



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Arnas Navašinskas

AKUSTINIO TRIUKŠMO FILTRAVIMO REALIU LAIKU
ALGORITMO TYRIMAS

Baigiamasis magistro projektas

Vadovas

Doc. dr. Armantas Ostreika

KAUNAS, 2018

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

AKUSTINIO TRIUKŠMO FILTRAVIMO REALIU LAIKU
ALGORITMO TYRIMAS

Baigiamasis magistro projektas
Informatika (kodas 621I1003)

Vadovas

(parašas) Doc. dr. Armantas Ostreika
(data)

Recenzentas

(parašas) Doc. dr. Tomas Blažauskas
(data)

Projektą atliko

(parašas) Arnas Navašinskas
(data)

KAUNAS, 2018



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Arnas Navašinskas

(Studento vardas, pavardė)

Informatika (kodas 621I1003)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Akustinio triukšmo filtravimo realiu laiku algoritmo tyrimas“
AKADEMINIO SAŽININGUMO DEKLARACIJA

20 ____ m. _____ d.
Kaunas

Patvirtinu, kad mano, **Arno Navašinsko**, baigiamasis projektas tema „Akustinio triukšmo filtravimo realiu laiku algoritmo 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.

(vardą ir pavardę įrašyti ranka)

(parašas)

Arnas Navašinskas. Akustinio triukšmo filtravimo realiu laiku algoritmo tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Armantas Ostreika; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatika 09P

Reikšminiai žodžiai: *Garsų analizė.*

Kaunas, 2018. 97 p.

SANTRAUKA

Realiu laiku akustinį triukšmą filtruojančio algoritmo tyrimas. Tiriamos garso charakteristikos bei ieškoma reikšmingų koreliacijų su akustiniu triukšmu. Išrenkamos charakteristikos, kurios geba identifikuoti akustinį triukšmą arba reikšmingus garsus. Šios charakteristikos apjungiamos į garsų atpažinimo algoritmą gebantį identifikuoti akustinį triukšmą bei reikšmingus garsus. Pasiūlomas garso filtravimo algoritmas, gebantis realiu laiku išfiltruoti dalį akustinio triukšmo spektro, taip sumažinant nereikšmingos bei perteklinės garsinės informacijos kiekį kasdieninėje žmogaus aplinkoje.

Navašinskas Arnas. *Investigation of Real-Time Acoustic Noise Filtering Algorithm*: Master's thesis in Lithuanian language / supervisor assoc. prof. Ostreika Armantas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics 09P

Key words: *Sound analysis*

Kaunas, 2018. 97 p.

SUMMARY

Real-time acoustic noise filtering algorithm research. The sound characteristics are studied and a significant correlation with acoustic noise is sought. Characteristics that are capable of identifying acoustic noise or significant sound are selected. These characteristics are combined into a sound recognition algorithm that allows identifying acoustic noise and significant sound. An audio filtering algorithm is proposed that allows real-time filtering of part of the acoustic noise spectrum, thus reducing the amount of insignificant and excessive audio information in the daily human environment.

TURINYS

Paveikslėlių sąrašas	3
Lentelių sąrašas.....	6
Terminų ir santrumpų žodynas	7
Įvadas.....	8
Darbo tikslas	8
Uždaviniai	8
Mokslinis naujumas.....	9
Dokumento struktūra	9
Problemos apibrėžimas.....	10
1. Garso analizės ir filtravimo algoritmų apžvalga	12
1.1. Greitosios Furjė transformacijos (GFT) algoritmas.....	12
1.2. Realaus laiko balso aktyvumo aptikimo (BAA) algoritmai.....	13
1.3. Adaptyviųjų filtrų algoritmai	15
1.4. Kalmano filtras	16
1.5. Vinerio filtrai	18
1.6. Vinerio spektrų atimtis.....	19
1.7. Medianos filtras	19
1.8. Impulsinio triukšmo pašalinimas naudojant tiesinę prognozę	20
1.9. Paprastas iškraipymų filtravimas kalbos įrašė	21
1.10. Neuroniniu tinklu pagrįsti algoritmai	22
1.11. Pasikartojančio modelio išgavimo technikos algoritmas	23
1.12. Algoritmų tinkamumo įvertinimas.....	24
2. Akustinio triukšmo filtravimo algoritmo projektavimas	27
2.1. Akustinio triukšmo filtravimo algoritmo reikalavimai	27
2.2. Akustinio triukšmo filtravimo algoritmo modeliavimas	29

3.	Akustinio triukšmo filtravimo eksperimentinė dalis.....	35
3.1.	Akustinio triukšmo charakteristikų analizė	35
3.2.	Akustinio triukšmo charakteristikų kombinacijų tyrimas.....	59
3.3.	Akustinio triukšmo filtravimo algoritmo tyrimas.....	77
4.	Išvados ir rekomendacijos.....	93
6.	Šaltiniai	95
7.	Priedai	98
7.1.	1 Priedas. Kalendorinis darbų atlikimo grafikas.....	98
7.2.	2 Priedas. Skaičiavimų lentelės.	99
7.3.	3 Priedas. MATLAB kodas.	99

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Kosinuso bangų 10,20,30,40 bei 50 Hz DFT analizė	13
2 pav. Gaunamų rezultatų iš garso šaltinio (angl. Wave File) pavyzdys.....	14
3 pav. PAED (SED), PED (AED), NPD (ZCD), SFD (WFD) algoritmų bandymai, pirmieji stulpeliai (juodi) reiškia kompresiją (nereikalingų dažnių išmetimą), antrieji(balti) reiškia slankiojo kablelio operacijų kiekį per sekundę, tretieji (pilki) reiškia balsų kokybę [14].....	15
4 pav. MKV filtro schema	16
5 pav. MKV išfiltruoto signalo (mėlyna spalva) pavyzdys	16
6 pav. MKV filtro mokymosi kreivė	16
7 pav. Kalmano filtravimo pavyzdys.....	17
8 pav. Vinerio filtravimo pavyzdys. Žalias originalus signalas, mėlynas užterštas, raudonas išfiltruotas [19].....	18
9 pav. Medianos filtro signalo filtravimo pavyzdys	20
10 pav. Impulsinio triukšmo pašalinimo sistemos struktūra	21
11 pav. Triukšmo suradimas ir panaikinimas	21
12 pav. Viena-sluoksnio neuroninio tinklo modelis	22
13 pav. „Išpėjimo ir pavojaus ženklai – Urbanistinio triukšmo filtravimas geresniam vairuotojo darbui“ mokslinio straipsnio pasiūlytas sensorių išdėstymas	23
14 pav. Pasikartojančio modelio išgavimo technikos algoritmo etapai [24]	24
15 pav. Algoritmų tinkamumo vertinimų grafikas	25
16 pav. Testavimo sistemos modelis	28
17 pav. Garso filtravimo modelis realaus laiko panaudos atveju	30
18 pav. Garso filtravimo modelis testavimo tikslu	31
19 pav. Monotoniškų spektrų filtras (žr. 3.2.1.5 skyrių)	33
20 pav. Adaptyvaus limitu, maksimalios jėgos filtras (žr. 3.2.1.6 skyrių)	34
21 pav. Signalų glodinimo pavyzdys.....	36
22 pav. Slenkančio vidurkio filtro duomenų koreliacijos rezultatų grafikas.....	36
23 pav. Energijos charakteristikos išskyrimo palyginimas su duomenimis	37
24 pav. Pilnos juostos energijos charakteristikos palyginimas su duomenų amplitude	38
25 pav. Pilnos juostos energijos charakteristikos išėjimo duomenų koreliacijos rezultatų grafikas	38
26 pav. Nulio kirtimo dažnio charakteristikos išskyrimo pavyzdys.....	39

