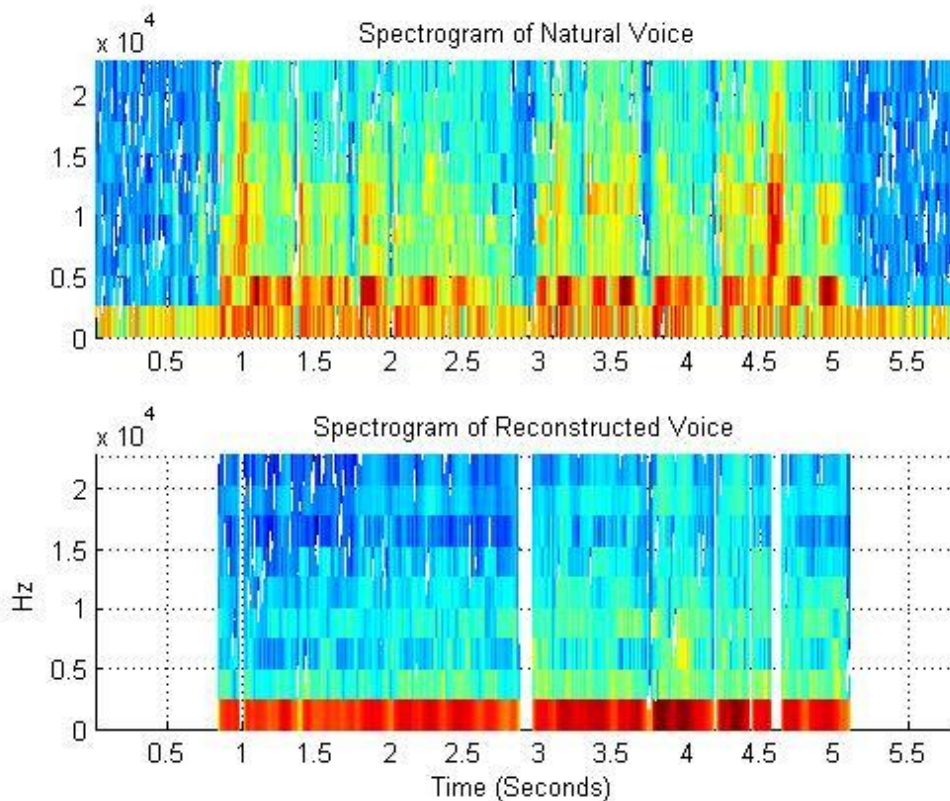
Όπου M η τάξη του προβλέπτη και k_i οι συντελεστές ανάκλασης (PARCOR ή reflection coefficients). Για να βρούμε τα LAR g_i χρησιμοποιήσαμε τη συνάρτηση `rc2lar()` και `lar2rc()` Για να βρούμε πάλι τα K , μετά την κβάντιση. Για να υπολογίσουμε τα α , από τους συντελεστές Κάππα, δημιουργήσαμε μια συνάρτηση με όνομα `rf2lpc()` που χρησιμοποιεί τον αλγόριθμο **Levinson-Durbin** για να υπολογίσει τους συντελεστές γραμμικής πρόβλεψης από τους συντελεστές ανάκλασης.

Η κβάντιση γίνεται μετά την ανάλυση και πριν από τη σύνθεση. Θεωρητικά, είναι αρκετά εύκολη η εφαρμογή της. Χρησιμοποιούμε τον προηγούμενο κώδικά μας και κβαντίζουμε τις τιμές πριν από τη σύνθεση.

Για την κβάντιση χρησιμοποιήσαμε τις συναρτήσεις `udencode()`, `uencode()` του Matlab, εισάγοντας τα bits που θέλαμε να κβαντίσουμε κάθε παράμετρο. Κάνοντας την κβάντιση για τη σύνθεση LPC βρήκαμε τα παρακάτω διαγράμματα

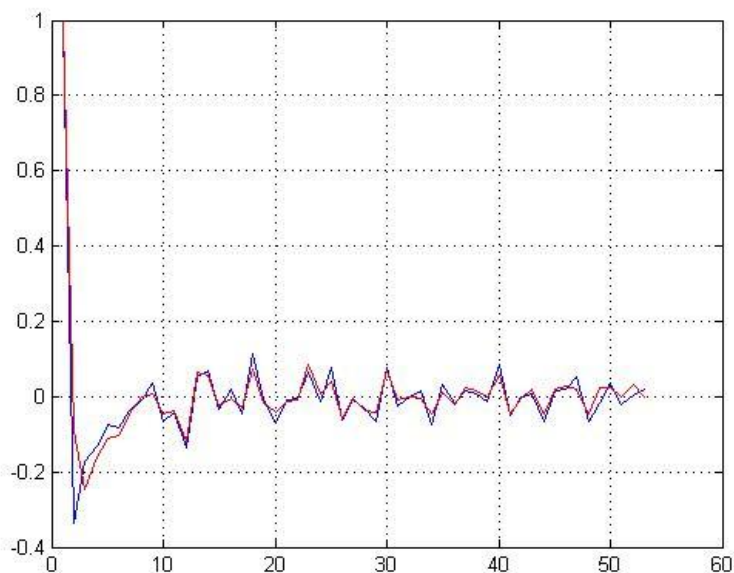


Σχ.9.1 Αρχικό σήμα φωνής (μπλε χρώμα) και σήμα μετά από σύνθεση LPC encoded (κόκκινο χρώμα)



