Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Balso suspaudimo algoritmų realizavimo įterptinėje sistemoje galimybių tyrimas

Baigiamasis magistro projektas

Ričardas Garmus

Projekto autorius

doc. dr. Mindaugas Knyva

Vadovas

Kaunas, 2020

Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Balso suspaudimo algoritmų realizavimo įterptinėje sistemoje galimybių tyrimas

Baigiamasis magistro projektas

Elektronikos inžinerija (6211EX012)

Ričardas Garmus

Projekto autorius

doc. dr. Mindaugas Knyva

Vadovas

prof. Darius Gailius

Recenzentas

Kaunas, 2020

Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Ričardas Garmus

Balso suspaudimo algoritmų realizavimo įterptinėje sistemoje galimybių tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Ričardo Garmaus, baigiamasis projektas tema „Balso suspaudimo algoritmų realizavimo įterptinėje sistemoje galimybių tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

Ričardas Garmus	
(vardą ir pavardę įrašyti ranka)	(parašas)

Garmus Ričardas. Balso suspaudimo algoritmų realizavimo įterptinėje sistemoje galimybių tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Mindaugas Knyva; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Elektronikos inžinerija, inžinerijos mokslai.

Reikšminiai žodžiai: kalbos signalo apdorojimas, kalbos kodavimas, kokybės įvertinimas, įterptinės sistemos

Kaunas, 2020. 45 p.

Santrauka

Baigiamajame magistro projekte tiriamos didelės spūdos balso algoritmų (CELP, MELP, MELPe, Codec2) realizavimo galimybės įterptinėse sistemose, įvertinant galimybes dirbti realiu laiku. Darbe apžvelgiami balso suspaudimo metodų struktūra bei veikimo principai. Panaudojus didelės spūdos koderių bibliotekas, sukurtas modelis *GNURadio* aplinkoje, kurioje algoritmus galima tikrinti realiame laike. Pasirinktas Codec2 1200 bps balso suspaudimo algoritmas pritaikytas lietuvių kalbai, pasitelkiant specialias LIEPA kalbos įrašų bibliotekas. Darbe atliekami subjektyvūs kalbos kokybės ir suprantamumo įvertinimai, siekiant palyginti balso suspaudimo algoritmus panaudojant lietuvių kalba įgarsintus įrašus. Didelės spūdos balso algoritmų veikimo greitis ir signalo vėlinimas įvertinami naudojantis trimis skirtingais įterptinių sistemų procesoriais: vidutinės spartos ARM Cortex – M4 180 MHz ir ARM Cortex – M7 bei didelės spartos ARM Cortex – A8 1 GHz procesoriumi.

Remiantis subjektyviais įvertinimais, geriausia kalbos kokybė gaunama naudojant MELPe 1200 bps balso suspaudimo algoritmą (3,62 ± 0,99 balo), prasčiausia – Codec2 1200 bps (2,98 ± 1,12 balo). Geriausias kalbos suprantamumas yra Codec2 3200 bps ir MELPe 1200 bps balso suspaudimo algoritmų (atitinkamai 3,6 ± 0,93 ir 3,58 ± 0,93 balo). Objektyvūs PESQ įvertinimai yra iki 1 balo žemesni, nei subjektyvaus metodo. Naudojant lietuvių kalba įgarsintus įrašus apmokytas Codec2 1200 bps balso suspaudimo algoritmas, geriau įvertintas tiek subjektyviais, tiek objektyviais bandymais.

Gauti rezultatai rodo, kad realiu laiku kiekvienoje įterptinėje sistemoje gali veikti visi darbe vertinami balso suspaudimo algoritmai išskyrus MELPe, kurio signalo vėlinimas visose sistemose viršija ITU–T G.114 standarto rekomenduojamą 150 ms ribą. Kodavimas visuose procesoriuose greičiausiai atliekamas naudojant Codec2 balso suspaudimo algoritmus, lėčiausiai – MELPe. Dekodavimas sparčiausiai vyksta naudojant CELP metodą

Garmus, Ričardas. The feasibility study of the implementation of a vocoders in an embedded system. Master's Final Degree Project / supervisor doc. dr., Mindaugas Knyva; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): Electronics Engineering, Engineering Sciences

Keywords: speech signal processing, speech coding, quality estimation, embedded systems

Kaunas, 2020. 45 p.

Summary

The final master's project investigates the possibilities of implementing high compression voice coding algorithms (CELP, MELP, MELPe, Codec2) in embedded systems, evaluating the possibilities to work in real time. The work reviews the structure and operating principles of voice compression methods. Using high compression vocoder libraries, a model has been developed in the *GNURadio* environment where algorithms can be tested in real time. The selected Codec2 1200 bps voice compression algorithm is adapted to the lithuanian language, using special *LIEPA* language recording libraries. Subjective evaluations of speech quality and intelligibility are performed to compare voice compression algorithms using lithuanian audio recordings. The speed and signal latency of high compression voice coding algorithms are evaluated using three different processors of embedded systems: medium-speed ARM Cortex-M4 180 MHz and ARM Cortex-M7 and high-speed ARM Cortex-A8 1 GHz processor.

Based on subjective evaluations, the best speech quality is obtained using the MELPe 1200 bps voice compression algorithm (3.62 ± 0.99 points), the worst - Codec2 1200 bps (2.98 ± 1.12 points). The best speech intelligibility is obtained by using Codec2 3200 bps and MELPe 1200 bps voice compression algorithms (3.6 ± 0.93 and 3.58 ± 0.93 points, respectively). Objective PESQ estimates are up to 1 point lower than the subjective method. Codec2 1200 bps voice compression algorithm trained using lithuanian audio recordings is better evaluated by both subjective and objective tests.

The obtained results show that all voice compression algorithms evaluated in the work can work in real time in each embedded system, except for MELPe, whose signal delay in all systems exceeds the recommended limit of 150 ms of the ITU-T G.114 standard. Encoding on all processors is fastest using Codec2 voice compression algorithms, and the slowest using MELPe. Decoding is fastest using the CELP compression method.

Turinys

Santrumpų ir terminų sąrašas

Santrumpos:

AMDF – (angl. Avarage magnitude difference function) vidutinio dydžio skirtumo funkcija;

CELP – (angl. code excited linear prediction) kodo sužadintas tiesinis prognozavimas;

DMOS – (angl. Degraded Mean Opinion Score) degradacijos vidutinės nuomonės balas;

GFT – greitoji Furje transformacija;

LPC – (angl. Linear prediction coding) tiesinio prognozavimo kodavimas;

LSP – (angl. Line spectral pairs) linijinės spektrinės poros;

MELPe – (angl. mixed exitation linear prediction enchanced) patobulintas mišraus sužadinimo tiesinis prognozavimas;

MELP – (angl. mixed excitation linear prediction) mišraus sužadinimo tiesinis prognozavimas;

MOS – (angl. mean opinion score) vidutinės nuomonės balas;

MSE – (angl. Mean squared error) vidutinis kvadratinis nuokrypis;

MSPE – (angl. minimum squared prediction error) vidutinio kvadratinio nuokrypio minimumas;

PESQ – (angl. perceptual evaluation of speech quality) kalbos kokybės suvokimo įvertinimas;

RMSE – (angl. root mean square error) vidutinės kvadratinės šaknies klaida;

SEGSNR – (angl. segmented signal to noise ratio) segmentuotas signalo ir triukšmo santykis;

SNR – (angl. signal to noise ratio) signalo ir triukšmo santykis;

Įvadas

Viena iš pagrindinių žmogaus bendravimo priemonių yra kalba. Modernios komunikacijos sistemos labai priklauso nuo kalbos apdorojimo ir išsiuntimo. Telefonų, internetinės televizijos, vaizdo konferencijos ir balso pranešimai – tai keletas šiuo metu naudojamu pavyzdžių. Naudojant tokį platų balso perdavimo pasirinkimą, visada išliks noras perduoti aukštos kokybės balsą kuo mažesniu juostos pločiu. Daugumos visų šiuolaikinių kalbos suspaudimo metodų funkcija yra diskretizuoti ir užkoduoti analoginį kalbos signalą taip, kad dekoderis galėtų atkurti pirminę kalbą iš koduojamos sekos. Nors dabartiniai duomenų perdavimo būdai teikia didesnį, nei reikalaujama, pralaidumą kalbai, tačiau dėl brangių tarifų išlieka poreikis mažinti siunčiamų duomenų kiekį, kartu mažinant ir užimamos juostos plotį.

Šiuolaikiniuose balso suspaudimo algoritmuose didžiausias kompromisas ieškomas tarp siunčiamų duomenų kiekio ir tarp kalbos kokybės bei suprantamumo. Tačiau reikia paminėti, kad taip pat, priklausomai nuo panaudojimo, reikia apsvarstyti kitus apribojimus, pvz., algoritmo sudėtingumą, kalbos vėlinimą bei našumą. Našumas ypač aktualus elektronikos prietaisams, kurie turi ne tik siųsti suspaustą balsą, bet ir atlikti kitas jiems priklausančias funkcijas.

Didelės spūdos balso suspaudimo algoritmai, kuriems reikalingas mažesnis nei 5000 bps duomenų srautas, naudojami karinėje technikoje, kadangi tokios didelės spūdos metodus sudėtinga dekoduoti, nežinant specifiškai pasirinkto kodavimo būdo. Tokio tipo metodai taip pat plačiai naudojami radijo mėgėjų, kurie palaiko komunikaciją pasitelkdami skaitmeninį balso perdavimo metodą.

Įvairių kalbos suspaudimo metodų įvertinimas yra sudėtinga užduotis, kadangi reikia įvertinti kalbos kokybę, suprantamumą, reverberaciją, triukšmą, klausymo sudėtingumą, garso lygio tinkamumą ir t. t. Dauguma šių parametrų yra subjektyvūs, kadangi kiekvienas žmogus gali suspaustą kalbą vertinti skirtingai. Norint palyginti balso suspaudimo algoritmus, sukurti ir subjektyvaus vertinimo testai, ir objektyvūs algoritmai.

Darbo tikslas – ištirti didelės spūdos balso suspaudimo algoritmų (Codec2, MELP, MELPe, CELP) realizavimo galimybes įterptinėse sistemose, įvertinant galimybes dirbti realiu laiku. Pritaikyti pasirinktą balso suspaudimo algoritmą lietuvių kalbai, apmokymui panaudojant specialias lietuvių kalbos įrašų bibliotekas.

Darbo uždaviniai:

1. atlikti balso suspaudimo algoritmų ir jų įvertinimo būdų apžvalgą;
2. sumodeliuoti balso suspaudimo algoritmus *GNURadio* aplinkoje
3. apmokyti pasirinkto balso suspaudimo algoritmo duomenų bazę naudojant lietuvių kalbos įrašus
4. atlikti balso suspaudimo algoritmų subjektyvius ir objektyvius įvertinimus
5. atlikti eksperimentinius tyrimus naudojant įterptinių sistemų maketus.

1. Apžvalginė dalis

1.1. Kalbos charakteristikos

Balso aparatas susideda iš centrinės ir periferinės dalių. Centrinė dalis yra smegenys ir nervai, jie kontroliuoja periferinę dalį, kuri yra tarsi balso generavimo sistema, susidedanti iš balso klosčių, burnos ertmės, nosiaryklės ir t. t. Iš balso susidaro kalba, kuri skirstoma į skardžią – kai balso klostės vibruoja ir į duslią – kai balso klostės nevibruoja. Šiuo atveju garsas sudaromas oro srautui tekant per rezonatorius. Skardus balsas yra periodinis, o duslūs garsai labai chaotiški, praktiškai atsitiktiniai. Skardūs garsai dažniausiai yra balsės, o duslūs – priebalsės.

Atvaizduoti balso atkūrimą patogu pagal šaltinio–filtro modelį (žr. 1.1 pav.), kuriame garsas sužadina filtrą, suformuotą iš rezonatorių [1]. Filtro išėjimas yra balsas, kurio charakteristikos nestacionarios laike, taigi, ir balso suspaudimo algoritmai nestacionarūs. Dauguma balso kodavimo algoritmų naudoja šį kalbos modelį dėl savo paprastumo, o didžiausias skirtumas tarp suspaudimo algoritmų yra tai, kaip kuriamas balso šaltinis.

1.1 pav. Šaltinio–filtro modelis(adaptuota pagal[1])

Vienas svarbiausių skardaus balso parametrų yra fundamentalus dažnis F0, kuris nusako kokiu dažniu virpa balso klostės. Tiksliai nustatyti besikeičiantį žmogaus balso toną gana sudėtinga, negelbėja ir tai, kad žmogaus ausis labai jautri šiems pokyčiams [1].

1.2. Balso suspaudimo algoritmų apžvalga

1.2.1. Tiesinio prognozavimo kodavimas

Tiesinio prognozavimo kodavimo (angl. *Linear prediction* coding (LPC) sistemoje kalbos signalas modeliuojamas naudojantis filtru H(z) [2]:

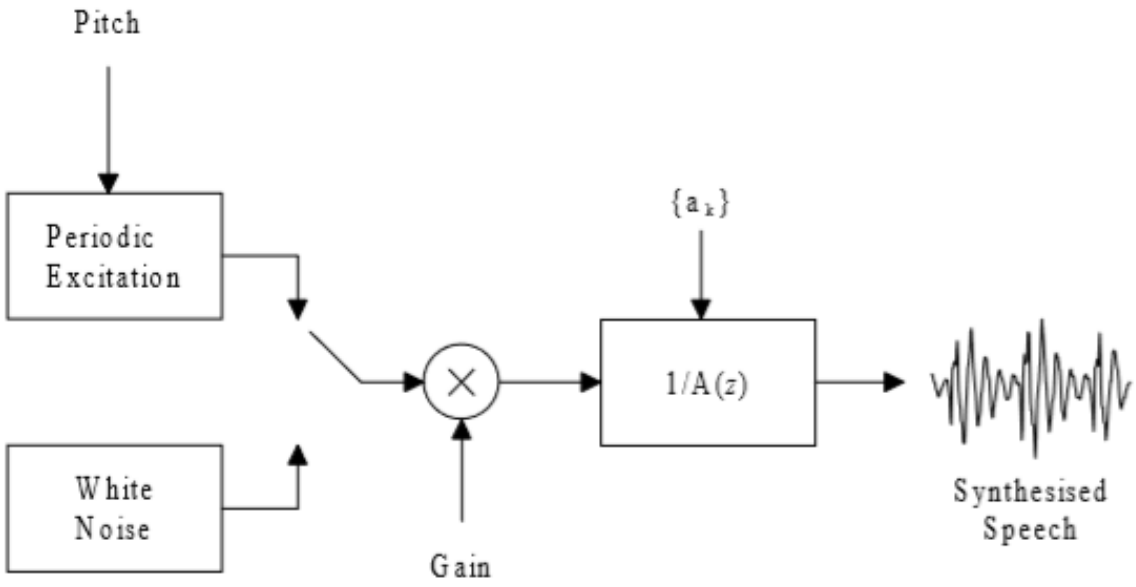
$$H(z) = \frac{U(z)}{A(z)} = \frac{U(z)}{1-\sum_{i=1}^{p} a_k z^{-i}} \quad , \tag{1}$$

čia U(z) – sužadinimo signalas, a_k – tiesinio prognozavimo koeficientai, p – koeficientų skaičius.

Sužadinimo signalai yra triukšmas, diskretus pulsas ar triukšmo ir pulso kombinacija [2]. Beveik visi sistemos nuliai atitinka balso aparato formantinius dažnius. Kuo daugiau naudojama tiesinio prognozavimo koeficientų, tuo geriau filtras modeliuos balso aparatą, tačiau taip didinamas perduodamų duomenų kiekis. Taigi, koeficientų kiekio pasirinkimą lemia norimas tikslumas bei maksimalus signalo juostos plotis. Tipiškai koeficientų kiekis p yra lygus dvigubam signalo juostos pločiui, matuojant kilo hercais [2]. Naudojantis šia sistema galima ir analizuoti ir sintezuoti kalbą [1].

LPC dekoderio blokinė diagrama pavaizduota 1.2 pav. Ji susideda iš dviejų skirtingų garso šaltinių, stiprinimo ir filtro. Priklausomai nuo to ar balsas yra skardus, ar duslus, naudojamas periodinis

sužadinimas arba baltas triukšmas. Periodiniam sužadinimui reikalingas balso tono periodas(angl. Pitch). Sužadinimo signalas stiprinamas ir filtruojamas. Taigi, norint užkoduoti balso signalą LPC metodu, reikia siųsti filtro koeficientus, stiprinimo vertę, balso periodo trukmę (kai balsas skardus) ir sprendimą ar balsas yra skardus, ar duslus.



1.2 pav. LPC dekodavimo metodo blokinė diagrama (paimta iš [1])

PC koderio pasirinkimas ar balsas duslus, ar skardus atliekamas matuojant kalbos signalo segmento energiją ir dažnį. Skardus balsas yra didesnės amplitudės ir turi didesnę energiją, o duslus balsas mažesnes energijos, tačiau didesnio dažnio [3]. Tiek skardaus, tiek duslaus balso vidutinė vertė arti nulio, taigi, dažnis matuojamas skaičiuojant kiek kartų buvo kirsta signalo x ašis. Tuomet sprendimas ar balsas duslus, ar skardus daromas lyginant apskaičiuotas reikšmes su iš anksto nuspręstomis. Taip pat vertinami buvę ir būsimi kalbos segmentai, kadangi neturėtų būti duslaus balso segmentų vidury skardžių. Reikia paminėti, kad naudojantis šiuo modeliu, dekoduojami tik skardūs ar duslūs garsai, o žmogaus kalba gali turėti garsus, kurie nėra tinkamai klasifikuojami kaip duslūs ar skardūs.

Balso tonas matuojamas pritaikant autokoreliacinį algoritmą arba vidutinio dydžio skirtumo funkciją (angl. *average magnitude difference function* (AMDF), kuri naudojama LPC–10 koderyje. Naudojant AMDF funkciją taip pat galima nustatyti ar balsas yra skardus, ar duslus [3].

Kiekvienas balso kalbos segmentas turi savo atskirus tiesinio prognozavimo filtro koeficientus a_k. Priklausomai nuo to ar balsas skardus ar duslus, skiriasi filtro koeficientų skaičius. LPC–10 koderyje skardžiam balsui naudojama dešimt tiesinio prognozavimo koeficientų, o dusliam balsui užtenka tik keturių. Norint surasti tinkamiausius filtro koeficientus, skaičiuojamas vidutinio kvadratinio nuokrypio (angl. *mean squared error* (MSE*)* minimumas. Yra du pagrindiniai MSE minimumo įvertinimo metodai: autokoreliacijos ir autokovariacijos. Autokoreliacijos metodas populiaresnis, kadangi gali būti naudojamas efektyvus *Levinso–Durbino* rekursinis algoritmas [1]. Tačiau kitame straipsnyje rašoma, kad autokoreliacijos skaičiavimo efektyvumas gali būti pagerintas naudojant *Shuro* algoritmą [4].