27 pav. Nulio kirtimo dažnio charakteristikos išeities duomenų koreliacijos rezultatų grafikas	39
28 pav. Spektrinės analizės palyginimas su akustiniu triukšmu užterštų duomenų amplitude ..	40
29 pav. Spektrinės analizės charakteristikos išeities duomenų koreliacijos rezultatų grafikas ..	40
30 pav. Dominuojančios dažnių komponentių palyginimas su duomenų amplitude.....	41
31 pav. Dominuojančių dažnių komponentių charakteristikos išeities duomenų koreliacijos rezultatų grafikas.....	42
32 pav. Spektro tiesiškumo charakteristikos išeities pavyzdys	43
33 pav. Spektro tiesiškumo charakteristikos išeities duomenų koreliacijos rezultatų grafikas ..	43
34 pav. Spektro geometrinio tiesiškumo išeitis palyginus su įrašo amplitude	44
35 pav. Spektro geometrinio tiesiškumo charakteristikos išeities duomenų koreliacijos rezultatų grafikas.....	44
36 pav. Spektro aritmetinio tiesiškumo išeitis palyginus su įrašo amplitude	45
37 pav. Spektro aritmetinio tiesiškumo charakteristikos išeities duomenų koreliacijos rezultatų grafikas.....	45
38 pav. Vidutinės kvadratinės vertės išeities palyginimas su garso įrašo amplitude	46
39 pav. Vidutinės kvadratinės vertės charakteristikos išeities duomenų koreliacijos rezultatų grafikas.....	46
40 pav. Spektro centroidės kitimas pravažiuojančių mašinų įrašė (aukštesni dažniai virsta žemesniais).....	47
41 pav. Spektro centroidės charakteristikos išeities duomenų koreliacijos rezultatų grafikas ...	48
42 pav. Spektro išsiskaidymo palyginimas su užteršto garso įrašo amplitude	49
43 pav. Spektro išsiskaidymo charakteristikos išeities duomenų koreliacijos rezultatų grafikas	49
44 pav. Charakteristikos išvedamų duomenų palyginimo pavyzdys.....	50
45 pav. Koreliacijos tarp kanalų naudojant kryžminę koreliaciją charakteristikos išeities duomenų koreliacijos rezultatų grafikas	50
46 pav. Koreliacijos tarp kanalų charakteristikos išeities duomenų koreliacijos rezultatų grafikas	51
47 pav. Mažos jėgos spektro kiekio charakteristikos išeities duomenų pavyzdys	52
48 pav. Mažos jėgos spektro kiekio charakteristikos išeities duomenų koreliacijos rezultatų grafikas.....	52
49 pav. Dažnių triukšmingumo charakteristikos rezultatų pavyzdys	53

50 pav. Dažnių triukšmingumo charakteristikos išėties duomenų koreliacijos rezultatų grafikas	53
51 pav. Spektrų atstumo charakteristikos išėties duomenų pavyzdys	54
52 pav. Spektrų atstumo charakteristikos išėties duomenų koreliacijos rezultatų grafikas	55
53 pav. charakteristikų koreliacijų palyginimas	58
54 pav. Vidutinės aukštos energijos dažnių charakteristikos išėties pavyzdys	64
55 pav. Vidutinės aukštos energijos dažnių charakteristikos koreliacijos bandymų grafikas	64
56 pav. LMVAPP koreliacija garso įrašams	65
57 pav. LMVAPP koreliacija garso įrašams pritaikius medianos filtrą	66
58 pav. Monotoniškų spektrų filtro išėties pavyzdys	68
59 pav. Adaptyvaus limito, maksimalios jėgos filtravimo pavyzdys	69
60 pav. Pasvertų filtravimo koreliacijų grafikas.....	70
61 pav. Spektrų atimties filtro išėties pavyzdys	71
62 pav. Spektrų atimties filtro koreliacijos grafikas	72
63 pav. S. F. Boll [36] balso atpažinimo algoritmo pavyzdys.....	72
64 pav. MMSE LSA [38] algoritmo išvesties pavyzdys	74
65 pav. Kombinuojamų charakteristikų koreliacijų palyginimas	76
66 pav. Pirmosios kombinacijos bandymo pavyzdys	78
67 pav. Pirmosios kombinacijos bandymų grafikas	78
68 pav. Antrosios kombinacijos bandymo pavyzdys	79
69 pav. Antrosios kombinacijos bandymų grafikas.....	80
70 pav. Psichoakustikos diktuojamas žmogaus jautrumas dažnių spektrui, esant tokiems šių garsų lygiams mums garsai atrodytų vienodo garso. [32]	80
71 pav. Algoritmo bandymai be psichoakustinio modelio	81
72 pav. Algoritmo bandymai pritaikius psichoakustinį modelį.....	82
73 pav. Geriausios nustatymų kombinacijos rezultatų bandymo grafikas	84
74 pav. analoginio triukšmo duomenų rodiklio išėties duomenų pavyzdys.....	88
75 pav. Patobulinto S. F. Boll [36] spektrų atimties algoritmo bandymo pavyzdys	89
76 pav. Patobulinto S. F. Boll [36] spektrų atimties algoritmo bandymų grafikas	89
77 pav. Patobulinto MMSE LSA [38] algoritmo bandymo pavyzdys	90
78 pav. Patobulinto MMSE LSA [38] algoritmo bandymų grafikas.....	91

LENTELIŲ SĄRAŠAS

1 lentelė. Algoritmų tinkamumo vertinimo duomenys.	25
2 lentelė. Visų charakteristikų bendrų koreliacijos nuokrypių palyginimas	57
3 lentelė. Koreliacijų nuokrypių palyginimas	75
4 lentelė. Algoritmo nustatymų bandymai (ištrauka iš priedo nr.2)	83
5 lentelė. Priedas: Magistrinio darbo kalendorinis grafikas.....	98