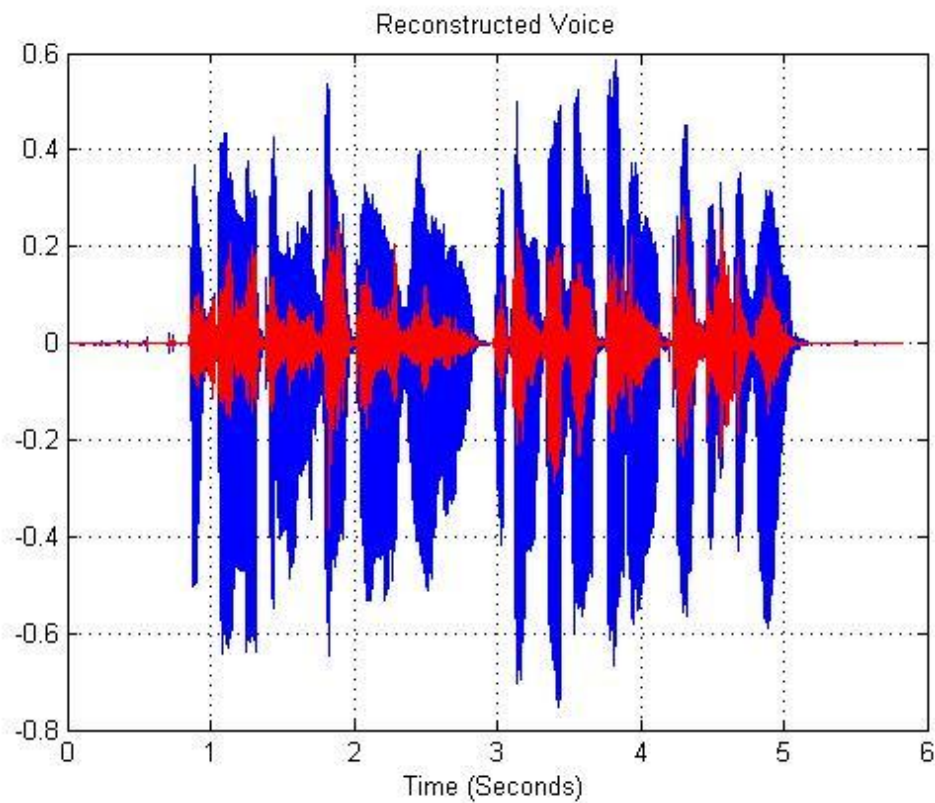
Σχ.9.2 Spectrogram of original and reconstructed voice using LPC encoded

Ακούγοντας τη συνθετημένη φωνή και παρατηρώντας τα δύο διαγράμματα συμπεραίνουμε ότι υπάρχει μεγάλη πιθανότητα να **υπάρχει λάθος!** στην κβάντιση. Το σήμα φωνής δεν ακούγεται όπως θα έπρεπε, πράγμα που φαίνεται και από το Σχ.9.1. Προσπαθώντας να διακρίνουμε το λάθος ελέγξαμε τις τιμές των κβαντισμένων παραμέτρων a , που όμως, όπως φαίνονται παρακάτω, φαίνοντια φυσιολογικές. Κατα πάσα πιθανότητα δεν υλοποιήσαμε σωστά την κβάντιση σε κάποια από τις άλλες παραμέτρους.