Tiesinio prognozavimo koeficientai nėra tinkami kvantavimui, kadangi bet kokia klaida dėl kvantavimo gali sukelti filtro nestabilumą, nes filtro koeficientai turi galimybę atsidurti už vienetinio apskritimo ribų. Norint išvengti šios problemos, LPC koeficientai konvertuojami į linijinius spektrinius dažnius (angl. *line spectral frequencies* (LSF). Šis metodas suteikia galimybę kvantuoti LPC koeficientus naudojant mažiau bitų, užtikrina, kad filtras visada būtu stabilus ir leidžia LPC koeficientus sklandžiai interpoliuoti [4].

Stiprinimas gaunamas skaičiuojant kalbos segmento vidutinės kvadratinės šaknies vertę.

1.2.2. CELP kodavimo metodas

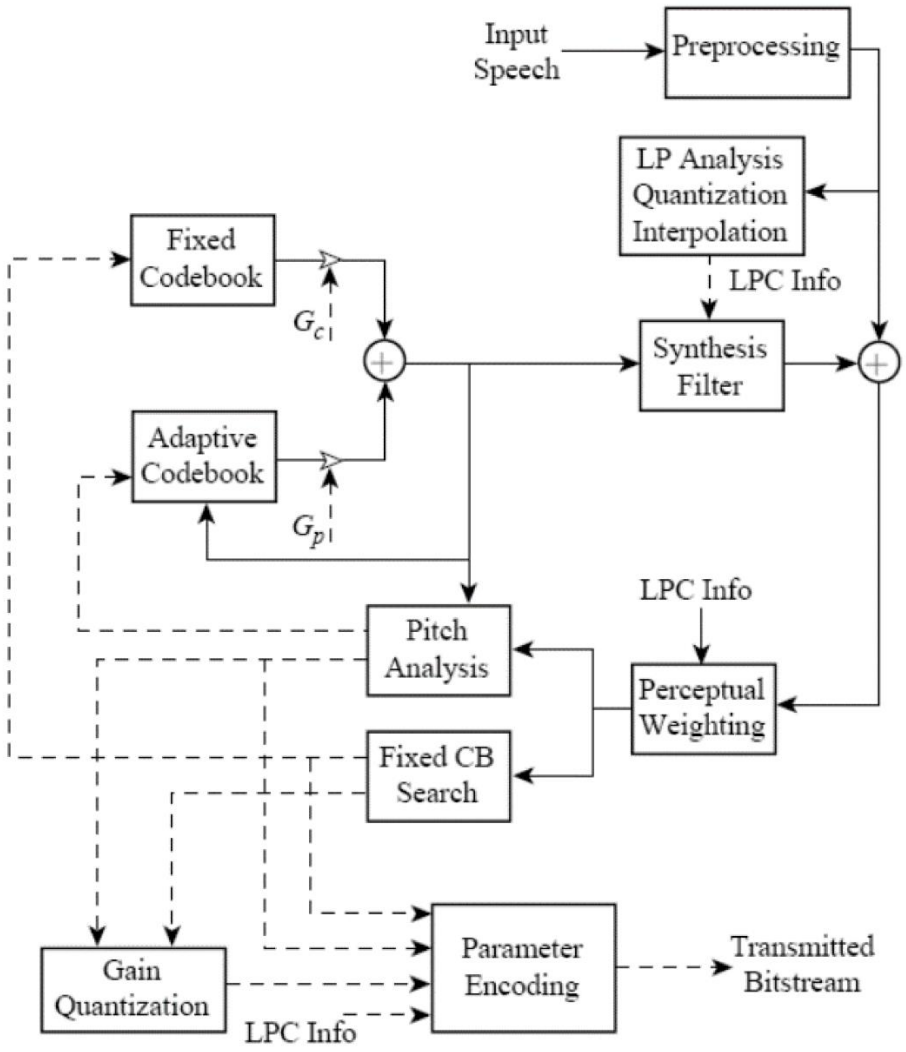
Kodo sužadintas tiesinio prognozavimo (angl. *code excited linear prediction* (CELP) balso suspaudimo metodas naudoja sintezės analizės būdą, tiesinį prognozavimą ir sverto vektoriaus kvantavimą. CELP balso spūdos metodas naudojamas daugelyje kodavimo standartų: (G.729, G.723.1, G.728, IS–54, IS–96, RPE–LTP (GSM), FS–1016 (CELP) [5]. CELP kodavime kalbos signalas pirmiausia dalijamas į imčių blokus, kurių ilgis apie 30 ms [6]. Pagrindiniai CELP kodavimo procesai po signalo padalinimo yra: tiesinis prognozavimas, adaptyvi kodų knygos paieška, stochastinės kodų knygos paieška. CELP koderis siunčia tiesinio prognozavimo koeficientus, adaptyvios ir statinės kodų knygos indeksus ir jų stiprinimą (žr. 1 lentelę).

1 lentelė. CELP kodavimo charakteristikos (adaptuota pagal [7])

	Tiesinis Prognozavimas	**Adaptyvi Kodų knyga**	**Statinė kodų knyga**
Atnaujinimo dažnis	30 ms	30/4 = 7,5 ms	30/4 = 7,5 ms
Parametrai	10 LPS	256 žodžiai	512 žodžių
Bitai per imties bloką	34 (3,4,4,4,4,3,3,3,3,3)	indeksas: 8+6+8+6 stiprinimas: 5x4	indeksas: 9x4 stiprinimas: 5x4
Reikalingas greitis	1133,33 bps	1600 bps	1866,67 bps
Papildoma informacija	Like 200 bps yra naudojami: 1 bitas sinchronizacijai, 4 bitai tiesiniai klaidos korekcijei, 1 bitas toliasniam plėtimuisi		

CELP kodavimo metodo blokinė diagrama pavaizduota 1.3 pav. Ji susideda iš tiesinio prognozavimo koeficientų analizes bloko, statinės kodų knygos, adaptyvios kodų knygos, tiesinio prognozavimo filtro bei svertinio suvokimo filtro.

CELP kodavimo metode sužadinimo signalas modeliuojamas naudojantis statine ir adaptyvia kodų knyga. Statinė kodų knyga turi 512 kodų, o adaptyvi 256 kodus. Adaptyvi kodų knyga modeliuoja ilgalaikį signalo periodiškumą, o statinės kodų knygos paieškos tikslas yra originalaus kalbos vektoriaus ir adaptyvios kodų knygos sužadinimo skirtumas. Statinė kodų knyga susideda iš specialių, retų ir persidengiančių trijų kvantų (1,0,–1) *Gauso* sekos, kurios ilgis yra 60 [6]. Abiejų kodų knygų stiprinimai ir indeksai siunčiami keturis kartus per kalbos segmentą. Adaptyvi kodų knyga sudaroma iš pavėlinto sužadinimo signalo, taigi, jos turinys keičiasi kiekviename kalbos segmente.



1.3. pav. CELP kodavimo metodo blokinė diagrama (paimta iš [8])

CELP balso spūdos metode kodų knygų indeksai ieškomi uždaroje kilpoje taip, kad būtu kuo mažesnis MSPE(angl. *minimum squared prediction error*), lyginant su svertinio suvokimo filtro nufiltruotu įėjimo signalu. Tiek statinės, tiek adaptyvios knygų paieškos procedūra yra praktiškai vienoda [6], skiriasi tik kodų knygos ir tikslas.

1.2.3. MELP kodavimo metodas

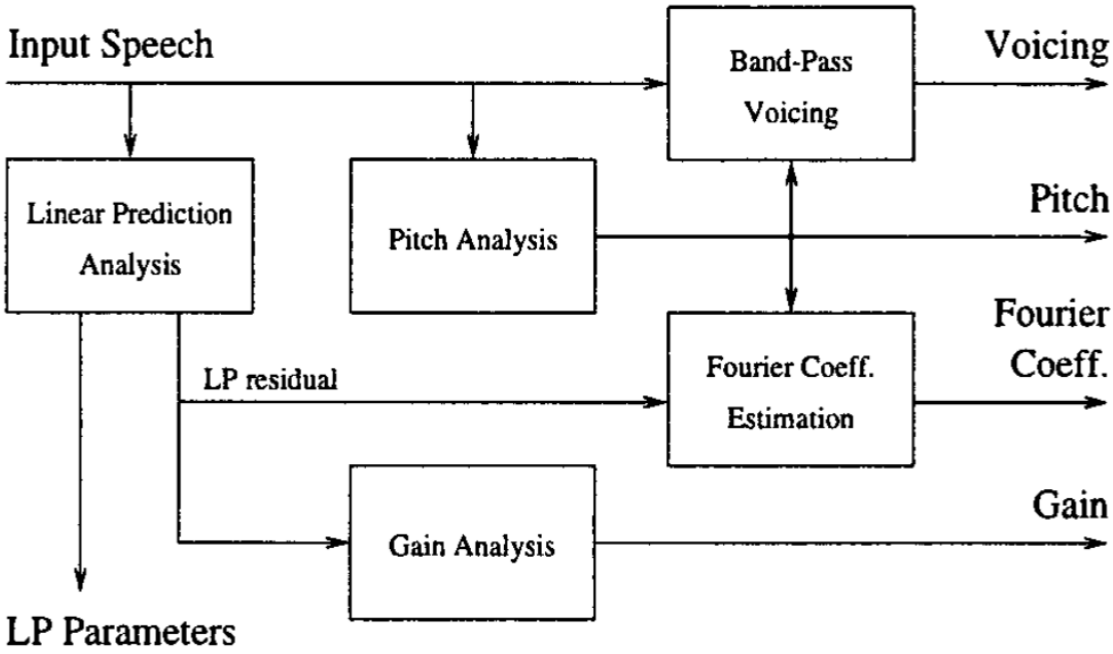
Mišraus sužadinimo tiesinio prognozavimo (angl. *mixed exication linear prediction* (MELP) balso suspaudimo metodas yra parametrinis būdas, kuris buvo naudojamas US federalinio standarto [9]. Jis sukurtas norint pagerinti esantį LPC–10 ir CELP spūdos metodus. MELP standarto užkoduoti duomenys siunčiami 2400 bits/s sparta [9]. Lyginant su LPC, MELP kodavimo metode įvesti drebantys balso segmentai.

MELP balso suspaudimo algoritmui reikalingas bitų kiekis vienam segmentui pateiktas 2 lentelėje. Šiuo metodu siunčiami Furje koeficientai, 10 LSF koeficientų, stiprinimas, balso tono periodas, balso būsena (skardus, duslus) bei aperiodiškumo vėliavėle, nusakanti ar balsas yra drebantis. Vieno segmento ilgis MELP balso suspaudimo algoritme yra 22,5 ms, jam perduoti reikia 53 bitų nepriklausomai nuo esamos balso būsenos. Šis kodavimo metodas išsaugo praeitų segmentų koeficientus tam, kad efektyviau vertintų kalbos charakteristikos pokyčius.

2 lentelė. MELP reikalingas bitų kiekis vienam segmentui (adaptuota pagal [11])

Parametrai	**Reikalingas bitų kiekis**
Balso tonas	7
LSF koeficientai	25
Stiprinimas	8
Balso būsena	4
Aperiodiškumo vėliavėlė	1
Furje koeficientai	8
Viso	53

MELP analizavimo blokinė diagrama pavaizduota 1.4 pav. Ją sudaro tiesinio prognozavimo analizė, balso tono analizė, *Furje* koeficientų įvertinimas bei stiprinimo analizė.



1.4 pav. MELP analizavimo blokinė diagrama (paimta iš [12])

Balso tono nustatymas vykdomas naudojant autokoreliacijos metodą, kai signalas nufiltruotas juostiniu filtru. MELP koderyje naudojami penki juostiniai filtrai, pirmasis skirtas nustatyti balso toną. Sprendimas ar balsas skardus, duslus, ar drebantis priklauso nuo autokoreliacijos stiprumo ir nuo kalbos signalo pikų aukščio [10]. Maža autokoreliacija, bet dideli pikai indukuoja, kad signalas drebantis. Tai pasireiškia dažniausiai tada, kai yra smarkūs balso tono pokyčiai. Tuomet MELP sintezatoriuje arba dekoderyje sužadinimo signalas sudaromas tiek iš triukšmo, tiek iš periodinio signalo generatoriaus.

Furje reikšmės, naudojamos MELP koderyje, sudarytos iš pirmų dešimt balso tono harmonikų, gautų iš LPC analizės likučio (t. y., skirtumas tarp gauto signalo naudojant LPC analizę ir originalaus signalo). Šios harmonikos kvantuojamos ir naudojamos generuojant periodinį sužadinimo signalą.

1.2.4. MELPe suspaudimo metodas

Lyginant su MELP koderiu, patobulintas mišraus sužadinimo tiesinis prognozavimo (angl. mixed exication linear prediction enchanced (MELPe) algoritmas prideda žymių patobulinimų. Turbūt didžiausias iš jų yra tai, kad MELPe gali užkoduoti tą pačią balso informaciją 600 bps ir 1200 bps greičiu. Pagal [11] MELPe iš esmės pagerinta signalo analizė(kodavimas) ir sintezė(dekodavimas). Šiame suspaudimo būde įgyvendintas naujas foninio triukšmo šalinimo algoritmas bei naujas galinis filtras. Šis koderis taip pat veikia su ankstesniu MELP koderiu.

3 lentelė. MELPe reikalingas bitų kiekis vienam segmentui 600 bps režime (adaptuota pagal [14])

Parametrai	**Reikalingas bitų skaičius 90 ms imties blokui**					
	1 režimas	2 režimas	3 režimas	4 režimas	5 režimas	6 režimas
Balso tonas	0	6	8	8	8	8
LSF	36	30	30	30	30	32
Stiprinimas	13	13	11	11	11	9
Balso būsena	5	5	5	5	5	5
Viso	54	54	54	54	54	54

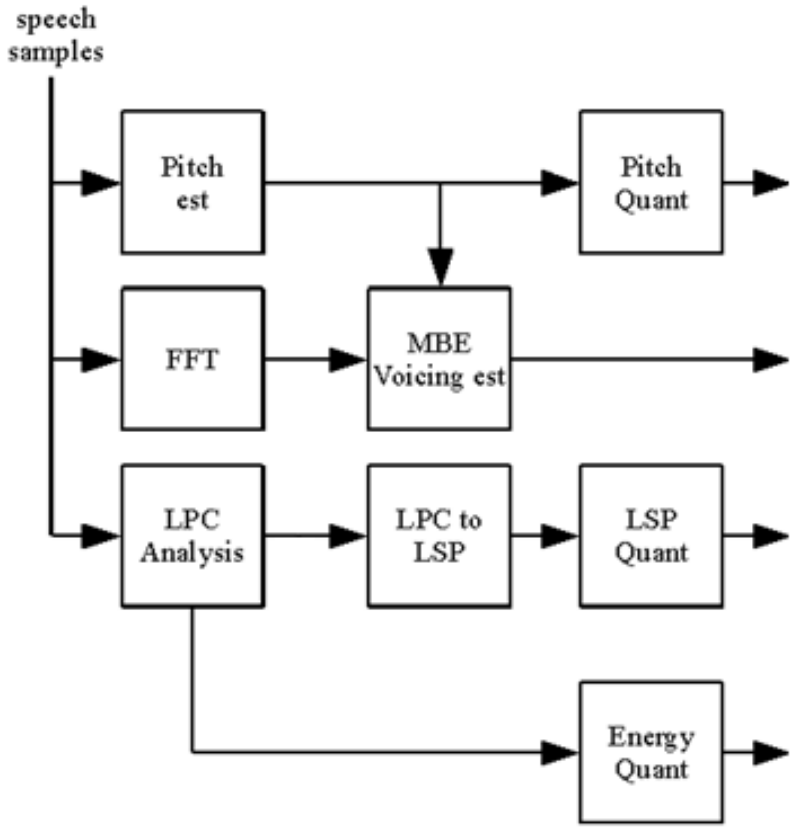
MELPe koderyje, veikiančiame 1200 bps režimu, analizuojamas 67,7 ms imties blokas (540 balso imčių) ir perduodamas 81 bitas duomenų, o 600 bps režimu analizuojamas 90 ms blokas (720 imčių). Iš to galima spręsti, kad tiek 1200 bps režimas, tiek 600 bps turi trigubą signalo vėlinimą, o tai neigiamai veikia komunikacijos sklandumą.

Pagal [12] subjektyvaus tyrimo rezultatus, kalbos suprantamumas geriausiai įvertintas MELPe 1200 bps, prasčiausiai 2400 bps režimuose.

1.2.5. Codec2 balso suspaudimo metodai

Codec2 yra atviro kodo didelės spūdos balso suspaudimo algoritmų paketas. Šis projektas nenaudoja patentuotu algoritmų, yra viešai prieinamas ir, dėl LGPL licenzijos, laisvai modifikuojamas [13]. Codec2 tikslas – suteikti vartotojui nemokamą ir tokios pat kokybės balso spūdos algoritmą kaip MELPe ar MELP. Naudojant Codec2 projektą, galima pasirinkti tarp 3200 bps, 2400 bps, 1600 bps, 1300 bps, 1200 bps ir 700 bps balso suspaudimo rėžimų. Šiuo metu testavimo stadijoje yra 450 bps algoritmas [14, p. 2].

1.5 paveikslėlyje pavaizduota Codec2 kodavimo algoritmo blokinė diagrama. Kaip ir CELP, MELP, MELPe, Codec2 naudoja tiesinio prognozavimo metodą, kurio koeficientai paverčiami į LSF. Pagal LPC analizės rezultatus skaičiuojama energija, kuri naudojama nustatyti signalo garso lygiui. Sužadinimo signalo generavimui Codec2 naudoja triukšmą arba periodinį signalą .



1.5 pav. Codec2 kodavimimo algoritmo blokinė diagrama(adaptuota iš [14])

Codec2 skirtingiems režimams reikalingas bitų kiekis pateiktas 4 lentelėje. Joje matoma, kad priklausomai nuo pasirinktos spūdos, siunčiami skirtingi parametrai ar jų kombinacija.