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Garsų analizė	(<i>Angl. sound analysis</i>) Informacijos bei reikšmės išgavimas iš garsų.
Algoritmas	(<i>Angl. algorithm</i>) Algoritmo samprata gali būti pernelyg abstrakti. Šiame dokumente tiriama algoritmai, kurie gali būti išverčiami į kompiuterio programinį kodą, tačiau yra abstraktesni už modelius.
SNR	(<i>Angl. SNR – signal to noise ratio</i>) Signalų ir triukšmo santykis išreiškiamas decibelais.
BAA	(<i>Angl. VAC - voice activity detection</i>) Balso aktyvumo aptikimas – algoritmas skirtas aptikti balso aktyvumą realiu laiku analizuojant garsų duomenis.
DFT	(<i>Angl. DFT - discrete Fourier transform</i>) Diskrečioji Furjė transformacija.
GFT	(<i>Angl. FFT – fast Fourier Transform</i>) greitoji Furjė Transformacija – algoritmų aibė optimizuojanti DFT.
O(n)	(<i>Angl. time complexity</i>) laiko kompleksškumas aprašomas „Big-O“ notacija, išreiškiantis algoritmo veikimo laiką [7].
IGFT	(<i>Angl. IFFT – inverse fast Fourier Transform</i>) atvirkštinė greitoji Furjė Transformacija. Tai GFT duomenų transformacija į originalų šaltinį, iš kurio buvo gauti GFT duomenys.
BAA	(<i>Angl. VAC – voice activity detection</i>) balso aktyvumo aptikimo algoritmas.
PED	(<i>Angl. AED – adaptive energy detector</i>) prisitaikančio energijos detektoriaus algoritmas.
PAED	(<i>Angl. SED simple energy detector</i>) paprastojo energijos detektoriaus algoritmas.
NPD	(<i>Angl. ZCD - zero crossing detector</i>) nulio perlipimo detektoriaus algoritmas.
SFD	(<i>Angl. WFD - weak fricatives detector</i>) silpno frikatyvo detektoriaus algoritmas.
AVD	(<i>Angl. AAV - autocorrelation vector variance</i>) autokoreliacijos vektoriaus dispersijos algoritmas.
AF	(<i>Angl. AF – Adaptive filter</i>) adaptyvisus filtras.
MKV	(<i>Angl. LMS - Least Mean Square</i>) mažiausio vidutinio kvadrato algoritmas.
SVD	(<i>Angl. FIR - finite impulse response</i>) slenkančio vidurkio filtras.
STS	(<i>Angl. SNR – signal to noise ratio</i>) signalų triukšmui santykis.
ITP	(<i>Angl. INR – impulse noise removal</i>) impulsinio triukšmo pašalinimo algoritmas.
PMIT	(<i>Angl. REPET - repeating pattern extraction technique</i>) pasikartojančio modelio išgavimo technikos algoritmas.
GMKA	(<i>Angl. DLMSS - deep learning for monaural speech separation</i>) giliojo mokymosi kalbos atskyrimui algoritmas.
NKD	(<i>Angl. ZCR – zero crossing rate</i>) Nulio kirtimo dažnis.

ĮVADAS

Kiekvieną dieną žmonės susiduria su akustinio triukšmo (nepageidaujama, sukeliančių diskomfortą, nemalonių žmogaus ausiai garsų) problema. Tai yra vis didėjanti problema, neatsiejama nuo vis sparčiau vykstančios urbanizacijos.

Žmogaus organizmo reakcija į triukšmą yra neigiama - iššaukia irzlumą, pablogėja sveikata, sumažėja darbingumas, sutrinka miegas [1]. Akustinis triukšmas tėra perteklinė informacija, kurios išvengus būtų pagerintas darbingumas, sveikata bei sumažinamas nuovargis. ES 2017 metų tyrimas suskaičiavo virš 100 milijonų ES piliečių kenčiančių didesnę nei leistiną miesto triukšmą bei akustinio triukšmo padariniams priskyrė 70 tūkstančių susirgimų ir 16 tūkstančių mirčių per metus [2].

Triukšmo bei jo šaltinių mažinimo metodai yra nepakankamai veiksnūs ir turi akivaizdžių trūkumų. Pasyvus triukšmo slopinimas (ausinės, ausų kištukai) žmogų izoliuoja nuo prasmingų garsų, didina riziką nukentėti padidinto pavojaus zonose, neišvengiamai limituoja komunikacijos galimybes. Trūksta aktyvių technologinių sprendimų akustinio triukšmo problemai spręsti.

Augant kompiuterių spartai kuriami vis tobulėsniai algoritmai duomenų analizei, triukšmo aptikimui bei pašalinimui. Daug dėmesio skiriama ryšių technologijoms, žmogaus balso filtravimui, tačiau daugelį žmonių kiekvieną dieną varginančio akustinio triukšmo problema apeinama. Trūksta aktyvių sprendimų, tyrimų, skirtų bendrai akustinio triukšmo problemai spręsti.

Šiame darbe tiriamas algoritmas, gebantis realiu laiku išfiltruoti akustinį triukšmą, bei perteklinius garsus. Tyrimo metu ištiriamos garso charakteristikos, pasiūlomos ir ištiriamos charakteristikų kombinacijos. Pasiūlomas akustinio triukšmo filtravimo algoritmas, gebantis sumažinti akustinio triukšmo ir perteklinių garsų kiekį. Pasiūlomos pastarojo algoritmo kombinacijos su egzistuojančiais sprendimais. Algoritmai testuojami su miesto bei darbo aplinkos įrašais.

Darbo tikslas

Pasiūlyti algoritmą, gebantį realiu laiku identifikuoti akustinį triukšmą ir perteklinius garsus, bei juos išfiltruoti paliekant reikšmingiausią informaciją garse.

Uždaviniai

1. Ištirti populiariausius garso analizės ir filtravimo algoritmus;
2. Ištirti garso charakteristikų išskyrimo algoritmus;
3. Ištirti charakteristikų apjungimo modelius;
4. Pasiūlyti akustinio triukšmo filtravimo algoritmą;
5. Ištirti galimas kombinacijas su esamais filtravimo algoritmais.

Mokslinis naujumas

1. Tiriamos garso charakteristikų kombinacijos;
2. Siūlomas akustinio triukšmo filtravimo algoritmas.

Dokumento struktūra

Šiame dokumente pateikiama egzistuojančių sprendimų bei algoritmų analizė, apžvelgiami tinkamiausi algoritmai, įvertinamas jų tinkamumas akustinio triukšmo filtravimui. Pateikiamas algoritmo projektavimo planas. Tiriamos garsų charakteristikos koreliuojant su akustiniu triukšmu. Reikšmingiausios charakteristikos kombinuojamos, siūlomas akustinio triukšmo identifikavimo bei filtravimo algoritmas. Ištiriamas akustinio triukšmo filtravimo algoritmo veiksmingumas. Siūlomos algoritmo kombinacijos su egzistuojančiais sprendimais.