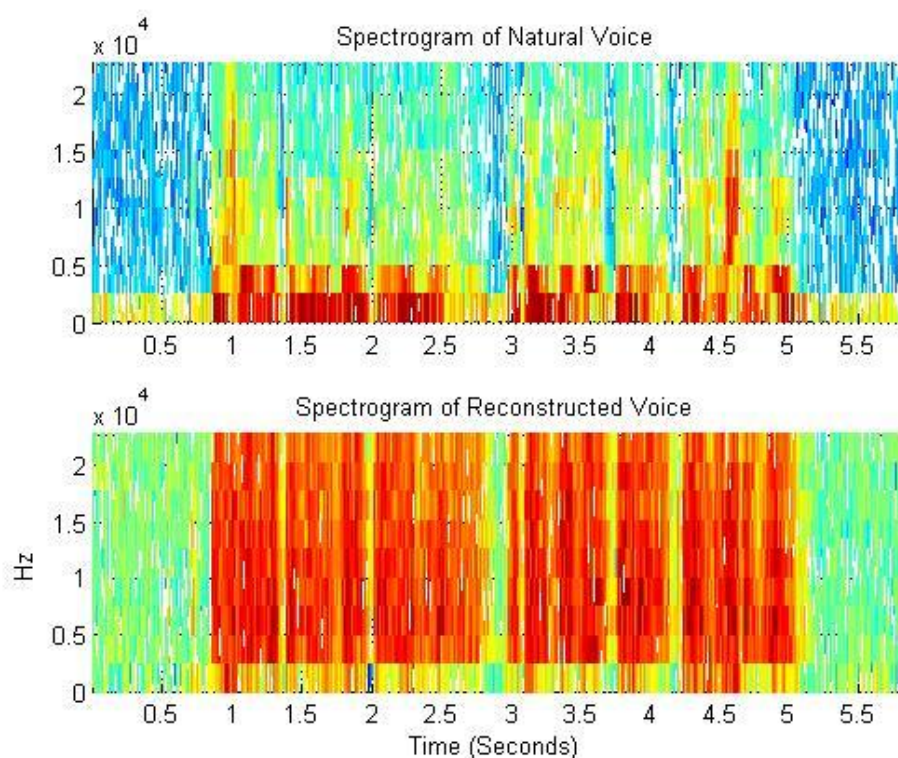


Σχ.9.3 Τιμές των αρχικών συντελεστών γραμ. πρόβλεψης a (μπλε) σε σύγκριση με τις τιμές των κβαντισμένων συντελεστών γραμ.πρόβλεψης (κόκκινο)

Κατά την κβάντιση των παραμέτρων του συνθέτη **CELP** παρατηρήσαμε ότι τα αποτελέσματα είναι φυσιολογικά. Παρακάτω φαίνονται τα διαγράμματα της σύνθεσης φωνής και το φασματογράφημα μετά από την κβάντιση.



Σχ.10.1 Αρχικό σήμα φωνής (μπλε χρώμα) και σήμα μετά από σύνθεση CELP encoded (κόκκινο χρώμα)



Σχ.10.2 Spectrogram of original and reconstructed voice using CELP encoded

Συγκρίνουμε τα αποτελέσματα σε σχέση με το πριν και μετά της κβάντισης και παρατηρούμε ότι υπάρχει μείωση στην ποιότητα σύνθεσης.

Θεωρητικά τα φάσματα και η κυματομορφή θα έπρεπε να βρίσκονται κοντά σε σχέση με αυτά πριν τη συμπίεση, με εξαίρεση κάποια παραμόρφωση σε διαστήματα που δε μας ενδιαφέρουν και τόσο. Με τη συμπίεση γλιτώνουμε αρκετές φορές λιγότερη πληροφορία, οπότε αν και η ποιότητα της σύνθεσης δεν είναι η ίδια, τη χρησιμοποιούμε πάντοτε.

Ο αρχικός ρυθμός πληροφορίας του σήματος φωνής, με την υπόθεση ότι έχουμε **16 μπιτς ανα δείγμα** προκύπτει ως

$$I = B \cdot F_s = 16 \cdot 48k \cdot \text{bits/sample} \cdot \text{samples/sec} = \mathbf{768 \text{ kbits/s}}$$

Ο τελικός ρυθμός προκύπτει ως

$$I' = (5 \cdot M + 6 + 1 + 5) \cdot f = \mathbf{27,2 \text{ kbits/s}}$$

$$\mathbf{\text{Compression Ratio} = 28.24:1}$$

References

- [1] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing*. Prentice-Hall, 2010.
- [2] L.R Rabiner and R.W Schafer, *Digital Processing of speech signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.
- [3] Andreas Spanias Ted Painter. *Perceptual coding of digital audio*. Proceedings of IEEE, 88(4), April 2000.
- [4] R.P. Ramachandran, M.M. Sondhi, N.Seshadri and B.s. Atal, “ *A two codebook format for robust quantization of linear spectral frequencies*, ” IEEE trans. Speech and Audio processing vol.3 pp. 157-167, 1995
- [5] *SPEECH COMPRESSION USING LINEAR PREDICTIVE CODING(LPC)* Nikhil Sharma* International Journal of Advanced Research in Engineering and Applied Sciences ISSN: 2278 – 6252
- [6] V.K. Ajmal, C. Mahesh, j. naveen, R. Ksihna, C. Rahulkrishnan, *Adaptive Noise Cancellation, Compression and encrypted transmission of Speech Signals*
- [7] Homomorphic Processing of Speech, article at: http://isites.harvard.edu/fs/docs/icb.topic541812.files/lec11_spr09.pdf
- [8] Speech Analysis - Synthesis system based on Homomorphic Filtering, ALAN V. Oppenheim
- [9] Speech compression article at: <http://www.data-compression.com/speech.shtml>
- [10] Speech coding article and implementation at: <http://markus-hauenstein.de/sigpro/codec/codec.shtml>
- [11] J.D MARKEL, A.H GRAY, Jr. Pages 10-96, 190-158, *Linear Prediction of Speech*,
- [12] Alan V. Oppenheim/ Ronald W. Schafer, *Digital signal Processing*,
- [13] Vinay K. Ingle, John Proakis, *Digital signal processing using MATLAB*