4 lentelė. Codec2 reikalingas bitų kiekis vienam imties blokui

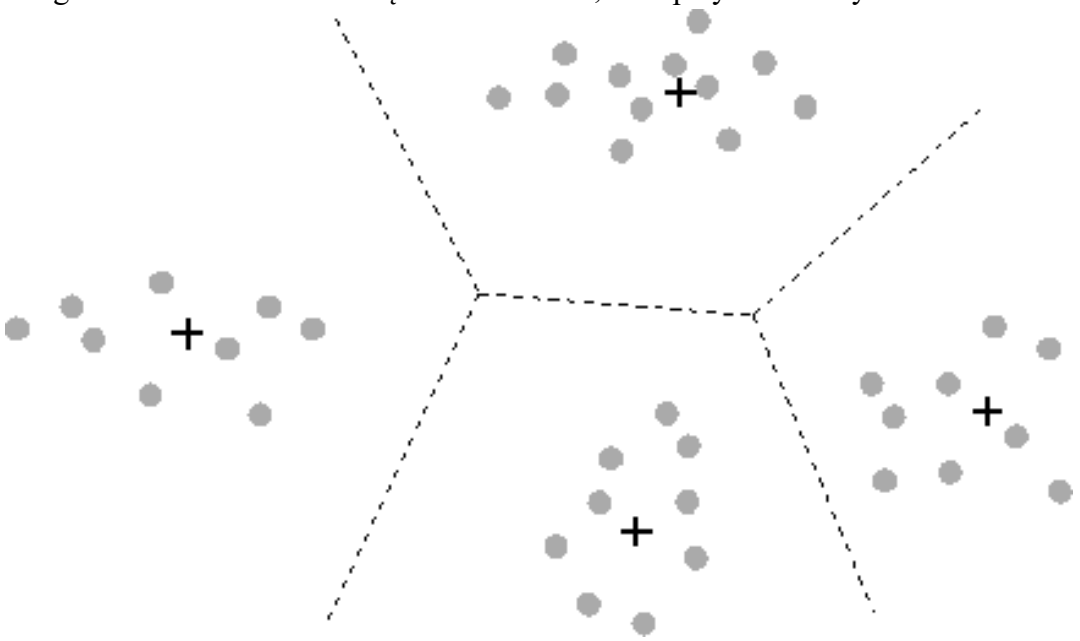
Parametras	**Codec2 rėžimas**						
	3200 bps	2400 bps	1600 bps	1400 bps	1300 bps	1200 bps	700 bps
Imties bloko dydis	160	160	320	320	320	320	320
LSF koeficientai	50	36	36	36	36	27	18
Balso tonas ir energija	0	8	0	36	0	16	0
Balso tonas	7	0	14	0	7	0	0
Energija	5	0	4	0	5	0	4
Duslaus, skardaus balso įvertinimas	2	2	4	4	4	4	6
Laisvi bitai	0	2	0	0	0	1	0
Viso:	64	48	64	56	52	48	28

Codec2 koderyje, veikiant 3200 bps ir 2400 bps, analizuojamas 20 ms imties blokas, kuris atitinkamai suspaudžiamas į 64 ir 48 bitus. Norint pasiekti didesnę duomenų spūdą, Codec2 algoritmų imties blokas išauga iki 40 ms (320 balso imčių). Didesnis imties blokas leidžia išnaudoti koreliacija tarp šalia esančių LPC koeficientų, todėl tokio dydžio imtis naudojama algoritmuose, kurių spūda didesnė nei 2400 bps. Iš 5 lentelėje pateiktų duomenų galima pastebėti, kad didžiausią įtaką balso spūdai turi LSF koeficientai.

1.3. Vektorių klasterizacija

Iš 1, 2, 3, 5 lentelių galima pastebėti, kad didžiausią duomenų dalį užima LPC koeficientai, todėl buvo ieškoma būdų kaip sumažinti reikalingą perduoti bitų kiekį. Plačiausiai naudojamas metodas – vektoriaus kvantavimas, naudojamas CELP,MELP, MELPe, Codec2 koderiuose. Vienas žinomiausių vektoriaus klasterizacijos algoritmų yra k–vidurkio algoritmas.

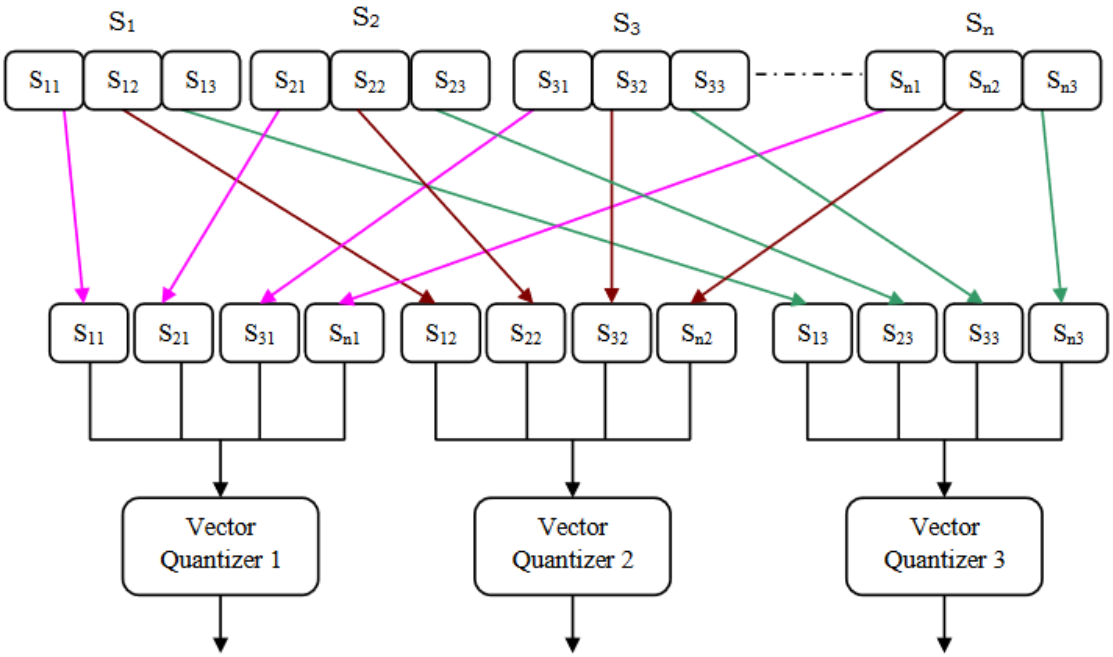
Turint n duomenų taškų realioje d–matmenų erdvėje R^d ir sveiką skaičių, algoritmo tikslas surasti k taškų rinkinį (centrus) R^d erdvėje taip, kad būtu minimalus vidutinis kvadratinis atstumas nuo kiekvieno duomenų taško iki artimiausio centro [15]. Šis atstumas vadinamas kvadratinės klaidos iškraipymu. Deja, bet šiai problemai išspręsti nėra polinominio algoritmo ir sprendimo radimo laikas eksponentiškai kyla nuo taškų skaičiaus [16]. Vektoriaus klasterizavimas vaizduojamas *Voronoi* diagramomis (žr. 1.6 pav.). Šitoks klasterizavimas leidžia smarkiai sumažinti siunčiamų duomenų kiekį, kadangi siunčiamas tik duomenų bazės indeksas, o ne patys duomenys.



1.6 pav. Duomenys ir jų surasti centrai, kai k=4 [17]

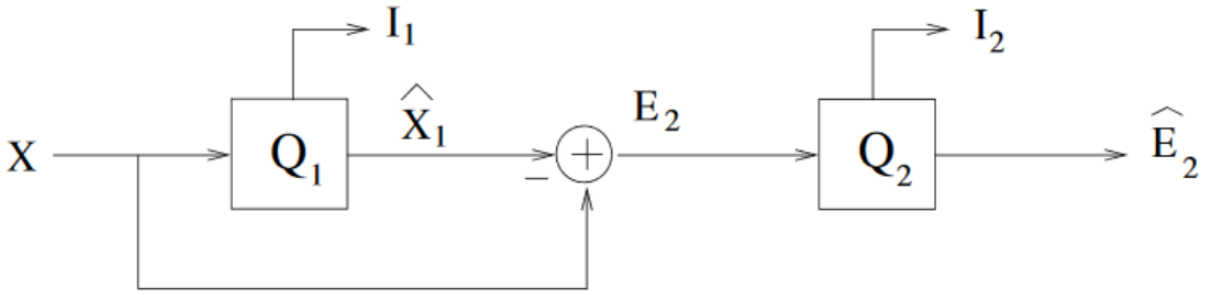
Didžioji dalis k–vidurkio mokymo algoritmų, naudojamų garso apdorojime, skiriasi tik tuo kaip skaičiuojamas kvadratinės klaidos iškraipymas. Pagrindiniai jų: *Euklido* atstumas, pasvertas *Euklido* atstumas, spektrinis atstumas, *Ikura–Saito* atstumas ir t.t. [18]. Kiekvienas iš jų turi savo skirtingas savybes ir reikalingi skirtingiems duomenims klasterizuoti. Vien MELPe balso suspaudimo algoritme pasitelkiami trys skirtingi iškraipymo skaičiavimo būdai [12].

Plačiausiai balso suspaudimo algoritmuose naudojami du klasterizacijos metodai: suskaidytas arba daugiapakopis klasterizavimas. Esminė suskaidytos klasterizacijos idėja – padalinti didelės dimensijos vektorių į mažesnius, kurie tuomet kvantuojami atskirai (žr. 1.7 pav.). Pagrindinis šio metodo klausimas – į kiek dalių skaidyti vektorių ir kokio dydžio turi būti kiekviena dalis.



1.7 pav. Suskaidytos klasterizacijos blokinė diagrama [19]

Daugiapakopis klasterizavimas vektorius kvantuoja keliais etapais naudojant nuoseklaus tobulinimo metodą. Dviejų pakopų klasterizavimo metodo blokinė diagrama pateikta 1.8 pav.



1.8 pav. Dviejų pakopų vektoriaus klasterizavimo algoritmo blokinė diagrama [20]

Šio metodo tikslas – rasti skirtingų etapų vektoriaus derinį, siekiant sumažinti iškraipymą. Daugiapakopio klasterizavimo metodas sudėtingesnis nei suskaidyto, kadangi negalima atskirai mokyti kiekvienos pakopos. Metodo trūkumas yra tai, kad jo vektoriaus paieška trunka ilgiau nei suskaidyto vektoriaus. Remiantis [21] straipsnu, daugiapakopis metodas geresnis ir naudojamas MELP, MELPe, Codec2 algoritmuose.

1.4. Kalbos įtaka balso suspaudimo algoritmams

Kadangi CELP, MELP, MELPe ir Codec2 koderiai naudoja vektorių klasterizaciją išgauti didesnei spūdai, tai galima manyti, kad balso suspaudimo algoritmų kalbos kokybė ir suprantamumas priklausys nuo kalbos, kuri buvo panaudota sudaryti duomenų bazę. Kalbos įtaka CELP tipo koderiams vertinama [22] straipsnyje, kurio eksperimentiniai rezultatai nurodo, kad skirtingos kalbos neturėjo didelės įtakos kalbos perteikimo kokybei, tačiau rezultatai stabilesni naudojant anglų kalbą. [23] Publikacijoje buvo tiriama kalbos įtaka LPC tipo koderiams. Joje rašoma, kad geresnė kalbos perteikimo kokybė gali būti pasiekiama kodų knygą sugeneravus kalba, kuria bus komunikuojama.

Tai patvirtinama [24] straipsnyje, kuriame skelbiama, kad naudojantis anglų kalbos įrašais, pastebima aukštesnė kokybė ir stabilumas lyginant su arabų ir lietuvių kalbomis.

1.5. Skyriaus apibendrinimas

Šiame skyriuje apžvelgtos pagrindinės balso charakteristikos, CELP, MELP, MELPe ir Codec2 balso suspaudimo algoritmai. Nustatyta, kad tiesinio prognozavimo koeficientai naudojami kiekviename balso suspaudimo algoritme – tai yra tarsi kiekvieno metodo bazė. Norint, kad filtro koeficientai būtu stabilūs, jie yra konvertuojami i linijines spektrines poras, kurios užtikriną, kad nuliai ir poliai patektų į vienetinio apskritimo ribas.

Didžiausias skirtumas tarp balso suspaudimo algoritmų – sužadinimo signalo ieškojimo būdai. CELP koderyje naudojamos dvi skirtingos kodų knygos, codec2 ieško ar balsas skardus ar duslus, MELP algoritmas papildomai įveda trečią drebančia balso būseną.

Iš apžvelgtų balso suspaudimo metodų, MELPe gali pasiekti didžiausią spūda – jam reikalingas 600 bps duomenų srautas. Nedaug atsilieka vienas iš Codec2 algoritmas, kuriam reikia 700 bps.

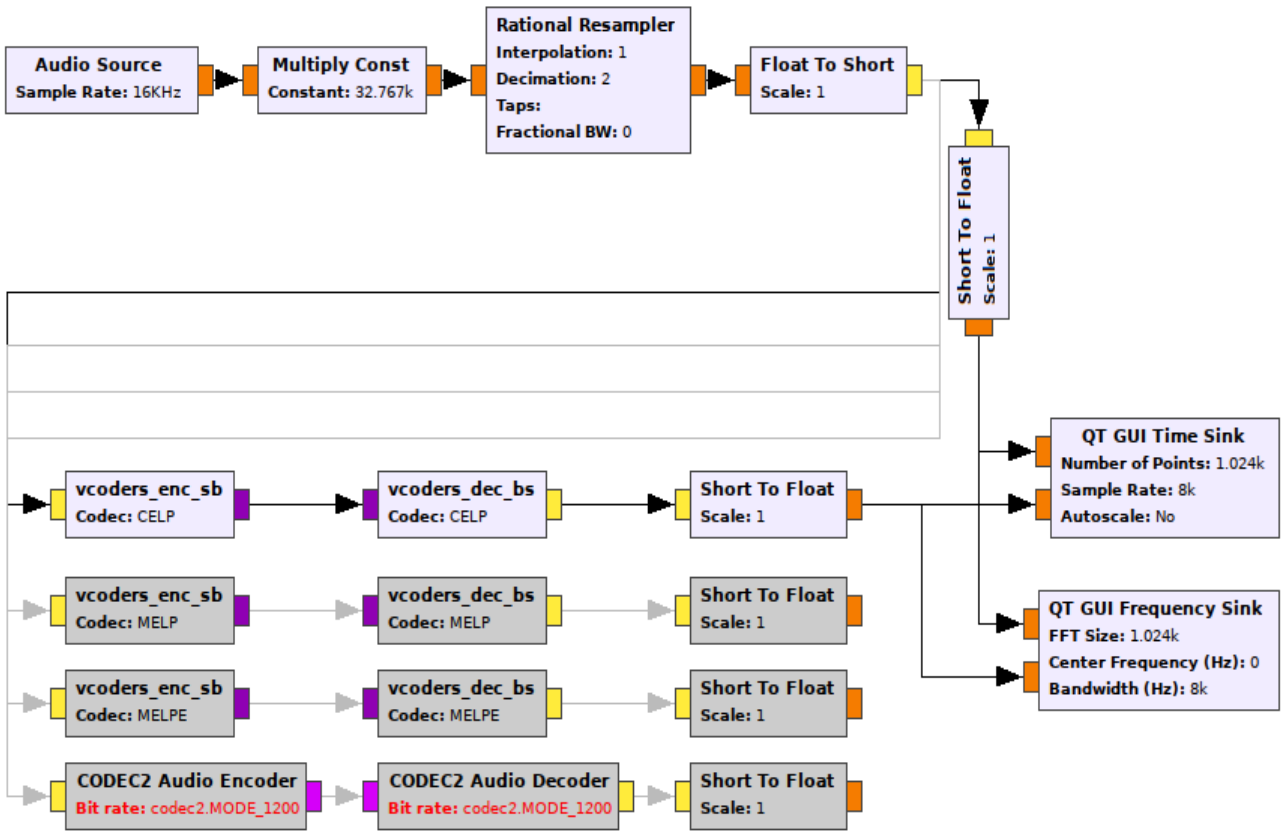
Didelis balso suspaudimas pasiekiamas klasterizuojant duomenų vektorius. Koderiuose naudojami du pagrindiniai klasterizavimo metodai: daugiapakopis ir suskaidytas. Nuo sudarytu kodų knygų priklauso balso suspaudimo algoritmo kokybė ir suprantamumas.

2. Metodinė dalis

2.1. Balso suspaudimo algoritmų modeliavimas GNURadio aplinkoje

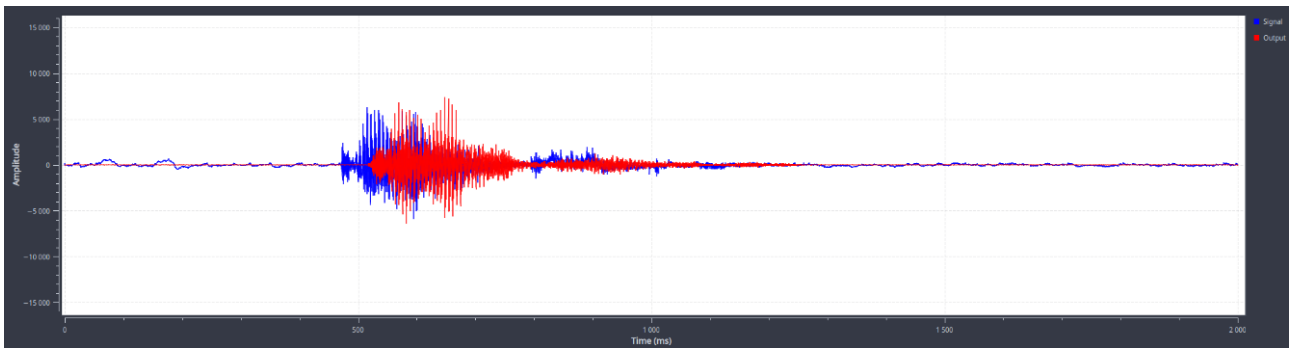
Balso suspaudimo algoritmai įgyvendinti *GNURadio* programinėje aplinkoje [25]. *GNURadio* leidžia lengvai tikrinti ir modifikuoti sukurtą modelį realiu laiku. Šių algoritmų tikrinimui buvo sukurti atskiri blokai tiek kodavimui, tiek dekodavimui. Kadangi Codec2 balso suspaudimo algoritmai *GNURadio* modeliavimo aplinkoje jau įgyvendinti, tai sukurtuose blokuose galimas pasirinkimas tarp CELP, MELP ir MELPe koderių.