PROBLEMOS APIBRĖŽIMAS

Sprendžiama problema - akustinio triukšmo mažinimas žmogaus aplinkoje, aplinkos garsų filtravimo pagalba. Akustinis triukšmas gali būti apibūdinamas kaip įvairaus pobūdžio garsai, kurie yra nemalonūs, garsūs, trukdantys išgirsti norimus garsus ir pan.

Žmogaus ausies mechanizmas bei psichologiniai procesai yra sukurti prisitaikyti prie triukšmo, ausis geba sumažinti juntamą garsumą, sumažinti jautrumą tam tikram dažnių spektrui. Išnagrinėjus žmogaus fiziologijos savigynos mechanizmus galima teigti jog daugiausia žalos padaro plataus spektro ir žemų bei aukštų dažnių triukšmas. Žemi dažniai perneša daugiausia energijos ir žmonės jiems nesa jautrūs (fizinio pažeidimo grėsmė), o aukštiems dažniams esant per jautrūs ir pastarųjų apstu žmonių aplinkoje (neigiamo psichologinio poveikio grėsmė)[42][43][44][45].

Akivaizdu, jog filtravimas turi vykti realiu laiku. Norima išfiltruoti kuo daugiau parazitinio triukšmo spektro, paliekant pakankamai (nesumažinant suprantamumo) reikšmingos garso informacijos, užtikrinant saugumą bei gebėjimą komunikuoti.

Triukšmo filtravimo algoritmas turėtų būti lankstus ir pritaikomas dažniausiai akustinio triukšmo užterštų aplinkų garsų filtravimui. Pateikiami pavyzdžiai su kuriais EU piliečiai susiduria dažnai ir tai sudaro didelę jų akustinės taršos dalį [30]:

Problema 1: Mieste šlapiu keliu važiuojančios mašinos sukelia didžiulį kiekį akustinio triukšmo. Esant dideliame triukšmo kiekiui sunku suprasti ar iš nugaros atvažiuoja mašina nes informacijos kiekis yra per didelis, nors ją girdėti ir išskirti iš kitų yra įmanoma. Sukeliama grėsmė sveikatai dėl triukšmo bei limituojamas bendras budrumas. Tai paveikia visus eismo dalyvius nuo pėsčiųjų iki vairuotojų.

Sprendimas 1: Algoritmas turėtų gebėti sumažinti triukšmo kiekį išskiriant reikšmingiausia informaciją (identifikuojant pravažiuojančias mašinas) ir išfiltruojant pasikartojančią, akustinio triukšmo informaciją.

Problema 2: Didžiausia grėsmė bei žala sukeliamą gamyklų, dirbtuvių darbininkams, kurie dirba su įrankiais nuolatos išskiriančiais plataus spektro 100dB+ akustinį triukšmą. Darbuotojai nedėvi ausų apsaugos nes ji nemaloni psichologiškai (nuslopina visus garsus), trukdo girdėti aplinką (sukelia saugumo problemų) ir pan.[42][43][44][45].

Sprendimas 3: Analogiškas sprendimas 1 sprendimui, tik nuolatinio pasikartojančio triukšmo informacija galėtų taip pat būti išfiltruojama (pvz.: šalimais vyksta pjovimo darbai, po kelių sekundžių nuo pjovimo pradžios šis triukšmas patampa pasenus ir nebereikalinga informacija jeigu jis nekintantis ir stacionarus).

Problema 3: Ofiso aplinkoje akustinį triukšmą sudaro oro cirkuliacijos, vėdinimo, šildymo sistemos bei bendradarbių triukšmas. Triukšmingos aplinkos pasekmės: sumažėjęs darbingumas, padidėjęs nuovargis, prastesnė psichologinė būseną, grėsmė sveikatai [42][43][44][45].

Sprendimas 3: Algoritmas turėtų gebėti sumažinti arba išvis panaikinti akustinį triukšmą iki kol priėjęs bendradarbis bando pasikalbėti.

Šie pavyzdžiai apibrėžia testavimo duomenis - akustinio triukšmo bei reikšmingos informacijos (taip pat vadinamos – originaliais duomenimis, kai kombinuojama su triukšmu) derinius.

1. GARSO ANALIZĖS IR FILTRAVIMO ALGORITMŲ APŽVALGA

Siekiant pasiūlyti veiksmingą akustinio triukšmo filtravimo algoritmą, ištirti egzistuojantys garso analizės ir filtravimo algoritmai.

Skaitmeniniai garsų analizės ir filtravimo būdai yra plačiai tyrinėjami. Pastarieji skiriasi savo analizės kompleksiskumu nuo statistinio vidurkio pritaikymo iki tikrų garsų egzistavimo aptikimo ir kompresijos. Pvz.: žmogaus balso aptikimo algoritmas naudojamas mob. tinklų operatorių (BAA) [4], adaptyvieji filtrai, neuroniniai tinklai.

Algoritmų yra daugiau nei įmanoma ištirti. Parinkti kriterijai, kurie leidžia algoritmus surūšiuoti pagal tinkamumą šio tyrimo paskirčiai, tam jog galima būtų prioretizuoti skiriamą laiką pagal algoritmo tinkamumą.

Parinkti kriterijai – algoritmo paprastumas – A, algoritmo tinkamumas (pagal duomenų apdorojimo paskirtį) – B, algoritmo greitaveika – C, algoritmo populiarumas, tyrimų informacijos kiekis, realizavimo pavyzdžių kiekis – D. Kiekvienas algoritmas įvertinamas pagal kriterijus. Šie vertinimai yra subjektyvūs, tačiau procesui apibrėžti sudaryta vertinimo balo formulė (1.1).

Ištyrus egzistuojančius sprendimus, galima įvertinti jų tinkamumą problemos sprendimui bei atitinkamai paskirstyti tyrimo laiką, net jeigu nei vieno algoritmo dalis nebūtų panaudojama. Tai būtina norint veiksmingai palyginti skirtingus algoritmus, nustatymus ir pan. Algoritmai lyginami pagal:

- 1.1. Garsų atpažinimo galimybes F (0-100%) kur 100% yra tiksliausiai atpažįstami garsai;
- 1.2. Garsų filtravimo galimybes G (0-100%) kur 100% yra tiksliausiai išfiltruojami garsai;
- 1.3. Greitaveiką H (0-100%) kur 100% yra greičiausias algoritmas ir 100% atitinka greičiausią algoritmą, kaip atsvaros tašką nuo kurio skaičiuojama;

Įvertinus algoritmų parametrus, bus išvedamas balas E pagal formulę (1.1):

$$E = (F/1.5 + G + H/1.5); \quad (1.1)$$

Toliau – apžvelgiami egzistuojantys algoritmai, šaltiniai, ištiriamas jų tinkamumas.

1.1. Greitosios Furjė transformacijos (GFT) algoritmas

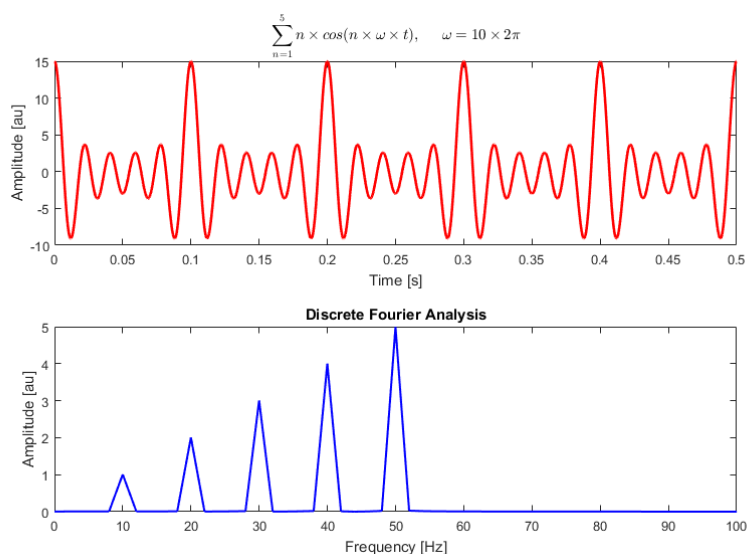
Greitosios Furjė transformacijos (GFT)(*angl. fast Fourier transform(FFT)*) yra vienas populiariausių ir svarbiausių algoritmų, tinkamų šiam tyrimui bei naudojamų kituose algoritmuose šio tyrimo metu.

GFT yra vadinami algoritmai gebantys išspręsti diskrečiosios Furjė transformacijos (DFT) uždavinį bent $O(N \log N)$ laiku [6], kitaip tariant tai yra algoritmai optimizuojantys DFT, kuris yra $O(n^2)$.

Furjė transformacijos yra viena svarbiausių koncepcijų informaciniuose moksluose. Tai nereguliarių signalų pateikimo kaip grynų dažnių kombinacija būdas. Panaudos atvejų pvz.: įtampos svyravimai laide, garso bangos, elektromagnetinis triukšmas ir pan. Jos visuotinai naudojamos apdorojant signalus, tačiau taip pat gali būti panaudotos suspaudžiant vaizdus ir garsų duomenis, sprendžiant diferencialines lygtis, tiriant akcijų rinką ir kt. [6].

Skaitmeninis signalas yra tiesiog skaičių sekos – diskretūs analoginio signalo rinkiniai. GFT transformuoja skaitmeninį signalą jį išreiškiant kaip svertinę dažnių sumą [6].

Pagrindinė GFT algoritmo paskirtis – konvertuoti signalą iš jo originalaus domeno (dažniausia laiko arba erdvės) į dažnių domeno reprezentaciją ir atvirkščiai (IGFT) [8] (žr. 1 pav.).



1 pav. Kosinuso bangų 10,20,30,40 bei 50 Hz DFT analizė [6]

GFT algoritmo aktualumas šiam tyrimui:

- Turint domeno dažnių reprezentaciją gaunami papildomi duomenys, pvz.: galima išskirti triukšmingiausius dažnius;
- Algoritmas geba transformuoti duomenis į pirminį domeną. Galima panaikinti tam tikrų dažnių garsus arba sumažinti jų amplitudę. Daugelis algoritmų naudoja GFT garsų analizei ir IGFT (atvirkštinį GFT procesą) filtravimui.

1.2. Realus laiko balso aktyvumo aptikimo (BAA) algoritmai

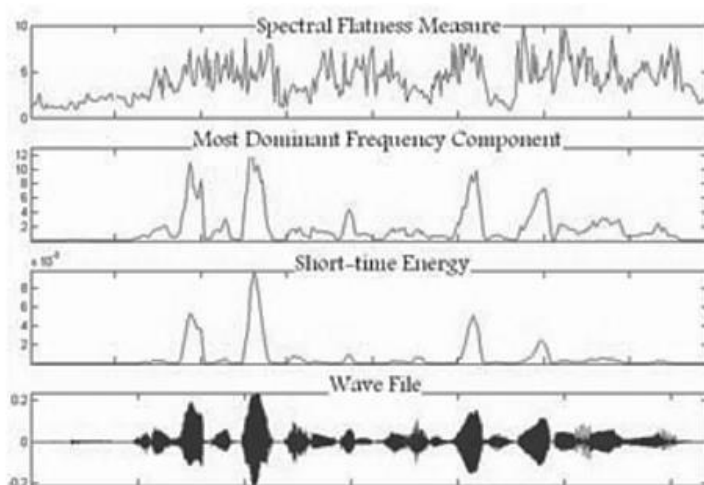
Egzistuoja daug realaus laiko balso aktyvumo aptikimo (BAA) algoritmo variacijų, įvairių implementacijų. BAA algoritmai realizuoja šiam tyrimui aktualius garsų charakteristikų analizavimo būdus, kurie leidžia identifikuoti svarbiausius garsus esant dideliame SNR (*angl. signal to noise ratio*), be to sprendžiamos triukšmo eliminavimo problemos garso analizavimo eigoje.

Apžvelgiama implementacija pateikta šaltinyje „A Simple But Efficient Real-Time Voice Activity Detection; M. H. Moattar and M. M. Homayounpour.“ [4]. Ši BAA implementacija apjungia spektro plokštumos (SP) (*angl. spectral flatness*) [12] ir trumpalaikės energijos (*angl. short-term energy*) [12] algoritmus. Šaltinyje aprašomi tyrimai atlikti naudojant įvairius triukšmo lygius, bei nustatymus.

Algoritmo veikimo metu apjungiamos trys garso charakteristikos gautos iš atskirų algoritmų sukombinuotų į vieną: spektro plokštumos (SP) (*angl. spectral flatness*) [12], trumpalaikės energijos (*angl. short-term energy*) [12], dominuojančio dažnio radimo (*angl. dominant frequency extraction*) [13] (žr. 2 pav.). Akivaizdu jog visos šios garso charakteristikos yra aktualios šiam tyrimui ir gali būti kombinuojamos garsų atpažinimui.

BAA naudojamas:

- Telekomunikacijose (VoIP);
- Balso įrašytuvuose;
- Atsakymo mašinose.