Sumodeliuotų koderių veikimas patikrintas į įėjimą duodant realiu laiku įrašomus mikrofono duomenis. Modelio diagrama pateikta 2.1 pav. Čia „Audio Source“ – mikrofono įėjimas, kuris diskretizuojamas 16 kHz dažniu. Tuomet signalas padauginamas iš 32767 konstantos ir taip konvertuojamas iš [-1; 1] rėžiuose esančio slenkančio kablelio į sveiko skaičiaus duomenis. Vėliau signalo diskretizavimo dažnis sumažinamas iki 8 kHz ir modeliavimo aplinkoje pakeičiamas duomenų tipas. Kiekvieno koderio veikimas tikrinamas į „QT GUI Time Sink“ ir „QT GUI Frequency Sink“ blokus prijungiant pasirinktą balso suspaudimo metodą. Pateiktoje diagramoje veikimas tikrinamas realiu laiku stebint laikinį ir greitosios *Furje* transformacijos (GFT) grafiką, tačiau pridėjus „Audio Sink“ arba „File Sink“ blokus, galima įrašus klausytis arba išsaugoti faile.



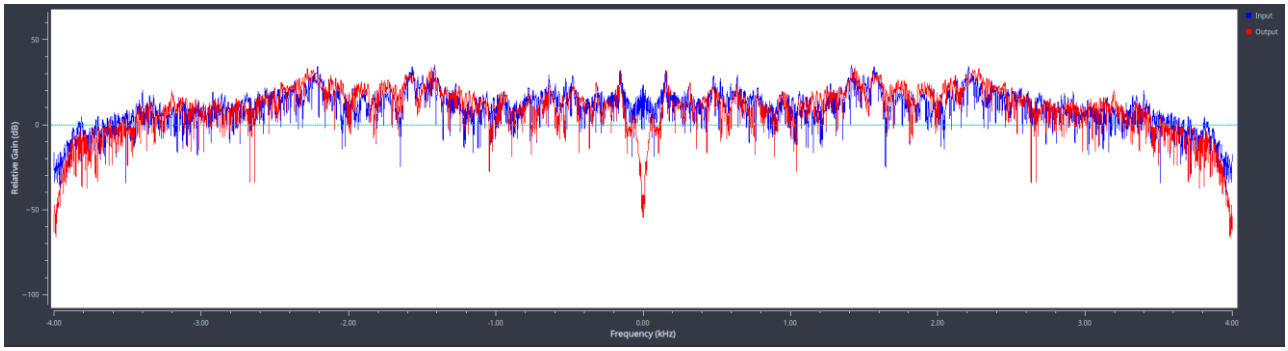
2.1 pav. GNURadio koderių tikrinimo diagrama

Originalaus ir CELP balso suspaudimo algoritmu apdoroto signalų oscilogramos yra pateikiamos 2.2 pav.



2.2 pav. CELP koderio laikinis ir originalus signalai, kai sakomas žodis „testas“. Originalus signalas – mėliynos spalvos, apdorotas – raudonos.

Analizuojant laikinio signalo grafiką, pastebimas signalo vėlinimas, kuris šiuo atveju yra 38 ms. Originalaus ir CELP koderio apdoroto GFT grafikai pavaizduoti 2.3 paveikslėlyje.



2.3 pav. CELP koderio apdoroto ir originalaus signalo GFT, kai sakomas žodis „testas“. Originalus signalas – mėlynos spalvos, apdorotas – raudonos.

GFT diagramoje matoma, kad signalas nufiltruotas iki 80 Hz. Taip pat galima pastebėti, jog atitinka balso fundamentinis dažnis ir jo harmonikos.

2.2. Balso suspaudimo algoritmų subjektyvaus ir objektyvaus vertinimo metodai

2.2.1. Balso suspaudimo algoritmų subjektyvaus vertinimo metodai

Vidutinės nuomonės balas (angl. *Mean Opinion Score* (MOS) – labai paplitęs garso ir vaizdo įvertinimo metodas [26]. Šiam įvertinimui yra parašytas ITU – T P.800 [27] standartas, kuriuo naudojantis galima vertinti kalbos kokybę, klausimo sudėtingumą bei garsumo preferenciją. Kiekvienas iš jų turi aprašytą penkių balų skalę, pagal kurią subjektas vertina vieną iš parametrų. Kalbos kokybės vertinimo skalė pavaizduota 5 lentelėje. Galutinis parametro vertinimo rezultatas gaunamas padarius visų vertinimų vidurkį.

5 lentelė. Kalbos kokybės vertinimo skalė

Kalbos kokybė	**Balas**
Puiki	5
Gera	4
Patenkinama	3
Prasta	2
Bloga	1

Degradacijos vidutinės nuomonės balas DMOS(angl. Degraded Mean Opinion Score) yra subjektyvus garso vertinimo metodas, labai panašus į vidutinės nuomonės balą. Šis metodas taip pat aprašytas ITU–T P.800 standarte. DMOS metodu vertinama ar imtis pablogėjo nuo prieš tai buvusios. Imties pablogėjimas vertinamas penkių balų skalėje(žr. 6 lentelė). Galutinis įvertinimas gaunamas padarius visų vertinimų vidurkį.

6 lentelė. Degradacijos vidutinės nuomonės balo skalė

Pablogėjimas	**Balas**
Pablogėjimas nejaučiamas	5
Pablogėjimas jaučiamas, tačiau neerzinantis	4
Pablogėjimas šiek tiek erzinantis	3
Pablogėjimas erzinantis	2
Pablogėjimas labai erzinantis	1

Abu šie metodai gali parodyti naudingos informacijos. Jeigu signale yra triukšmo, tai MOS metodo skalė gali turėti suspaustą dinaminį diapazoną lyginant su triukšmu neužterštu signalu, o DMOS metodas turi pastovų dinaminį diapazoną tiek triukšmu užterštu, tiek švariu signalu. Tačiau MOS metodas geriau parodo kalbėjimo kokybės priimtinumą, nes yra didesnis skirtumas tarp triukšmu užteršto ir švaraus signalo.

Ganėtinai svarbus balso algoritmų įvertinimas yra kalbos suprantamumas. Kaip ir paminėtos aukščiau, ši įvertinimo skalė aprašyta ITU – T P.800, o metodologija ITU – T P.807 [28] standartuose. Naudojantis šiuo įvertinimu, galima spręsti sakinių supratimo sudėtingumą. Išversta vertinimo skalė pateikta 7 lentelėje.

7 lentelė. Kalbos suprantamumo vertinimo skalė

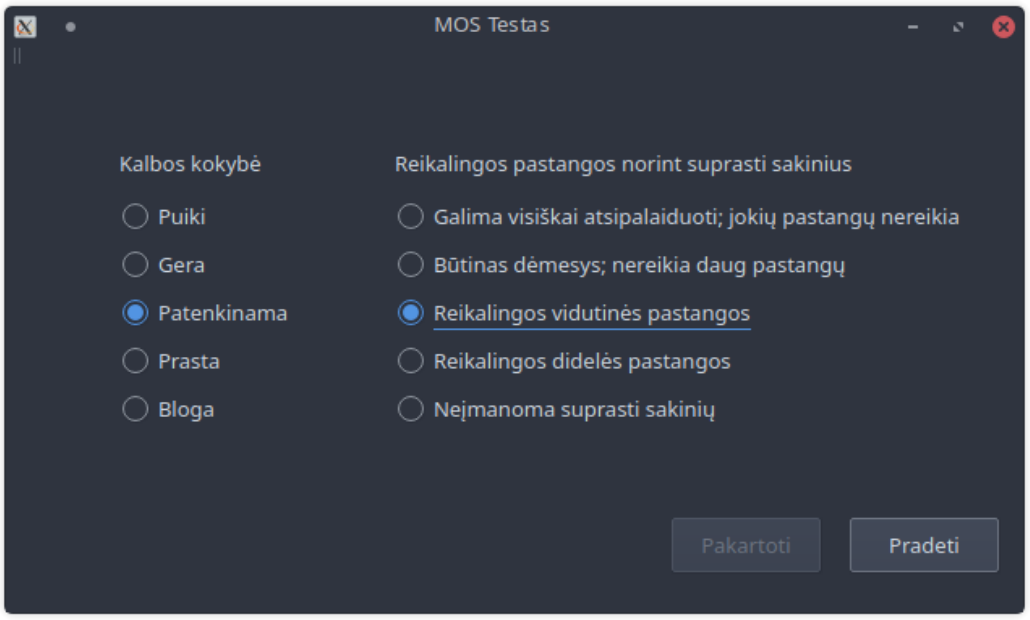
Reikalingos pastangos, norint suprasti sakinius	**Balas**
Galima visiškai atsipalaiduoti; jokių pastangų nereikia	5
Būtinas dėmesys; nereikia daug pastangų	4
Reikalingos vidutinės pastangos	3
Reikalingos didelės pastangos	2
Neįmanoma suprasti sakinių	1

Pagal ITU – T P.800 standartą reikalingi bent keturi subjektai. Šių testų rezultatai priklauso nuo aplinkos bei įrangos, kuri naudojama pateikiant klausimą.

Norint atlikti subjektyvius testus, buvo sukurta programa(žr. 2.4 pav.), kurioje kiekvienas pateiktas balso įrašo failas įvertinamas dviem MOS skalėm: kalbos kokybe ir suprantamumu. Kalbos kokybė ir suprantamumas pasirinkti todėl, kad bandyme naudojami įrašai nėra užteršti triukšmo. Kiekvienam "wav" formato garso failui automatiškai sukuriamas naujas įvertinimo langas. Visi vertinimai patogiai saugomi tekstiniame faile lengvam apdorojimui. Grafinė aplinka sudaryta naudojantis *Qt Creator* [29] programine įranga. Kiekvieną balso įrašą galima klausyti kelis kartus paspaudus mygtuką „pakartoti".

Kievienam subjektyvaus vertinimo nariui buvo išsiųsta ši programa. Visi subjektai patvirtino, kad nėra dalyvavę panašaus tipo apklausose bent vienerius metus. Prieš atliekant testą buvo atsakoma į

visų dalyvių pateiktus klausimus, taip pat paaiškinta, kad kalbos kokybė yra įrašo naturalaus skambėsio vertinimas t.y. ar balsas yra „robotiškas“, ar girdima reverbaraciją ir t.t. Deja, nebuvo galimybės užtikrinti vienodos klausymo aplinkos ir įrangos, subjektai galėjo naudotis tiek ausinėmis, tiek garso kolonėlemis.

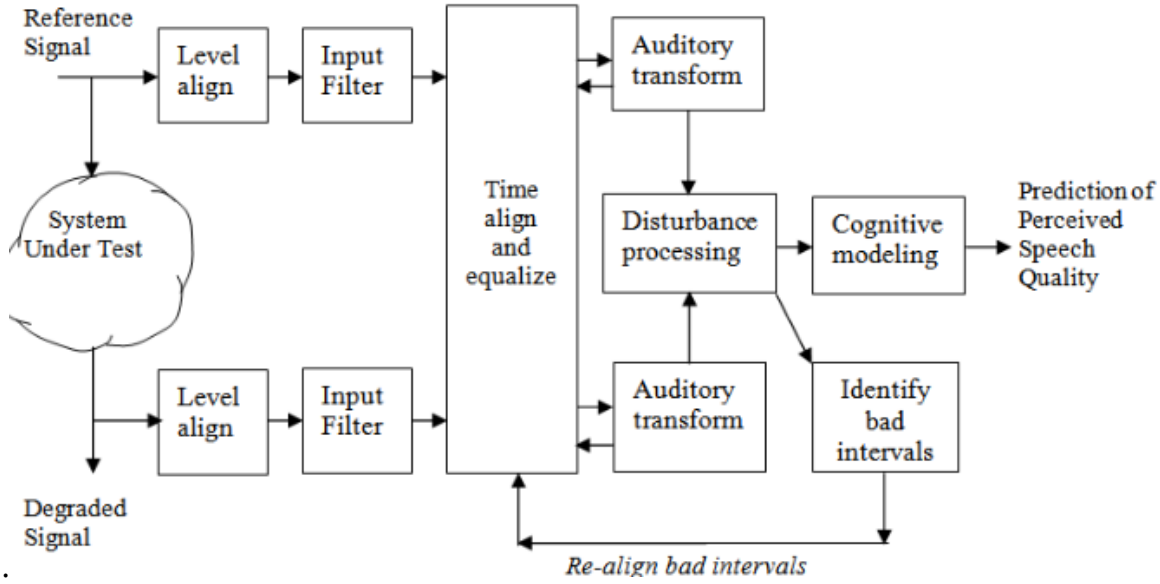


2.4 pav. Subjektyviam vertinimui naudojamos programos langas

2.2.2. Balso suspaudimo algoritmų objektyvaus vertinimo metodai

Signalo ir triukšmo santykis (angl. s*ignal to noise ratio* (SNR) – populiarus objektyvaus vertinimo metodas. Jis plačiai naudojamas dėl savo paprastumo, tačiau turi itin mažą koreliaciją su subjektyviu MOS metodu [30]. Žymiai geresnis rezultatas gaunamas, jeigu naudojamas segmentuotas signalo ir triukšmo santykis (angl. *segmented signal to noise ratio* (SEGSNR) [30]. Signalo segmentai tipiškai suskirstomi į 15–20 ms rėmus. Šio metodo trūkumas yra tai, kad jeigu bus sukaičiojamas intervalas, kuriame žmogus nešneka, tai paprastas triukšmas gali smarkiai iškreipti SEGSNR rezultatą. To galima išvengti naudojant SEGNSR, kai signalo energija yra tam tikro lygio. Kadangi abiem metodams reikalingas originalus x(n) ir apdorotas y(n) signalas, neįmanomą gauti objektyvaus vertinimo realiu laiku. Taip pat SNR ir SEGSNR metodai signalus lygina laike, taigi šie metodai labai jautrūs sinchronizacijai. Jeigu signalai nesinchronizuoti, tuomet SNR ir SEGSNR rezultatai bus labai netikslūs [31].

Vienas plačiausiai naudojamų objektyvaus testavimo metodų yra kalbėjimo kokybės suvokimo įvertinimas (angl. *perceptual evaluation of speech quality* (PESQ), kuris aprašytas ITU – T P.862 standarte [32]. PESQ metodas lygina originalų ir degraduotą įrašus ir kalbos kokybės įvertinimo balą [33], kurio reikšmės yra [-0,5; 4,5] rėžiuose. PESQ metodo veikimo blokinė diagrama pateikta 2.5 pav. Kaip matoma, naudojant šį objektyvaus įvertinimo priemonę nereikia keisti signalo lygio, jį sinchronizuoti ar filtruoti, visa tai jau atliekama pačiame metode. Gauti geresnei koreliacijai su MOS balu, yra sudarytos PESQ standarto rekomendacijos [34], kuriose PESQ balui pritaikoma specifinė transformacijos funkcija.



2.5 pav. PESQ metodo blokinė diagrama [35]

Objektyviam vertinimui pasirinktas PESQ metodas, kuriam atlikti iš oficialaus ITU – T puslapio parsisiunčiamas programos kodas [32]. PESQ įvertinimo programai reikalingas originalus ir degraduotas įrašas. Šiuo atveju originalus failas – sumažinto diskretizavimo dažnio bei pakeisto duomenų kvantavimo įrašas, o degraduotas – balso suspaudimo algoritmu užkoduotas ir dekoduotas „wav“ formato įrašas. Programos rezultatas – du MOS balai: „RAW MOS“ tai originalus PESQ balas; „MOS–LQO“ – „RAW MOS“ balas, kuriam pritaikyta transformavimo funkcija [34], leidžianti PESQ objektyvaus vertinimo metodui turėti didesnę koreliaciją su subjektyviu MOS balu.

2.3. Balso suspaudimo algoritmo duomenų bazės kūrimas

Siekiant pagerinti balso suspaudimo algoritmo kalbos kokybę ir suprantamumą, kodų knygos sudaromos iš specialių lietuvių kalbos įrašų bibliotekų. Šiame darbe apmokymui pasirinktas Codec2 1200 bps balso suspaudimo algoritmas, kadangi jis yra laisvai modifikuojamas ir dėl vienodos spūdos nesudėtingai lyginamas su MELPe 1200 bps metodu.

2.3.1. Duomenų paruošimas

Norint gauti LSF koeficientų duomenų bazę, skirtą Codec2 1200 bps algoritmui, reikalingas platus balso įrašų garsynas. Buvo pasirinktas *LIEPA* garsynas, kurį iš viso sudaro 100 val. trukmės garso įrašai. *LIEPA* garsynas sukurtas dalyvaujant 376 diktoriams, iš kurių 248 moterys ir 128 vyrai [36]. Garsyno įrašų informacija:

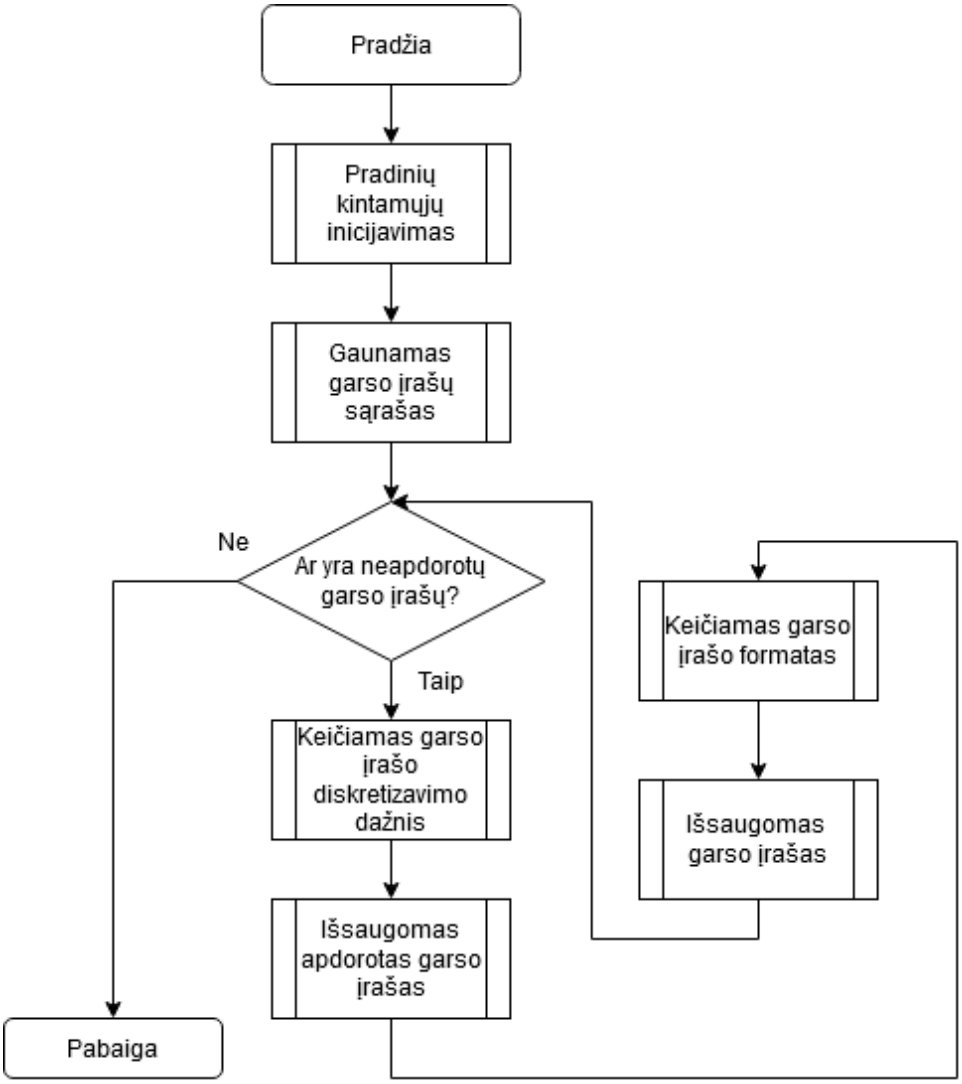
- Diskretizavimo dažnis 22 kHz
- Duomenų formatas „wav“
- 16 bitų sveikojo skaičiaus kvantavimas
- 1 garso kanalas

Kiekvieno įrašo pavadinimas prasideda „S“ arba „Z“ raidėmis, reiškiančiomis, kad tai yra arba sakinys, arba žodis. Toliau seka diktoriaus vardo ir pavardės sutrumpinimas bei to diktoriaus įrašo numeris

Norint apmokyti Codec2 1200 bps balso suspaudimo algoritmo duomenų bazę, pirmiausia reikia tinkamai paruošti balso įrašus – turi būti vienas garso įrašas, kurio parametrai:

- Diskretizavimo dažnis 8 kHz
- Duomenų formatas „raw“
- 16 bitų sveikojo skaičiaus kvantavimas
- 1 garso kanalas

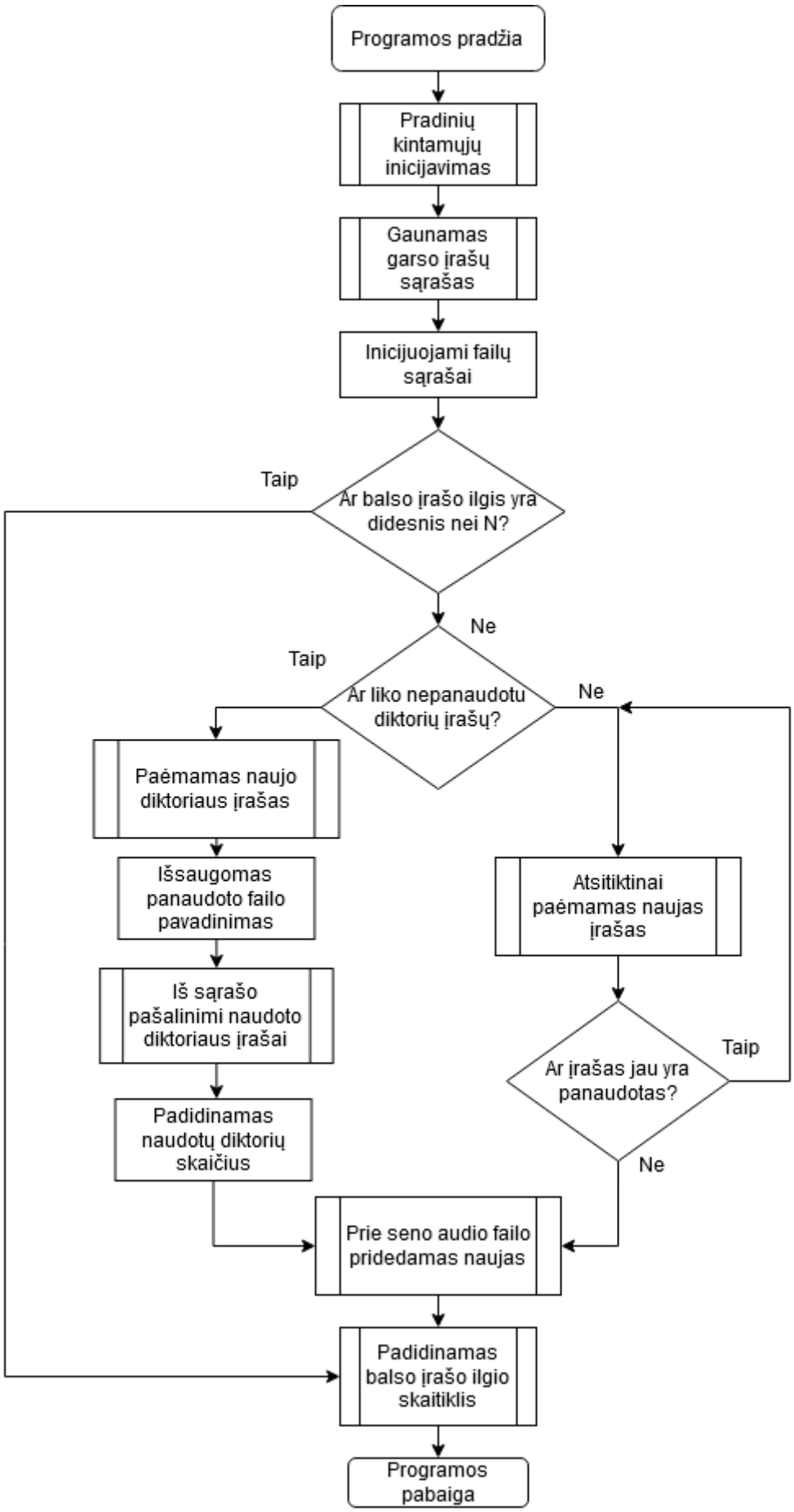
LIEPA balso įrašų paruošimo programos algoritmas pateiktas 2.6 paveikslėlyje.



2.6 pav. *LIEPA* balso įrašų paruošimo programos algoritmas

Programa sudaryta naudojantis *Python3.6* programavimo kalba. Programos pradžioje inicializuojami pradiniai kintamieji – esamų ir apdorotų garso įrašų vieta kietajame diske. Tuomet nuskaitomas kiekvieno failo pavadinimas, kuris išsaugomas į sąrašą. Kiekvieno įrašo diskretizavimo dažnis keičiamas pasitelkiant *librosa Python* biblioteką [37] atliekant signalo decimaciją. Vis dar „wav“ formato įrašai išsaugomi, kadangi bus reikalingi pasirinkto objektyvaus vertinimo metodo panaudojime. Duomenų formatas iš „wav“ į „raw“ keičiamas naudojantis „SoX“ [38] programų paketu. „Raw“ formatas yra toks pat įrašas kaip ir „wav“, tačiau neturi antraštės. Kiekvienas „raw“ failas įrašomas į kietąjį diską tolimesniam apdorojimui.

Vieno garso įrašo sudarymo algoritmas pateiktas 2.7 pav.

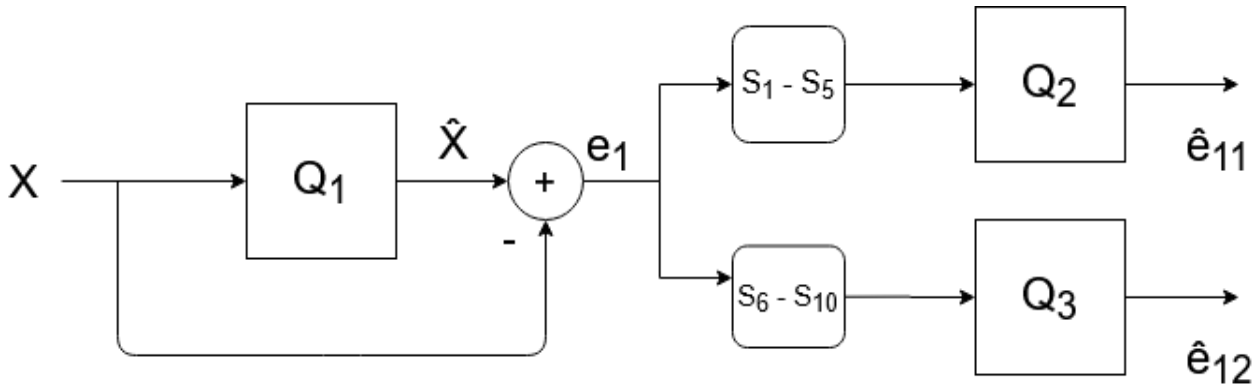


2.7 pav. Vieno garso įrašo sudarymo algoritmas

Python3.6 programos tikslas – sudaryti reikiamo ilgio balso įrašą, panaudojant kuo daugiau skirtingų diktorių įgarsinimų. Programos pradžioje inicijuojami pradiniai kintamieji: reikalingas įrašo ilgis bei „raw“ formato failų vieta kietajame diske. Tuomet nuskaitomas kiekvieno failo pavadinimas ir išsaugomas į sąrašo kintamąjį A, kuris yra kopijuojamas diktorių skaičiaus ir failų gavimo apdorojimui skirtam kintamajam B. Toliau seka balso įrašo sudarymo algoritmas – tikrinama ar liko įrašu su skirtingų diktorių įgarsinimais, jeigu taip – šių failų pavadinimai išsaugomi kintamajame sąraše C ir naudojami galutinio įrašo rezultatui gauti. Tuomet visi tų pačių diktorių įgarsinti įrašai pašalinami iš sąrašo B, padidinamas naudotų skirtingų diktorių įgarsinimų skaitliukas. Jeigu neliko balso įrašų B sąraše, tuomet atsitiktinai gaunamas failas iš sąrašo A ir tikrinama ar įrašas egzistuoja kintamajame C. Tai kartojama kol gaunamas reikalingo ilgio failas. Galutinis įrašas sudaromas pasitelkiant *SoX* programos paketą.

2.3.2. Naujos duomenų bazės sukūrimas

Naujos duomenų bazės, skirtos Codec2 1200 bps balso suspaudimo algoritmui, sukūrimui naudojama programa [39], kuri buvo panaudota šio koderio originalių LSF koeficientų klasterizavimui. Programa sukuria tris kodų knygas, jos gaunamos mišraus vektorių klasterizavimo metodu, kurio blokinė diagrama pavaizduota 2.8 pav.



2.8 pav. Mišraus klasterizavimo metodo blokinė diagrama

Šiuo metodu pirmos pakopos vektorių likutis e_1 padalinamas į dvi lygias dalis e_{11} ir e_{12}, kuriomis naudojantis sekančioje pakopoje sudaromos dvi naujos kodų knygos. Kvadratinės klaidos iškraipymas pirmoje pakopoje skaičiuojamas pagal *Euklido* atstumą:

$$f_i = \sum_{j=1}^{p}(x_{ij} - c_{ij})^2 \quad , \tag{2}$$

čia p yra vektoriaus ilgis, x_{ij} j LSF vektoriaus komponentas, c_{ij} – tikrinamo LSF vektoriaus komponentas.

Mišraus klasterizavimo metodo antroje pakopoje kvadratinės klaidos iškraipymas skaičiuojamas pagal pasvertą *Euklido* atstumą:

$$f_i = \sum_{j=1}^{p} w_j(x_{ij} - c_{ij})^2 \quad , \tag{3}$$

čia p – vektoriaus ilgis, x_{ij} – j LSF vektoriaus komponentas, c_{ij} – tikrinamo LSF vektoriaus komponentas, w_i – kiekvienam vektoriui suteikiamas svoris, kuris skaičiuojamas atvirkštinio harmoninio vidurkio metodu:

$$w_i = \frac{1}{x_i - x_{i-1}} + \frac{1}{x_{i+1} - x_i} \quad , \tag{4}$$

čia x_i – LSF vektoriaus reikšmė.

Aprašyto skaičiavimo metodo esmė – kai du LSF vektoriai arti vienas kito, tai kalbos spektre šioje pozicijoje gaunamas pikas, todėl jam turėtu būti priskiriamas didesnis svoris [40].

Po kiekvienos pakopos kodų knygos sudarymo, skaičiuojama vidutinės kvadratinės šaknies klaida(angl. *root mean square error* (RMSE):

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(x_i - c_i)^2}{n}} \quad , \tag{5}$$

čia n – vektoriaus ilgis, x_i – originalus LSF vektoriaus komponentas, c_i – artimiausias kodų knygoje surastas LSF vektoriaus komponentas

LSF koeficientai gaunami naudojant „c2sim“ programą Codec2 pakete. Koeficientų gavimui pirmiausią reikia sudaryti įrašą, kurio ilgio pasirinkimui, buvo atliekamas PESQ balo priklausomybės nuo įrašo ilgio tyrimas. Naudojantis 1800,78 s, 3601,43 ir 7204,13 įrašų ilgiais, apmokyta Codec2 1200 bps kodų knyga ir gaunamas PESQ balas. Gauti rezultatai pateikti 8 lentelėje.

8 lentelė. PESQ balo priklausomybė nuo įrašo ilgio

Įrašo ilgis, s	**Diktorių skaičius**	**$RMSE_1$**	**$RMSE_2$**	**PESQ balas**
1800,78	298	0,062583	0,047543	2,235
3601,43	374	0,062303	0,047212	2,239
7204,13	374	0,062081	0,04697	2,246

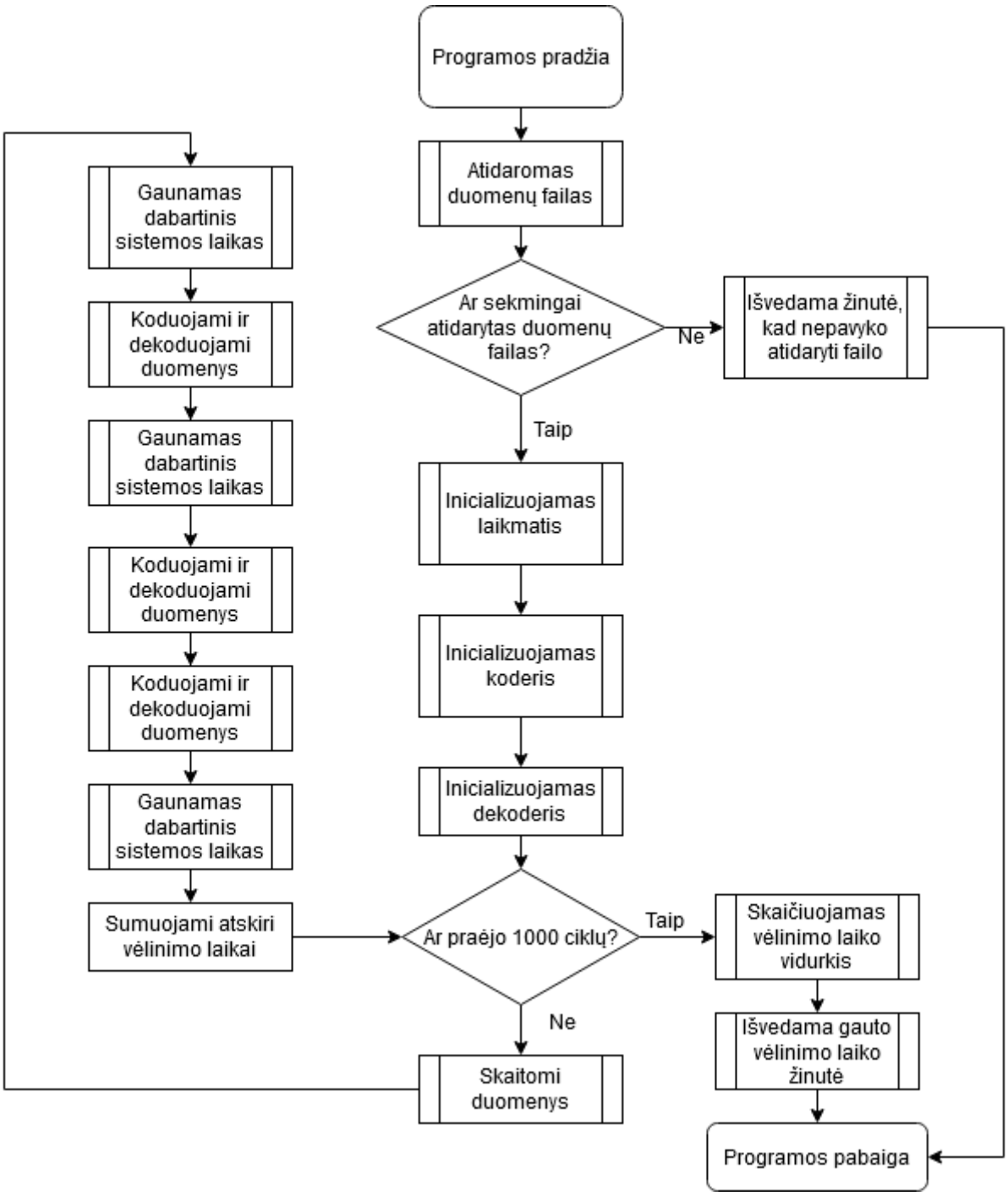
Iš lentelės duomenų pastebima, kad ilgesnis įrašas turi didesnį PESQ balą, tačiau skirtumas tarp pusvalandžio ir dviejų valandų įrašo tėra 0,011 balo. Tiek pirmos pakopos $RMSE_1$, tiek antros pakopos $RMSE_2$ reikšmių skirtumas tarp įrašų siekia tik iki 0,00057. Taigi, galutinei kodų knygai sudaryti naudojamas 7204,13 s trukmės įrašas.

2.4. Balso suspaudimo algoritmų greičio ir signalo vėlinimo palyginimo metodas

Balso suspaudimo algoritmų greičio palyginimui pasirinktos trys skirtingos sistemos. Kiekviena sistema turi skirtingą procesorių: STM32F429 maketas su vidutinės spartos ARM Cortex – M4 180 MHz procesoriumi [41], STM32F723 maketas su vidutinės spartos ARM Cortex – M7 216 MHz [42] ir BeagbleBone Black su didelės spartos ARM Cortex – A8 1 GHz procesoriumi [43]. Didžiausias BeagleBone Black maketo skirtumas nuo kitų sistemų yra tai, kad jis naudoja Linux Debian operacinę sistemą, o STM32F723 tai, kad turi slenkamo kablelio skaičiavimo modulį.