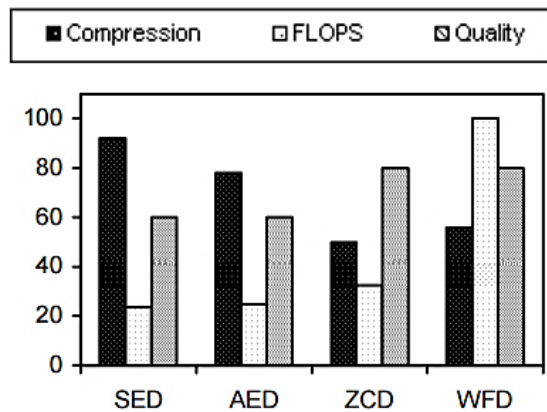
2 pav. Gaunamų rezultatų iš garso šaltinio (*angl. Wave File*) pavyzdys [12]

Analogiškas, tačiau skirtingai realizuotas BAA algoritmas aprašomas šaltinyje „VAD Techniques for Real-Time Speech Transmission on the Internet, ABHIJEET SANGWAN, CHIRANTH M.C. H.S.JAMADAGNI, RAHUL SAH, R. VENKATESHA PRASAD, VISHAL GAURAV“ [13]. Šioje algoritmo realizacijoje kombinuojamos šios charakteristikos:

- Paprastasis energijos detektorius (PAED) (*angl. simple energy detector (SED)*)
- Modifikuotas prisitaikantis energijos detektoriaus (PED) (*angl. adaptive energy detector (AED)*) algoritmas;
- Nulio perlipimo detektoriaus (NPD) (*angl. zero crossing detector (ZCD)*) algoritmas;

- Silpno frikatyvo detektoriaus (SFD) (*angl. weak fricatives detector (WFD)*) algoritmas naudojant autokoreliacijos vektoriaus dispersijos (AVD) (*angl. autocorrelation vector variance(AVV)*) algoritmą.

Atlikti šio algoritmo tyrimai, aprašantys visų keturių charakteristikų bandymus, jie pavaizduoti (žr. 3 pav.).



3 pav. PAED (SED), PED (AED), NPD (ZCD), SFD (WFD) algoritmų bandymai, pirmieji stulpeliai (juodi) reiškia kompresiją (nereikalingų dažnių išmetimą), antrieji (balti) reiškia slankiojo kabelio operacijų kiekį per sekundę, tretieji (pilki) reiškia balsų kokybę [14]

Daugelis šių algoritmų, kombinuojami su paprastu triukšmo filtravimu atsižvelgiant tik į dažnių spektrą. BAA algoritmų analizuojamos garso charakteristikos gali būti panaudotos universalus algoritmo kūrimui.

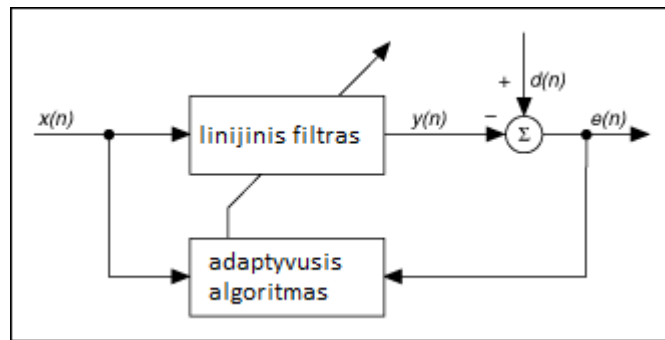
1.3. Adaptyviųjų filtrų algoritmai

Adaptyvieji signalų apdorojimo metodai vis labiau populiarėja, ypač ryšių srityje. Adaptyviųjų filtrų (PFA) (*angl. Adaptive filter*) algoritmų idėja yra paprasta – signalas filtruojamas (transformuojamas) pagal tam tikrą algoritmą, kurio transformavimo funkcija yra kontroliuojama pagal kintamąjį. Kintamasis nuolat keičiamas pagal kitą algoritmą, priklausomai nuo duomenų pakitimų [10].

Adaptyvūs mažiausios kvadratinės paklaidos filtrai yra naudojami apdoroti nestacionariems signalams, bei ten, kur reikalingi maži apdorojimo vėlinimai, pvz.: daugiakanaliniame triukšmo slopinime, radarų signalų apdorojime, kanalo išlyginime, aido naikinime, kalbos kodavime [18].

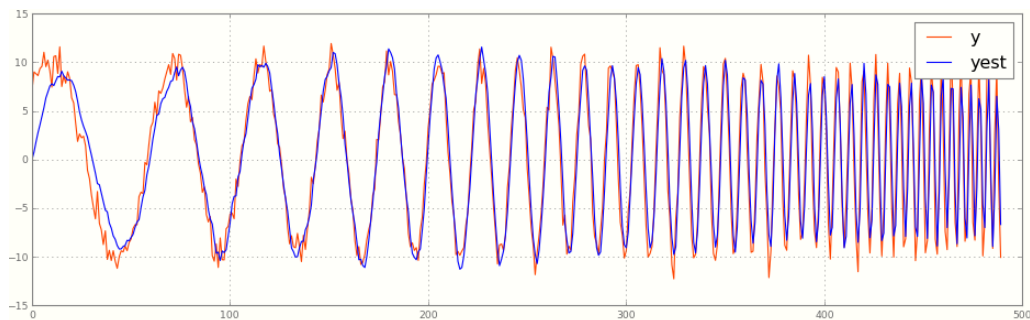
Mažiausio vidutinio kvadrato algoritmas (MKV) (*angl. LMS - Least Mean Square*) yra prisitaikančių filtrų klasės algoritmas, kuris yra stochastinis (dirbantis su atsitiktiniais duomenimis). Tik stochastiški prisitaikantys filtrai gali susidoroti su realaus laiko realaus pasaulio duomenimis, jie nesimoko iš praeities esamu laiku, mokosi tik iš esamos duomenų būsenos (nors ir turi nuosavą esamą būseną paveldėtą iš netolimos praeities) [15].

Prisitaikantis filtras yra skaičiavimo įtaisas, kuris iteratyviai modeliuoja sąryšį tarp įeinančio ir išeinančio signalų filtre (žr. 4 pav.).

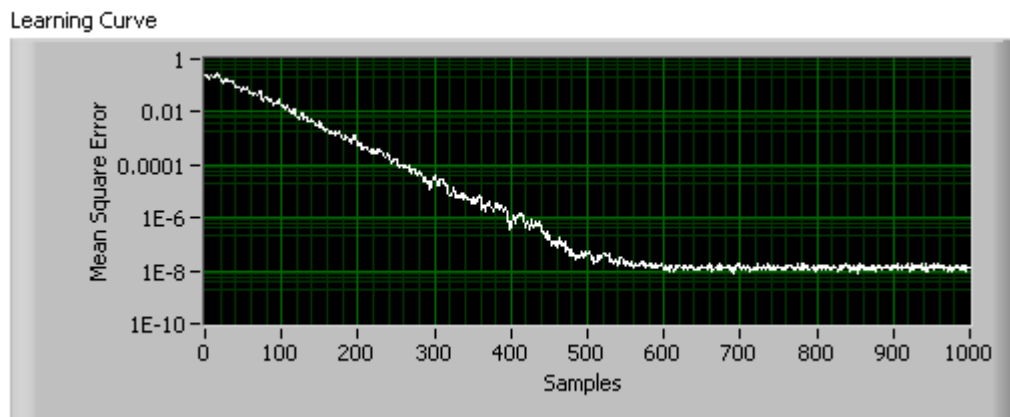


4 pav. MKV filtro schema [15]

Filtro realizavimas, galimybės, mokymosi kreivės bei greitimeika apžvelgiama šaltinyje [16] (žr. 5 pav., 6 pav.). Filtras nėra tinkamas universaliam garsų identifikavimui, tačiau gali būti pritaikomas pasikartojantiems, pastoviems garsams filtruoti (grėžimas, oro vėdinimo sistema ir pan.).