Tiek STM32F429, tiek STM32F723 maketai sukonfigūruoti maksimaliam procesoriaus dažniui pasiekti. Kad būtu išnaudojama didžiausia procesoriaus galia, BeagleBone Black sistemoje yra išjungiama grafinė aplinka. Kiekvienam balso suspaudimo algoritmui kompiliuojant pasirenkamas aukščiausias optimizacijos lygis.

BeagleBone Black sistemoje naudojamas balso suspaudimo metodų vėlinimo skaičiavimo algoritmas pateiktas 2.9 pav.



2.9 pav. Balso vėlinimo skaičiavimo algoritmas *Linux* operacinėje sistemoje

Programos pradžioje atidaromas duomenų failas – 8640 imties balso įrašas, kuriame sakomas žodis „testas“. Kadangi matuojamas vėlinimas, tai skaičiuojama kiek truko užkoduoti ir dekoduoti vieną, specifiniam koderiui skirtą, imties bloką. Matavimas atliekamas naudojantis *Linux* operacinės sistemos laikmačiais, specifiškai *CLOCK_MONOTONIC_RAW* [44]. Buvo pastebėta, kad matuojant algoritmo trukmę, gaunamos nevienodos reikšmės, todėl remiantis [45] straipsnio rezultatais, užtrukęs laikas skaičiuojamas pagal formulę:

$$T_e = \frac{T_1 - T_2}{N_1 - N_2} \quad , \tag{6}$$

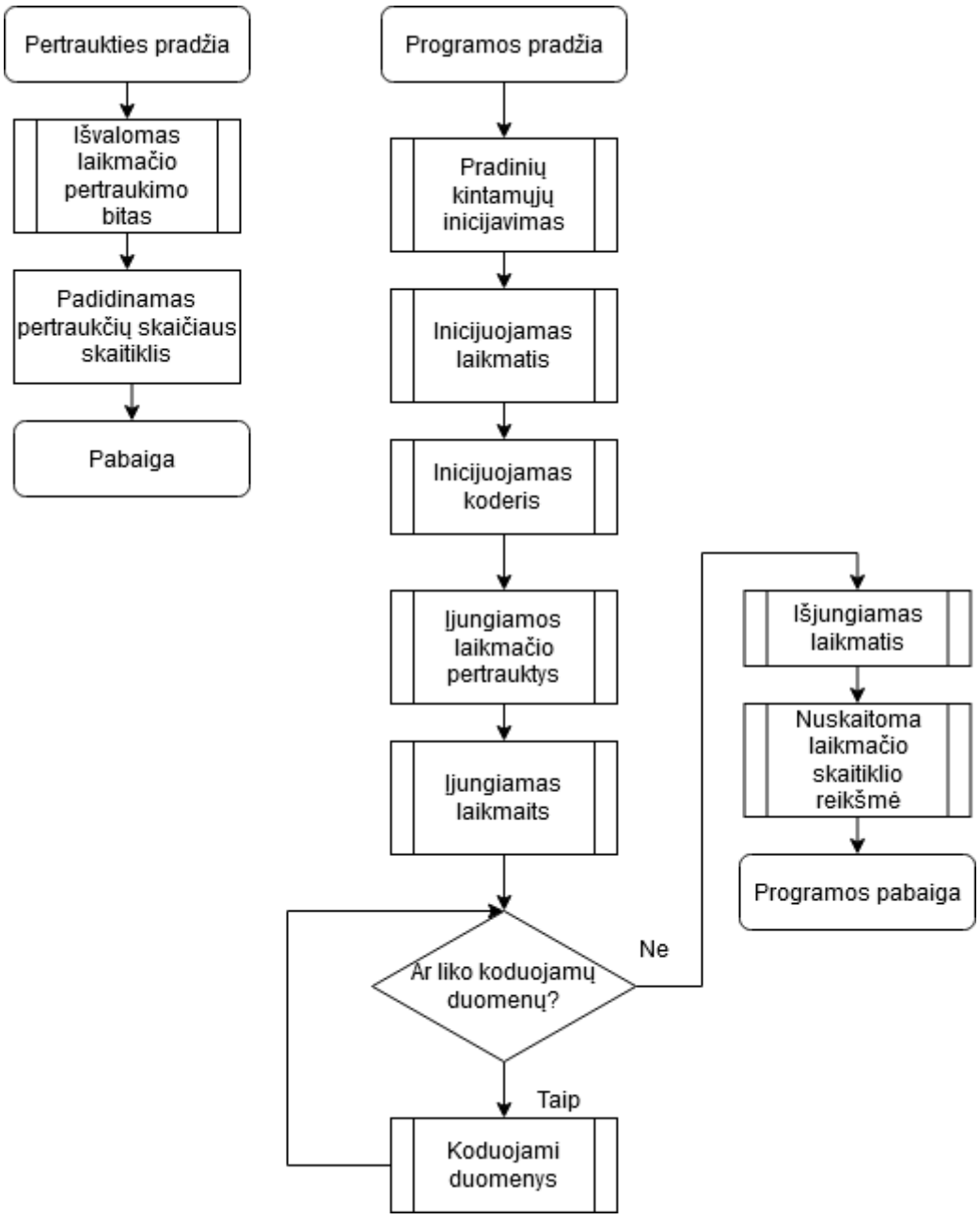
čia T_e – algoritmo trukmę, T_1 – matavimo trukmė, kai jis atliekamas N_1 kartų, T_2 – matavimo trukmė, kai jis atliekamas N_2 kartų

Pasirinkus $N_1 = 1$ ir $N_2 = 2$ bei eksperimentą atliekant K kartų gaunama, kad bendro vėlinimo trukmė skaičiuojama pagal:

$$T_e = \frac{\sum_{i=1}^{K} T_{i2} - T_{i1}}{K} \quad , \tag{7}$$

Alternatyviai matuojama atskira kodavimo ir dekodavimo trukmė, tik šiuo atveju naudojamas visas balso įrašas. 8640 imties įrašo ilgis pasirinktas todėl, kad toks yra mažiausias bendras visų kodavimo algoritmų vardiklis. Tai leidžia balso spūdos metodus lyginti vienodomis sąlygomis.

STM32F429 ir STM32F723 balso suspaudimo kodavimo laiko skaičiavimo algoritmas pavaizduotas 2.10 pav.



2.10 pav. Kodavimo trukmės skaičiavimo algoritmas įterptinėje sistemoje

Pagrindinis skirtumas, lyginant du trukmės matavimo algoritmus, kurie pavaizduoti 2.9 pav ir 2.10 pav., yra tai, kad 2.10 pav. pateikto algoritmo STM32F429 ir SMT32F723 sistemose matavimui naudojama laikmačio pertrauktis, kuri sukonfigūruota veikti 10 kHz dažniu. Algoritmo trukmė gaunama skaičiuojant pagal formulę:

$$T = pertraukčių\ skaičius * \frac{Skatiklio\ periodas+1}{APB2_{clk}} + \frac{Skaitiklio\ reikšmė}{APB2_{clk}}, \quad (8)$$

čia T – algoritmo trukmė sekundėmis, $APB2_{clk}$ – skaitiklio veikimo dažnis, Skaitiklio periodas – nustatytas į 21599 STM32F723 maketui ir į 17999 STM32F429 maketui; skaitiklio reikšmė – dabartinė laikmačio skaitiklio reikšmė nuskaitymo momentu. $APB2_{clk}$ pasirinktas 180 MHz naudojant M4 procesorių, o 216 MHz naudojant procesorių M7.

2.5. Skyriaus apibendrinimas

Šiame skyriuje aprašomas kodavimo ir dekodavimo blokų sukūrimas koderių bibliotekų veikimo tikrinimui GNURadio modeliavimo aplinkoje. Naudojantis šiais blokais galima realiu laiku klausytis balso suspaudimo algoritmų rezultato, matyti laikinę ir GFT diagramas

Kadangi tiriami įrašai nėra užteršti triukšmo, tai balso suspaudimo algoritmų subjektyviems bandymams atlikti pasirinkti ITU – T P.800 standarte aprašyti kalbos kokybės ir suprantamumo įvertinimai. Jiems atlikti sudaryta speciali programa, kuria naudojantis kiekvieną įrašą galima lengvai įvertinti. Balso suspaudimo algoritmų objektyviems testams atlikti, pasirinktas ITU – T P.862 standarte aprašytas PESQ vertinimo metodas, juo naudojantis nereikia keisti signalų lygio, jų filtruoti ar sinchronizuoti.

Norint pagerinti kalbos kokybę ir suprantamumą, pasirinktame Codec2 1200 bps koderyje sukuriama nauja kodų knyga, pasitelkiant *LIEPA* specialią lietuvių kalbos įrašų biblioteką. Kodų knygos kūrimui panaudota mišraus klasterizavimo programa, kurios kvadratinės klaidos iškraipymas matuojamas remiantis *Euklido* ir pastverto *Euklido* atstumais. Atliktas PESQ rezultato priklausomybės nuo įrašo ilgio tyrimas parodė, kad ilgesnis įrašas didelės įtakos objektyvaus vertinimo balui neturi, skirtumas tėra iki 0,011 balo lyginant pusvalandžio ir dviejų valandų įrašus.

Balso suspaudimo algoritmų greičio ir signalo vėlinimo įvertinimams atlikti, pasirinkti trys skirtingi procesoriai: vidutinės spartos ARM Cortex–M4 180 MHz, vidutinės spartos ARM Cortex–M7 216 MHz, turinti atskirą FPU elementą bei didelės spartos ARM Cortex–A8 1 GHz procesorius, kurio makete naudojama *Linux Debian* sistema. Kad būtų galima įvertinti greitį ir vėlinimą, pasitelkiami specialūs laikmačiai. Kiekvienas algoritmas buvo vertinimas vienodoms sąlygoms – pasirinktas didžiausias optimizacijos lygis, greičio palyginimui naudojamas vienodo ilgio įrašas, o vėlinimo – mažiausias galimas imties blokas.

3. Tyrimo rezultatai

3.1. Balso suspaudimo algoritmų subjektyvūs įvertinimai

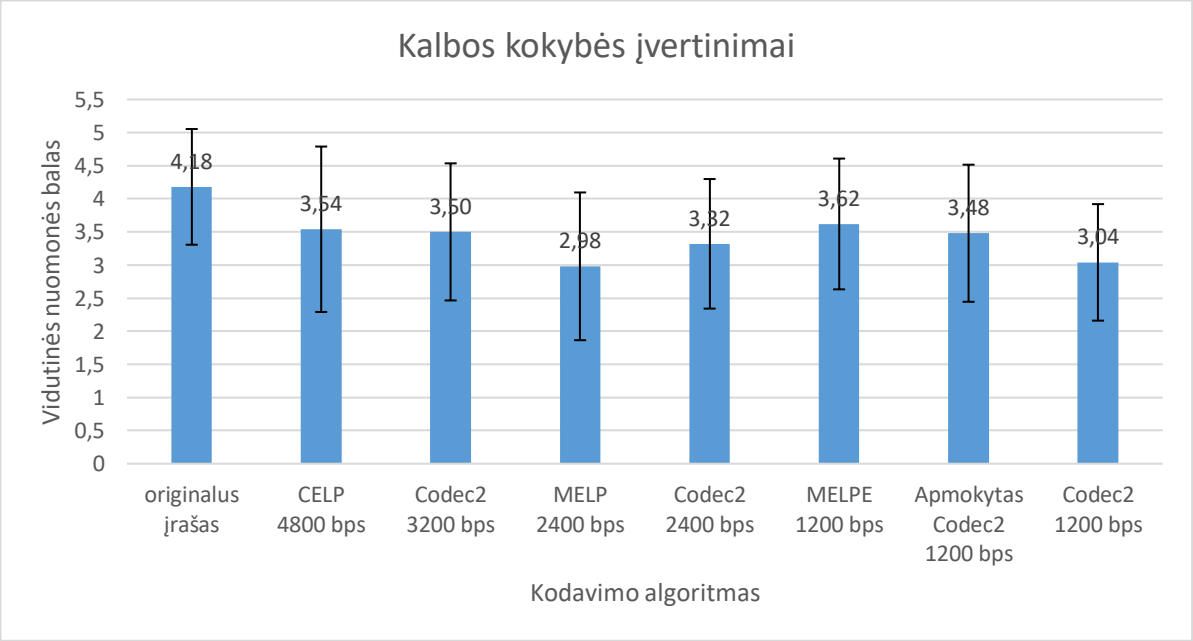
Didelės spūdos balso suspaudimo algoritmų subjektyvus vertinimas buvo atliekamas kiekvienam subjektui išsiuntus sudaryta programėlę, kurios rezultatai grąžinami apdorojimui. Iš viso apklausti 25 subjektai, tarp kurių 15 vyrų ir 10 moterų.

Testą sudaro vienas vyro ir vienas moters balsų įrašai, kurie nebuvo panaudoti naujos duomenų bazės kūrime. Abu failai apdoroti naudojantis CELP, Codec2, MELP, MELPe balso suspaudimo algoritmais, tarp jų apmokytas ir originalus Codec2 1200 bps. Gauti kalbos kokybės vidutinės nuomonės balai ir standartiniai nuokrypiai matomi 9 lentelėje, o visi įvertinimai pateikti 1 priede.

9 lentelė. kalbos kokybės MOS balai ir standartiniai nuokrypiai

Balso suspaudimo algoritmas	MOS balas	Standartinis nuokrypis
Originalus įrašas	4,18	0,87
CELP 4800 bps	3,54	1,25
Codec2 3200 bps	3,50	1,06
MELP 2400 bps	2,98	1,17
Codec2 2400 bps	3,32	0,98
MELPe 1200 bps	3,62	0,99
Apmokytas Codec2 1200 bps	3,48	1,04
Codec2 1200 bps	3,04	0,88

Kalbos kokybės įvertinimai parodomi 3.1 paveikslėlyje esančioje diagramoje.



3.1 pav. Kalbos kokybės įvertinimai naudojant skirtingus suspaudimo algoritmus

Iš lentelės ir grafiko duomenų pastebima, kad, kaip ir tikėtasi, originalūs įrašai turi aukščiausią balą bei žemiausią standartinį nuokrypį. Gauti rezultatai rodo, jog geriausias balso suspaudimo algoritmas, vertinant pagal MOS skalę, yra MELPe 1200 bps (3,62±0,99 balo), nedaug nuo jo atsilieka lietuvių

kalbos duomenų bazę naudojantis, tokią pačią spūdą pasiekiantis, Codec2 1200 bps algoritmas (3,48±1,04 balo). Blogiausi rezultatai – Codec2 1200 bps ir MELP 2400 bps balso suspaudimo metodų (2,98±1,17 ir 3,04±0,88 balo), tačiau Codec2 1200 bps įvertinimas stabiliausias, turintis 0,879 standartinį nuokrypį.

Kiekvieno įrašo standartinis nuokrypis didesnis negu 0,8, tai gali būti paaiškinama tuo, kad balso įrašai buvo klausomi skirtingomis sąlygomis, nekontroliuota nei subjekto aplinka, nei klausymo aparatūra.

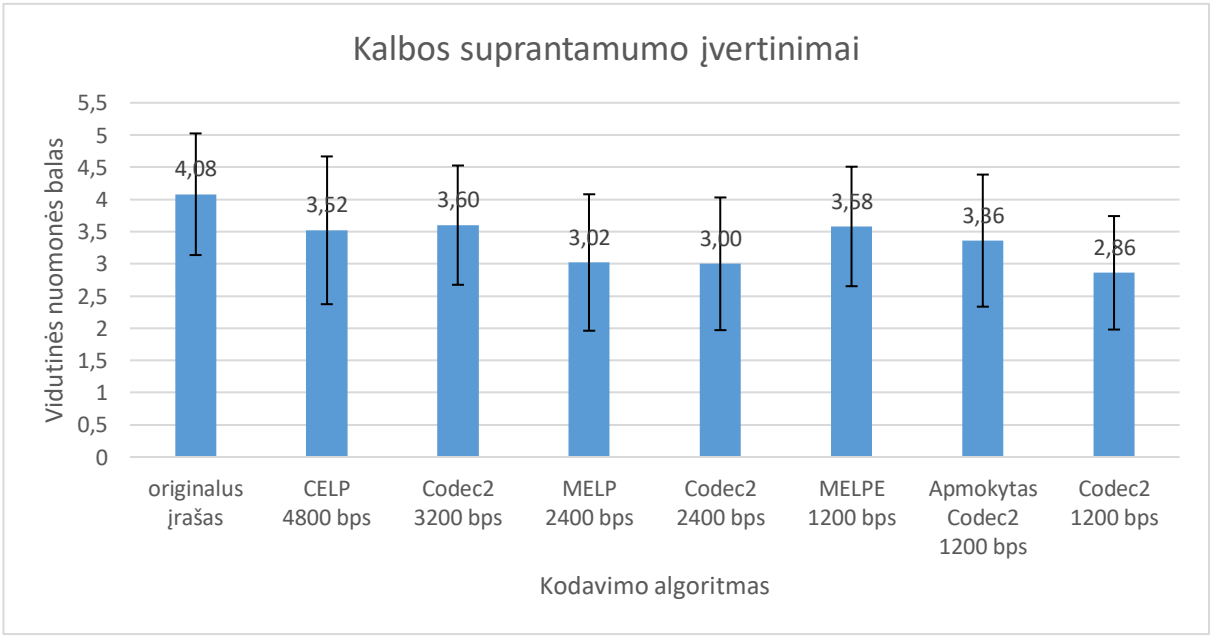
Lyginant lietuvių kalbos duomenų bazę naudojanti ir originalų Codec2 1200 bps balso suspaudimo algoritmus matoma, kad apmokyto algoritmo kalbos suprantamumo vertinimas 0,44 balo aukštesnis, tačiau jo standartinis nuokrypis didesnis(0,16). Taigi, galima manyti, jog verta naudoti kodų knygą, sudarytą iš lietuvių kalbos įrašų.

Vidutiniai kalbos suprantamumo balai ir standartiniai nuokrypiai kiekvienam balso suspaudimo algoritmui pateikti 10 lentelėje, o visi įvertinimai 2 priede.

10 lentelė. kalbos suprantamumo MOS balai ir standartiniai nuokrypiai

Balso suspaudimo algoritmas	**balas**	**Standartinis nuokrypis**
Originalus įrašas	4,08	0,94
CELP 4800 bps	3,52	1,15
Codec2 3200 bps	3.6	0,93
MELP 2400 bps	3,02	1,06
Codec2 2400 bps	3	1,03
MELPe 1200 bps	3,58	0,93
Apmokytas Codec2 1200 bps	3,36	1,03
Codec2 1200 bps	2,86	0,88

Kalbos suprantamo įvertinimai grafiškai atvaizduoti 3.2 pav.



3.2 pav. Kalbos suprantamumo įvertinimai naudojant skirtingus suspaudimo algoritmus

Iš gautu rezultatų pastebima, kad originalių įrašų balai (4,08 ± 0,944) labiau išsibarstę nei Codec2 1200 bps (2,88 ± 0,88). Kalbos suprantamumas geriausiai įvertintas Codec2 3200 bps kodavimo algoritme (3,6 ± 0,93 balo) bei MELPe 2400 bps (3,58 ± 0,93 balo), prasčiausiai – Codec2 1200 bps (2,86 ± 0,88 balo). Praktiškai vienodas kalbos suprantamumas gaunamas vienodą spūdą turinčių MELP 2400 bps ir Codec2 bps balso suspaudimo algoritmų.

Lyginant apmokytą ir originalų Codec2 1200 bps algoritmus, pastebimas 0,5 balo aukštesnis įvertinimas, kai naudojama kodų knyga, sudaryta iš lietuvių kalbos balso įrašų. Taigi, tiek kalbos kokybės, tiek kalbos suprantamumo įvertinimuose, apmokyto Codec2 1200 bps rezultatai yra geresni.

3.2. Balso suspaudimo algoritmų objektyvus vertinimas

Naudojantis PESQ objektyvaus vertinimo metodu, buvo gaunamas MOS–LQO balas, tikrinant kiekvieną subjektyviame teste esantį balso įrašą. Vyro balso įrašo PESQ vertinimo rezultatai yra pateikti 11 lentelėje, o moters 12 lentelėje.

11 lentelė. PESQ balai, naudojant vyro balso įrašą

Balso suspaudimo algoritmas	**balas**
CELP 4800 bps	2,966
Codec2 3200 bps	2,805
MELP 2400 bps	1,995
Codec2 2400 bps	2,696
MELPe 1200 bps	2,542
Apmokytas Codec2 1200 bps	2,607
Codec2 1200 bps	2,561

Iš gautu rezultatų pastebima, kad bendrai visų balso suspaudimo metodų balai yra žemesni nei subjektyvaus vertinimo, jų skirtumas siekia iki 1 balo MELPe 1200 bps, apmokyto Codec2 1200 bps ir MELP 2400 bps algoritmų .

Objektyvaus metodo rezultatai rodo, kad kuo mažesnė spūda, tuo aukštesnis balas, tokia tendencija yra tiek vyriško, tiek moteriško balsų įrašuose. Abiejų garso įrašų rezultatai ganėtinai panašūs, didžiausias skirtumas (0,518 balo) naudojant MELP 2400 bps suspaudimo algoritmą.

12 lentelė. PESQ balai, naudojant moters balso įrašą

Balso suspaudimo algoritmas	**balas**
CELP 4800 bps	3,01
Codec2 3200 bps	3,089
MELP 2400 bps	2,513
Codec2 2400 bps	2,787
MELPe 1200 bps	2,586
Apmokytas Codec2 1200 bps	2,561
Codec2 1200 bps	2,536

Lyginant apmokyto ir originalaus Codec2 1200 bps balso suspaudimo metodus pastebima, kad, pagal objektyvų įvertinimą, skirtumas tarp lietuvių kalbą naudojančio ir originalaus koderio nėra toks

ryškus kaip pagal subjektyvų įvertinimą. Tačiau tiek vyriško, tiek moteriško balso įrašuose, aukštesnis balas yra apmokyto suspaudimo algoritmo (atitinkamai 0,046 ir 0,024 balo).

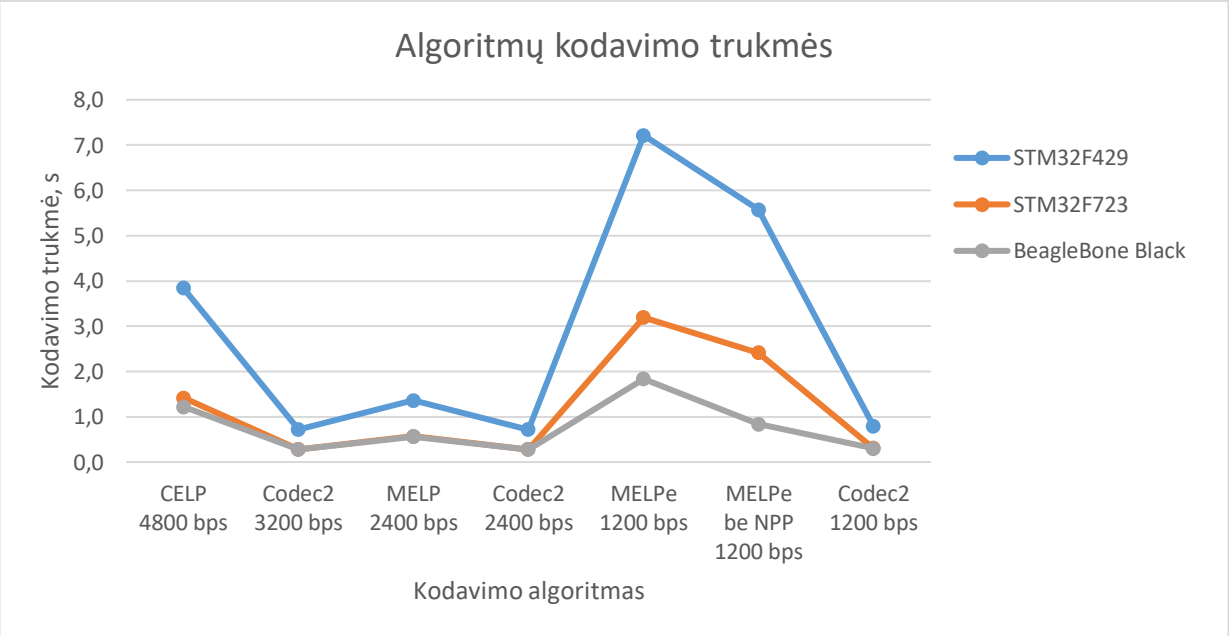
3.3. Balso suspaudimo algoritmų greičio ir signalo vėlinimo palyginimas

Didelės spūdos balso suspaudimo algoritmų greitis ir vėlinimas buvo tikrinamas naudojant STM32F429, STM32F723 ir BeagleBone Black sistemas. Codec2 1200 bps balso suspaudimo algoritmas patikrintas pasitelkiant atnaujintą lietuvių kalbos duomenų bazę. Kiekviename balso suspaudimo algoritme matuojama kodavimo ir dekodavimo imtis buvo vienoda, t.y. 8640 imtis. Gauti kodavimo trukmės rezultatai pateikti 13 lentelėje.

13 lentelė. Balso suspaudimo algoritmų kodavimo trukmės

Kodavimo algoritmas	**Kodavimo trukmė, s**		
	STM32F429	STM32F723	BeagleBone Black
CELP 4800 bps	3,83966	1,417	1,21568
Codec2 3200 bps	0,71791	0,27811	0,27682
MELP 2400 bps	1,35982	0,56833	0,55793
Codec2 2400 bps	0,71948	0,27839	0,27734
MELPe 1200 bps	7,21485	3,19034	1,83316
MELPe be NPP 1200 bps	5,57322	2,41353	0,83939
Codec2 1200 bps	0,79134	0,30509	0,30120

Gauti kodavimo trukmės rezultatai kiekviename balso suspaudimo algoritme pavaizduoti 3.3 paveikslėlyje esančiame grafike.



3.3 pav. Algoritmų kodavimo trukmės

Grafike stebimas pastovus dėsningumas tarp skirtingų sistemų, t.y. MELPe balso suspaudimo algoritmas kodavimą atliko lėčiausiai visuose tirtuose sistemose, net kai yra išjungiamas įrašo filtravimas (3.3 grafike ir 12 lentelėje „MELPe be NPP 1200 bps“). Signalas greičiausiai koduojamas

Codec2 3200 bps koderyje, jis truko 0,27682 s naudojant Arm Cortex – A8, 0,27811 s Arm Cortex – M7 ir 0,71791 s Arm Cortex – M4 procesorių.

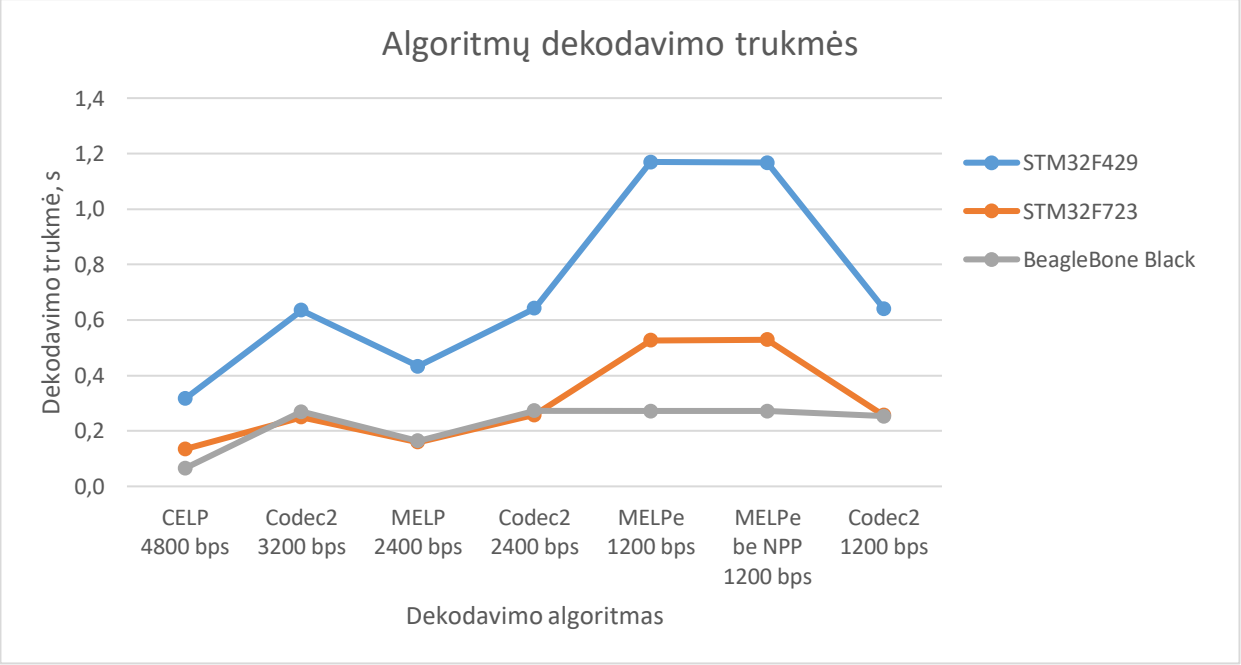
Matoma, kad STM32F723, naudojanti Arm Cortex – M7 procesorių sistemą, MELP ir Codec2 kodavimo metoduose praktiškai neatsiliko nuo BeagleBone maketo su ARM Cortex – A8, skirtumas šiuose algoritmuose siekia iki 0,01 s.

Balso suspaudimo algoritmų išmatuotos dekodavimo trukmės pateiktos 14 lentelėje.

14 lentelė. Balso suspaudimo algoritmų dekodavimo trukmės

Dekodavimo algoritmas	**Dekodavimo trukmė, s**		
	STM32F429	STM32F723	BeagleBone Black
CELP 4800 bps	0,31629	0,1347	0,0655
Codec2 3200 bps	0,63443	0,24949	0,26893
MELP 2400 bps	0,43278	0,15903	0,16357
Codec2 2400 bps	0,64189	0,25744	0,27219
MELPe 1200 bps	1,16985	0,52627	0,27178
MELPe be NPP 1200 bps	1,16742	0,52848	0,27178
Codec2 1200 bps	0,64086	0,25677	0,25266

Gauti dekodavimo trukmės rezultatai kiekviename balso suspaudimo algoritme pavaizduoti 3.4 paveikslėlyje esančiame grafike.



3.4 pav. Algoritmų dekodavimo trukmės

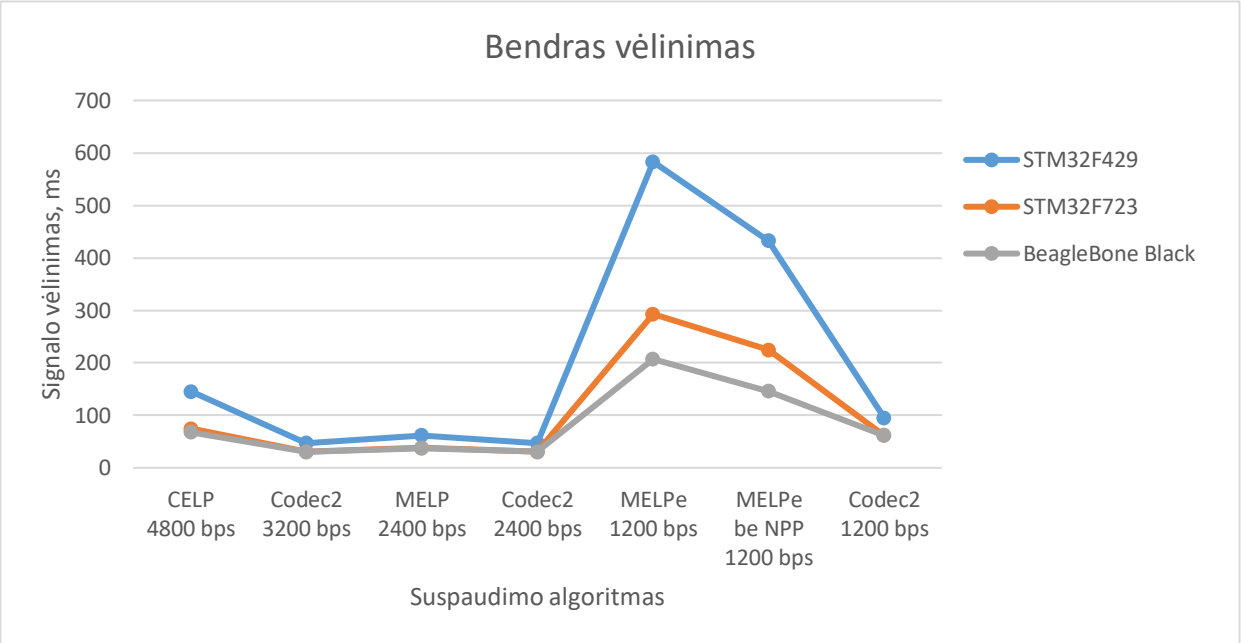
Vertinant lentelės ir grafiko duomenis pastebima, kad dekodavimas Codec2 balso suspaudimo algoritmuose užtrunka beveik tokį patį laiką, skirtumas siekia iki 0,01 s. Duomenys sparčiausiai dekoduojami CELP 4800 bps koderyje, o lėčiausiai MELPe 1200 bps visose tirtuose sistemose.

Balso suspaudimo algoritmų bendras kodavimo ir dekodavimo vėlinimas pateiktas 15 lentelėje. Šiuo atveju įvertinamas reikalingas imties ilgis – jis į sistemą įneša papildoma signalo vėlinimą.

15 lentelė. Balso suspaudimo algoritmų vėlinimai

Suspaudimo algoritmas	**Imties ilgis, ms**	**Bendras vėlinimas, ms**		
		STM32F429	STM32F723	BeagleBone Black
CELP 4800 bps	30	145,57	74,54	67,85
Codec2 3200 bps	20	46,99	30,84	30,59
MELP 2400 bps	22,5	61,61	38,27	37,57
Codec2 2400 bps	20	47,06	30,84	30,77
MELPe 1200 bps	67,5	582,91	292,97	207,14
MELPe be NPP 1200 bps	67,5	432,69	224,6	146,28
Codec2 1200 bps	40	95,41	61,98	61,83

Gauti bendro vėlinimo matavimo rezultatai kiekviename balso suspaudimo algoritme pavaizduoti 3.5 paveikslėlyje esančiame grafike.



3.5 pav. Balso suspaudimo algoritmų bendras vėlinimas

Iš lentelės ir grafiko duomenų grafiko pastebima, kad visose tirtose sistemose didžiausias signalo vėlinimas gaunamas naudojant MELPe 1200 bps balso suspaudimo algoritmą, mažiausias – Codec2 3200 bps. Bendras vėlinimas tarp ST32F723 ir BeagleBone Black sistemų praktiškai nesiskiria visuose balso suspaudomo algoritmuose išskyrus MELPe 1200 bps, kurio skirtumas yra iki 85,83 ms.

Norint palaikyti kokybišką pokalbį pagal ITU – T G.114 standartą [46], [46], bendras visos sistemos vėlinimas neturėtu viršyti 150 ms. Taigi, nedarant papildomų optimizacijų, ši sąlyga negali būti išpildyta naudojant MELPe 1200 bps balso suspaudimo metodą visose tirtose sistemose, išskyrus kai yra išjungiamas filtras, šiuo atveju MELPe 1200 bps metodą galima naudoti su ARM Cortex – A8 procesoriumi.

3.4. Skyriaus apibendrinimas

Šiame skyriuje pateikti atliktų balso suspaudimo algoritmų subjektyvių ir objektyvių tyrimų rezultatai, greičio ir vėlinimo palyginimai. Subjektyvūs ir objektyvūs testai atlikti vyro ir moters balso įrašus apdorojus CELP, MELP, MELPe, Codec2 suspaudimo metodais, tarp jų ir apmokytas Codec2 1200 bps algoritmas. Iš viso buvo apklausti 25 subjektai.

Gauti tyrimo rezultatai rodo, kad kalbos kokybė geriausia įvertinta (3,62 ± 0,987 balo) MELPe 1200 bps balso suspaudimo metodo, prasčiausiai – Codec2 1200 bps ir MELP 2400 bps (atitinkamai 2,98 ± 1,116 ir 3,04 ± 0,879 balo). Kalbos suprantamumas geriausiai įvertintas Codec2 3200 bps ir MELPe 1200 bps balso suspaudimo algoritmų (atitinkamai 3,6 ± 0,926 ir 3,58 ± 0,929 balo), prasčiausiai – Codec2 1200 bps ir Codec2 2400 bps (atitinkamai 2,86 ± 0,88 ir 3 ± 1,03 balo). Lietuvių kalbos įrašų biblioteką naudojantis Codec2 1200 bps algoritmas turi geresnius rezultatus tiek kalbos kokybės (0,44 balo skirtumas), tiek kalbos suprantamumo(0,5 balo skirtumas) įvertinimuose. Visuose įvertinimuose standartinis nuokrypis yra didesnis nei 0,8, tai gali būti paaiškinama nekontroliuojamoms subjektų klausymo sąlygomis.