5 pav. MKV išfiltruoto signalo (mėlyna spalva) pavyzdys [15]

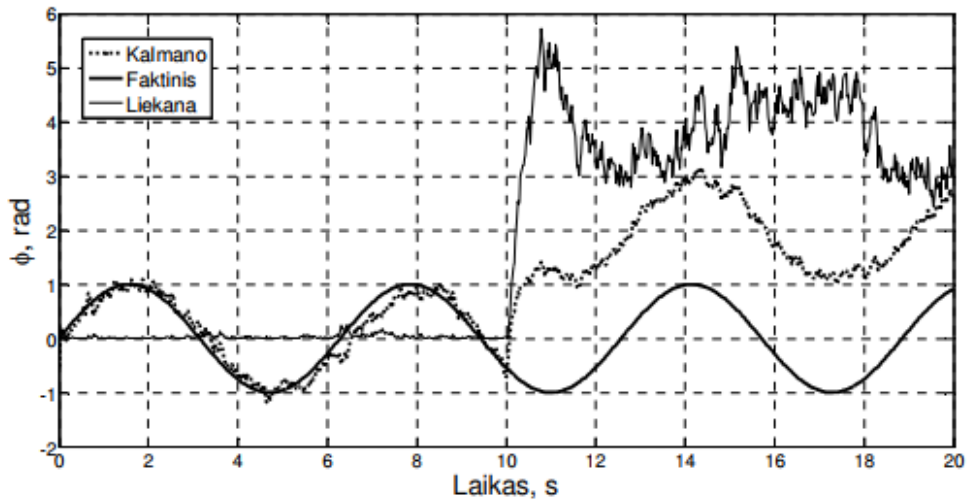


6 pav. MKV filtro mokymosi kreivė [15]

1.4. Kalmano filtras

Kalmano filtras yra adaptyviųjų filtrų dalis, tačiau jis dažnai tiriamas, tobulinamas ir naudojamas bene pusėje klausos aparatų gamintojų algoritmų [33]. Dėl šio filtro tinkamumo ir pritaikymo verta jį išanalizuoti atskirai.

Kalmano filtras apsimoko adaptyviai ir išlaiko savo poziciją iki sekančio mokymų atnaujinimo (žr. 7 pav.).



7 pav. Kalmano filtravimo pavyzdys [33]

Kalmano filtro darbą kiekviename žingsnyje galima suskirstyti į dvi fazes: prognozavimo ir korekcijos [35].

Prognozuojama būsena nustatoma kaip ankstesnės būsenų kaitą nusakančios matricos F_k ir ankstesnės pakoreguotos prognozuojamos būsenos sandauga $X_{k-1|k-1}$, pridėdant reakciją į įėjimus nusakančios matricos B_{k-1} ir įėjimo u_{k-1} sandaugą:

$$X_{k|k-1} = F_k X_{k-1|k-1} + B_{k-1} u_{k-1} \quad (1.2)$$

Prognozuojama būsenos kovariacija nustatoma iš ankstesnės pakoreguotos prognozuojamos kovariacijos, ankstesnės būsenų kaitą nusakančios matricos ir triukšmo kovariacijos Q_{k-1} :

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1} \quad (1.3)$$

Korekcija y_k - matavimo paklaida randama kaip skirtumas tarp faktinės matavimo reikšmės z_k ir numatytos matavimo reikšmės (matricos H_k , nusakančios matavimo modelį ir prognozuojamos būsenos $x_{k|k-1}$ sandaugos):

$$y_k = z_k - H_k x_{k|k-1} \quad (1.4)$$

Paklaidos kovariacija gaunama iš matavimo modelį nusakančios matricos G_k , prognozuojamos būsenos kovariacijos H_k^T ir triukšmo kovariacijos R_k :

$$S_k = G_k P_{k|k-1} H_k^T + R_k \quad (1.5)$$

Optimalus stiprinimo koeficientas nustatomas iš prognozuojamos būsenos, matavimo modelio ir paklaidos kovariacijos:

$$K_k = P_{k|k-1} + H_k^T + R_k \quad (1.6)$$

Pakoreguota prognozuojama būseną apskaičiuojama pridant prie prognozuojamos būsenos optimaliai sustiprintą paklaidą:

$$X_{k|k} = X_{k|k-1} + K_k Y_k \quad (1.7)$$

Pakoreguota prognozuojama kovariacija gaunama iš atitinkamų matmenų vienetinės matricos, optimalaus stiprinimo, matavimo modelio ir prognozuojamos kovariacijos:

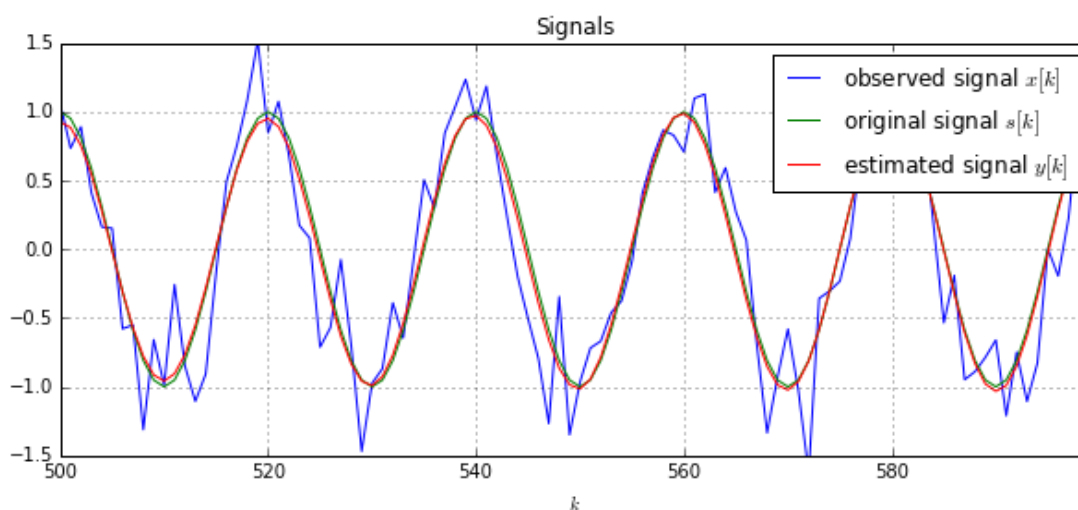
$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (1.8)$$

1.5. Vinerio filtrai

Vinerio teorija, suformuluota Norberto Vinerio (Norbert Wiener), yra nuo duomenų priklausančių minimalios kvadratinės paklaidos (MKP) filtrų pagrindas. Šie filtrai plačiai naudojami tiesinėje prognozėje, signalo kodavime, aido (*anlg. echo*) naikinime, signalo atstatyme, radaruose ir pan.