Objektyvaus metodo bandymų rezultatui gauti, panaudoti vyro ir moters įgarsinti įrašai, su kuriais buvo atlikti subjektyvūs bandymai. Gautuose įvertimuose pastebimas iki vieno balo skirtumas lyginant objektyvius testus su subjektyviais bandymais. Matoma tendencija, kad mažesnės spūdos balso suspaudimo algoritmas turi aukštesnį balą. Lyginant lietuvių kalbos įrašų biblioteką naudojantį ir originalų Codec2 1200 bps algoritmus, pastebimas mažesnis balo skirtumas(0,046 vyriško balso ir 0,024 moteriško balso įrašuose), nei subjektyviuose testuose. Taigi, tiek subjektyvūs, tiek objektyvūs balso suspaudimo įvertinimo metodai rodo, kad verta naudoti kodų knygą sudaryta iš lietuvių kalbos įrašų.

Kodavimas visuose procesoriuose greičiausiai atliekamas naudojant Codec2 balso suspaudimo algoritmus, lėčiausiai – MELPe. Dekodavimas yra sparčiausias naudojant CELP metodą. Didžiausias signalo vėlinimas gaunamas naudojant MELPe balso suspaudimo algoritmą, jis siekia iki 582,91 ms. Mažiausias signalo vėlinimas visuose tirtuose procesoriose yra naudojant Codec2 3200 bps balso suspaudimo algoritmą. MELPe balso suspaudimo algoritmo bendro vėlinimo skirtumas tarp ARM Cortex M4 ir likusių sistemų gali būti paaiškinamas tuo, kad jis neturi slenkančio kablelio skaičiavimo modulio. Remiantis ITU – T G.114 standarto rekomendacijomis, maloni ir sklandi komunikacija negali vykti visuose pasirinktuose procesoriuose naudojant MELPe algoritmą, jeigu nėra išjungimas NPP filtras.

Išvados

1. Iš balso suspaudimo algoritmų apžvalgos nustatyta, kad tiesinio prognozavimo koeficientai naudojami kiekviename koderyje, tai yra tarsi kiekvieno metodo bazė. Didžiausias skirtumas tarp balso suspaudimo algoritmų – sužadinimo signalo ieškojimo būdai. Didelė spūda pasiekiama naudojant skirtingus vektorių klasterizavimo metodus.
2. Koderių bibliotekų tikrinimui *GNURadio* modeliavimo aplinkoje, sukurti kodavimo ir dekodavimo blokai, kuriais naudojantis patikrintas kiekvieno koderio veikimas.
3. Baigiamajame darbe, pasitelkiant LIEPA lietuvių kalbos įrašų biblioteką, apmokytas Codec2 1200 bps balso suspaudimo algoritmas. Atliktas PESQ rezultato priklausomybės nuo įrašo ilgio tyrimas rodo, kad įrašo ilgis didelės įtakos objektyviam įvertinimui neturi, kadangi skirtumas tarp pusvalandžio ir dviejų valandų įrašo yra tik 0,011 balo.
4. Gauti subjektyvaus įvertinimo tyrimo rezultatai rodo, kad kalbos kokybė geriausia įvertinta (3,62 ± 0,987 balo) MELPe 1200 bps balso suspaudimo metodo, prasčiausiai – Codec2 1200 bps ir MELP 2400 bps (atitinkamai 2,98 ± 1,116 ir 3,04 ± 0,879 balo). Kalbos suprantamumas geriausiai įvertintas Codec2 3200 bps ir MELPe 1200 bps balso suspaudimo algoritmų (atitinkamai 3,6 ± 0,926 ir 3,58 ± 0,929 balo), prasčiausiai – Codec2 1200 bps. Visuose gautuose įvertinimuose standartinis nuokrypis didesnis nei 0,8, tai gali būti paaiškinama nekontroliuojamomis subjektų klausymo sąlygomis. Gautuose objektyviuose tyrimo rezultatuose pastebimas iki vieno balo skirtumas lyginant objektyvius testus su subjektyviais bandymais. Remiantis kalbos kokybės(0,44 balo skirtumas), kalbos suprantamumo(0,5 balo skirtumas) ir PESQ(iki 0,046 balo skirtumas) įvertinimais nustatyta, kad lietuvių kalbos įrašų biblioteką naudojantis Codec2 1200 bps algoritmas yra geresnis lyginant su originaliu
5. Gauti tyrimo rezultatai rodo, kad kodavimas visuose procesoriuose greičiausiai atliekamas naudojant Codec2 balso suspaudimo algoritmus, lėčiausiai – MELPe 1200 bps. Dekodavimas sparčiausias, kai naudojamas CELP 4800 bps metodas. Didžiausias signalo vėlinimas gaunamas pasitelkiant MELPe 1200 bps balso suspaudimo algoritmą – jis siekia iki 582,91 ms. Mažiausias signalo vėlinimas visuose tirtuose procesoriose gaunamas panaudojus Codec2 3200 bps balso suspaudimo algoritmą. Remiantis ITU – T G.114 standarto rekomendacijomis, kokybiška komunikacija negali vykti visuose pasirinktuose procesoriuose naudojant MELPe 1200bps algoritmą, jeigu nėra išjungimas NPP filtras.

Literatūros sąrašas

1. D. G. Rowe, „Techniques for Harmonic Sinusoidal Coding“, p. 155.
2. M. Hasegawa-Johnson ir A. Alwan, „Speech Coding: Fundamentals and Applications“, *Wiley Encyclopedia of Telecommunications*, J. G. Proakis, Sud. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2003.
3. J. Bradbury, „Linear Predictive Coding“, p. 23.
4. L. R. Rabiner ir R. W. Schafer, „Theory and Application of Digital Speech Processing by“. /paper/Theory-and-Application-of-Digital-Speech-Processing-Rabiner-Schafer/66d60f4118a8b3992afbe6a8308ff90b3925b073 (žiūrėta birž. 01, 2020).
5. T. Ogunfunmi ir M. Narasimha, *Principles of Speech Coding*. CRC Press, 2010.
6. M. Tandel, V. Shah, ir B. Patel, „Implementation of CELP CODER and to evaluate the performance in terms of bit rate, coding delay and quality of speech“, *2011 3rd International Conference on Electronics Computer Technology*, bal. 2011, t. 6, p. 86–89, doi: 10.1109/ICECTECH.2011.5942056.
7. M. Mulye ir S. K. Jagtap, „Overview of Code Excited Linear Predictive Coder“, 2014. /paper/Overview-of-Code-Excited-Linear-Predictive-Coder-Mulye-Jagtap/53ee76b64b4b7b185e175b2df67629caefba9ee1 (žiūrėta geg. 31, 2020).
8. J. Gibson, „Mutual Information, the Linear Prediction Model, and CELP Voice Codecs“, *Information*, t. 10, nr. 5, p. 179, geg. 2019, doi: 10.3390/info10050179.
9. A. Goalic, J. Trubuil, ir N. Beuzelin, „Long range real-time underwater acoustic communication at low bit rate with channel coding protection“, *MILCOM 2008 - 2008 IEEE Military Communications Conference*, lapkr. 2008, p. 1–7, doi: 10.1109/MILCOM.2008.4753059.
10. E. C. Tan ir T. T. Teo, „REAL-TIME IMPLEMENTATION OF MELP VOCODER“, t. 44, nr. 3, p. 21, 2004.
11. „About MELP and MELPe Vocoder (codec)“, *MELPe STANAG-4591*. https://melpe.com/about-melpe/ (žiūrėta geg. 20, 2019).
12. G. Guilmin, F. Capman, B. Ravera, ir F. Chartier, „New Nato Stanag Narrow Band Voice Coder at 600 Bits/s“, *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, geg. 2006, t. 1, p. I–I, doi: 10.1109/ICASSP.2006.1660114.
13. „gnu.org“. https://www.gnu.org/licenses/lgpl-3.0.en.html (žiūrėta geg. 12, 2020).
14. drowe67, *drowe67/codec2*. 2020.
15. T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, ir A. Y. Wu, „An efficient k-means clustering algorithm: analysis and implementation“, *IEEE Trans. Pattern Anal. Machine Intell.*, t. 24, nr. 7, p. 881–892, liep. 2002, doi: 10.1109/TPAMI.2002.1017616.
16. O. Goldreich, *P, NP, and NP-Completeness: The Basics of Computational Complexity*, 1 edition. New York: Cambridge University Press, 2010.
17. „2.2 Vector quantization“. http://www.heikohoffmann.de/htmlthesis/node28.html (žiūrėta geg. 20, 2019).
18. L. C. W. Pols, „DISTANCE MEASURES: PHYSICAL AND PERCEPTUAL ASPECTS 1)“, p. 8.
19. „Vector Quantization Techniques 4.1 Introduction“.
20. S. Chatterjee ir T. V. Sreenivas, „Two Stage Transform Vector Quantization of LSFs for Wideband Speech Coding“, p. 5, 2006.
21. U. Sinervo, J. Nurminen, A. Heikkinen, ir J. Saarinen, „EVALUATION OF SPLIT AND MULTISTAGE TECHNIQUES IN LSF QUANTIZATION“, p. 5.
22. M. Itani ir S. Paulikas, „Influence of languages on CELP codecs performance“, *Information Technology and Control, Kaunas*, t. 37, p. 141–144, saus. 2008.

23. M. Nasief, N. Messiha, H. Mansour, ir M. Hossam, „The Effect of the Spoken Language on the Linear Prediction Vector Quantization Distortion for Linear Prediction Coders“, *Minia journal of engineering and technology*, saus. 2013.
24. S. Paulikas ir M. Itani, „Lithuanian Speech Records Database for Voice Codecs Quality Assessment“, *Information Technology and Control*, t. 39, p. 38–42, saus. 2010.
25. „GNU Radio - The Free & Open Source Radio Ecosystem · GNU Radio“. https://www.gnuradio.org/ (žiūrėta geg. 23, 2020).
26. R. C. Streijl, S. Winkler, ir D. S. Hands, „Mean Opinion Score (MOS) Revisited: Methods and Applications, Limitations and Alternatives“, *Multimedia Syst.*, t. 22, nr. 2, p. 213–227, kovo 2016, doi: 10.1007/s00530-014-0446-1.
27. A. Der, C. Yang, D.- Huang, M. Dong, ir H. Li, „Perceptual speech quality improvement for vocoder based on amplitude spectrum of residual signal“, *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, liep. 2015, p. 682–686, doi: 10.1109/ChinaSIP.2015.7230491.
28. „P.807 : Subjective test methodology for assessing speech intelligibility“. https://www.itu.int/rec/T-REC-P.807-201602-I/en (žiūrėta geg. 16, 2020).
29. T. Q. Company, „Qt Creator - A Cross-platform IDE for Application Development“. https://www.qt.io/product/development-tools (žiūrėta geg. 20, 2020).
30. K. Kokkinakis ir P. C. Loizou, „Evaluation of objective measures for quality assessment of reverberant speech“, *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, geg. 2011, p. 2420–2423, doi: 10.1109/ICASSP.2011.5946972.
31. P. Krishnamoorthy, „An Overview of Subjective and Objective Quality Measures for Noisy Speech Enhancement Algorithms“, *IETE Technical Review*, t. 28, nr. 4, p. 292, 2011, doi: 10.4103/0256-4602.83550.
32. „P.862 : Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs“. https://www.itu.int/rec/T-REC-P.862-200102-I/en (žiūrėta geg. 16, 2020).
33. W. M. Liu, K. A. Jellyman, J. S. D. Mason, ir N. W. D. Evans, „Assessment of Objective Quality Measures for Speech Intelligibility Estimation“, *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, Toulouse, France, 2006, t. 1, p. I-1225-I–1228, doi: 10.1109/ICASSP.2006.1660248.
34. „P.862.1 : Mapping function for transforming P.862 raw result scores to MOS-LQO“. https://www.itu.int/rec/T-REC-P.862.1-200311-I/en (žiūrėta geg. 20, 2020).
35. A. Olatubosun ir P. O. Olabisi, „Intrusive Assessment of Speech Quality over Wireless Networks“, p. 10.
36. „Raštija“. https://www.xn--ratija-ckb.lt/liepa/infrastrukturines-paslaugos/garsynas-liepa/7569 (žiūrėta geg. 12, 2020).
37. „LibROSA — librosa 0.7.2 documentation“. https://librosa.github.io/librosa/ (žiūrėta geg. 17, 2020).
38. „SoX - Sound eXchange | HomePage“. http://sox.sourceforge.net/ (žiūrėta geg. 17, 2020).
39. „codec2/vq_train_jvm.c at master · drowe67/codec2“. https://github.com/drowe67/codec2/blob/master/misc/vq_train_jvm.c (žiūrėta geg. 19, 2020).
40. R. C. de Lamare ir A. Alcaim, „Strategies to improve the performance of very low bit rate speech coders and application to a variable rate 1.2 kb/s codec“, *IEE Proceedings - Vision, Image and Signal Processing*, t. 152, nr. 1, p. 74–86, vas. 2005, doi: 10.1049/ip-vis:20051189.
41. „STM32F429/439“, *STMicroelectronics*. https://www.st.com/en/microcontrollers-microprocessors/stm32f429-439.html (žiūrėta geg. 18, 2020).
42. „STM32F723ZE - High-performance and DSP with FPU, ARM Cortex-M7 MCU with 512 Kbytes Flash, 216 MHz CPU, Art Accelerator, L1 cache, SDRAM - STMicroelectronics“.

https://www.st.com/en/microcontrollers-microprocessors/stm32f723ze.html (žiūrėta geg. 18, 2020).
43. „BeagleBoard.org - black“. https://beagleboard.org/black (žiūrėta geg. 18, 2020).
44. „clock_gettime(2): clock/time functions - Linux man page“. https://linux.die.net/man/2/clock_gettime (žiūrėta geg. 18, 2020).
45. C. Moreno ir S. Fischmeister, „Accurate Measurement of Small Execution Times—Getting Around Measurement Errors“, *IEEE Embedded Syst. Lett.*, t. 9, nr. 1, p. 17–20, kovo 2017, doi: 10.1109/LES.2017.2654160.
46. „G.114 : One-way transmission time“. https://www.itu.int/rec/T-REC-G.114 (žiūrėta geg. 24, 2020).

Priedai

1 priedas. Gauti kalbos kokybės įvertinimo balai

originalus įrašas	CELP 4800 bps	Codec2 3200 bps	MELP 2400 bps	Codec2 2400 bps	MELPE 1200 bps	Apmokintas Codec2 1200 bps	Codec2 1200 bps
4	3	3	3	2	4	3	3
4	2	3	2	3	2	4	4
4	3	2	4	2	2	4	3
4	2	2	3	3	4	3	3
5	3	2	4	3	2	4	3
5	2	3	3	2	4	4	4
4	4	3	3	2	4	5	4
5	2	3	2	3	3	5	3
5	2	1	5	3	3	5	4
4	4	4	3	2	4	5	5
5	3	3	5	3	4	5	4
4	2	1	3	1	1	2	3
5	2	3	3	4	4	4	3
5	3	4	5	3	3	4	4
5	3	5	4	4	2	2	2
5	2	4	4	3	5	4	3
5	4	3	4	3	4	4	4
5	2	4	4	2	3	5	4
5	2	4	3	2	3	3	3
4	2	2	2	2	2	3	2
4	3	2	3	3	3	3	2
5	1	3	5	4	4	5	3
5	1	3	5	4	4	5	3
4	4	4	1	4	2	2	2
4	5	4	2	4	3	2	3
4	4	3	1	4	4	2	2
4	4	4	2	3	3	2	3
2	5	4	2	5	4	2	2
5	5	4	2	4	4	3	2
4	5	3	4	3	4	3	3
4	5	3	3	5	5	3	3
4	5	4	2	4	5	3	1
3	5	5	2	4	4	3	3
4	5	5	3	3	5	4	3
1	4	4	3	3	3	2	1

1 priedo lentelės tęsinys.

4	5	3	3	5	4	3	3
5	5	5	2	4	5	4	4
4	4	5	3	3	5	5	4
5	5	5	3	4	4	4	4
4	5	4	2	4	5	3	3
5	5	4	5	5	4	4	5
3	4	4	3	4	3	2	3
3	4	3	1	3	3	2	3
4	3	3	1	2	3	2	3
3	5	5	3	5	5	4	2
3	5	5	3	5	5	4	2
4	4	5	3	4	4	4	2
5	3	4	2	3	3	4	3
5	4	3	2	3	4	4	4
3	3	3	4	3	4	3	3

2 priedas. Gauti kalbos suprantamumo įvertinimo balai

originalus įrašas	CELP 4800 bps	Codec2 3200 bps	MELP 2400 bps	Codec2 2400 bps	MELPE 1200 bps	Apmokintas Codec2 1200 bps	Codec2 1200 bps
5	3	3	4	2	3	4	3
5	2	3	3	2	2	4	3
4	2	2	3	1	3	4	3
5	2	2	3	3	3	3	3
5	3	3	4	3	3	4	4
5	2	3	3	2	4	4	4
4	4	3	3	2	4	5	4
5	3	3	2	3	3	5	3
5	3	3	5	2	2	5	2
3	4	4	3	2	3	5	5
5	3	4	4	2	4	4	4
3	2	2	2	1	2	3	3
4	2	3	2	3	3	3	4
5	3	3	5	3	3	4	4
4	4	5	4	5	3	3	2
5	3	4	4	2	5	3	3
5	3	3	4	3	3	4	4
5	2	5	4	2	3	5	2
5	2	4	3	2	3	3	3
4	2	3	5	2	4	5	2
3	4	2	5	3	3	4	2
5	1	3	4	2	2	4	2
5	1	2	4	3	2	4	3
4	5	4	2	4	3	3	2
4	4	4	1	4	3	2	3
3	4	3	2	4	3	1	2
3	4	4	1	3	3	2	2
4	5	4	3	5	5	3	3
5	5	4	2	4	4	3	2
4	5	3	4	3	4	3	3
4	5	3	3	5	5	3	3
4	5	4	2	4	4	2	2
4	5	4	3	3	4	3	3
4	5	5	2	2	5	4	3
2	4	3	2	3	3	2	2

2 priedo lentelės tęsinys.

4	5	3	3	4	4	4	3
5	5	5	4	4	5	3	4
5	4	5	3	3	5	5	5
5	4	5	2	3	4	3	2
3	4	4	3	3	5	3	3
3	5	5	5	4	3	1	2
3	4	4	3	4	3	2	3
2	3	5	2	5	4	3	2
2	4	5	2	4	4	3	1
3	4	3	2	3	5	2	3
3	4	4	2	2	5	3	2
4	4	5	2	4	5	3	2
3	4	4	2	4	4	4	4
5	3	3	3	2	3	2	3
5	3	3	3	2	4	4	2