Vinerio filtro paskirtis – sugeneruoti norimą arba užsibrėžtą atsitiktinį procesą, pasinaudojant linijiniu nepriklausomu nuo laiko filtravimu. Vinerio filtrai paremti statistiniu filtravimu (MKP) [20].

Vinerio filtro koeficientai yra paskaičiuojami tokie, jog minimizuotų vidutinį kvadratinį atstumą tarp norimo signalo ir signalo filtro išėjimo. Vinerio teorijoje signalas yra laikomas stacionariu procesu. Tačiau, jei filtro koeficientai yra periodiškai perskaičiuojami kiekvienam atskaitų blokui, tada filtras adaptuojasi prie vidutinių signalo charakteristikų bloke ir filtras tampa blokais adaptyvus. Toks filtras naudojamas signalams, kurie laikomi stacionariais bloko trukmės laiko intervale [17] (žr. 8 pav.).



8 pav. Vinerio filtravimo pavyzdys. Žalias originalus signalas, mėlynas užterštas, raudonas išfiltruotas [19]

Tipiniai deterministiniai filtrai yra sukurti dirbti tam tikrose dažnių ribose, tačiau Vinerio filtrų dizainas, parametrų dėka, leidžia vartotojui nustatyti kokio signalo tikimasi.

Vinerio filtrai yra paplitę ir garsų filtravime, todėl pasirinkta juos ištirti. Dažniausia Vinerio filtrai naudojami triukšmo filtravime ir yra apjungiami su kitais algoritmais.

Vienas paprasčiausių Vinerio filtrų be jokių papildomų modifikacijų yra slenkančio vidurkio filtras (SVD) (*angl. FIR - finite impulse response*) filtras. Slenkančio vidurkio filtrai, tai lengviausiai suprantami ir realizuojami SVD filtrai. Nežiūrint jų paprastumo jie puikiai tinka signalo formos glodinimui, atsitiktinio triukšmo lygio mažinimui. Be to pasižymi trumpu reakcijos į žingsnio signalą laiku. Tačiau šie filtrai turi rimtų trūkumų dažnio srityje [21]. SVD filtrų tyrimai garsų filtravime aprašomi šaltinyje „FIR Filter For Audio Practitioners; Hadi Sumoro, Xian Yu“ [22].

1.6. Vinerio spektrų atimtis

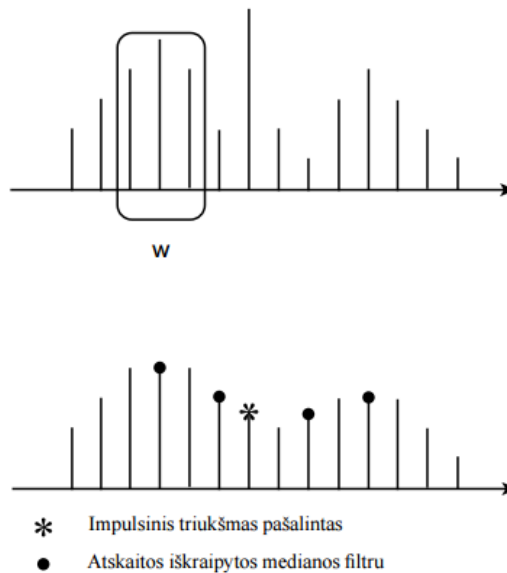
Spektrų atimtis (*angl. spectral subtraction*) – tai triukšmu iškraipyto signalo amplitudės spektro atstatymo metodas, kai vidutinis triukšmo spektras yra atimamas iš iškraipyto signalo spektro. Pradžioje yra randamas triukšmo spektras. Vėliau jis atnaujinamas iš signalo periodų, kuriuose nėra signalo, o yra tik triukšmas. Tarp atnaujinimo periodų triukšmas laikomas stacionariu ar lėtai kintamu procesu, kurio spektras praktiškai nekinta. Atstatant signalą laike, gautas amplitudžių ir fazių spektras yra transformuojamas į laiko sritį pasinaudojant atvirkštine Furjė transformacija (IGFT).

Spektrų atimties metodas nėra labai imlus skaičiavimui. Tačiau, dėl atsitiktinio triukšmo kitimo, naudojant spektrų atimtį galima gauti neigiamas amplitudės spektro dedamąsias. Amplitudės spektro dedamosios yra neneigiami dydžiai, todėl neigiamos dedamosios turi būti atvaizduotos į neneigiamas. Toks netiesinis taisymo procesas iškraipo atstatyto signalo pasiskirstymą. Šie iškraipymai tuo labiau juntami, kuo mažesnis signalo ir triukšmo santykis (STS) [23, 5p.].

Spektrų atimties algoritmas turi savų trūkumų, tačiau yra daugelis skirtingų būdų pašalinti atsiradusius signalo iškraipymus. Algoritmas gali būti pritaikomas signalo triukšmingumo mažinimui.

1.7. Medianos filtras

Medianos filtras yra klasikinis impulsinio triukšmo [29] šalinimo būdas. Medianos filtruose, tam tikro ilgio langas slenka signalu ir vidurinė atskaita lange yra pakeičiama visų atskaitų esančių lange mediana (vidurkiu) [23] (žr. 9 pav.).



9 pav. Medianos filtro signalo filtravimo pavyzdys [23]

Aibės skaičių mediana yra netiesinė aibės statistika su naudinga savybe, - ji nejautri atskaitoms su nepagrįstai didelėmis reikšmėmis. Ši medianos filtrų savybė yra plačiai naudojama vaizdų apdorojime.

Medianos filtras negali pašalinti realaus impulsinio triukšmo, kuris paprastai yra ilgesnis negu viena atskaita. Be to, šie filtrai iškraipo ir tas signalo atskaitas, kurios nėra pažeistos impulsiniu triukšmu. Dėl to šis filtras nėra tiesiogiai tinkamas akustiniam triukšmui filtruoti, tačiau kombinuojant su kitomis filtravimo technikomis gali būti naudingas.

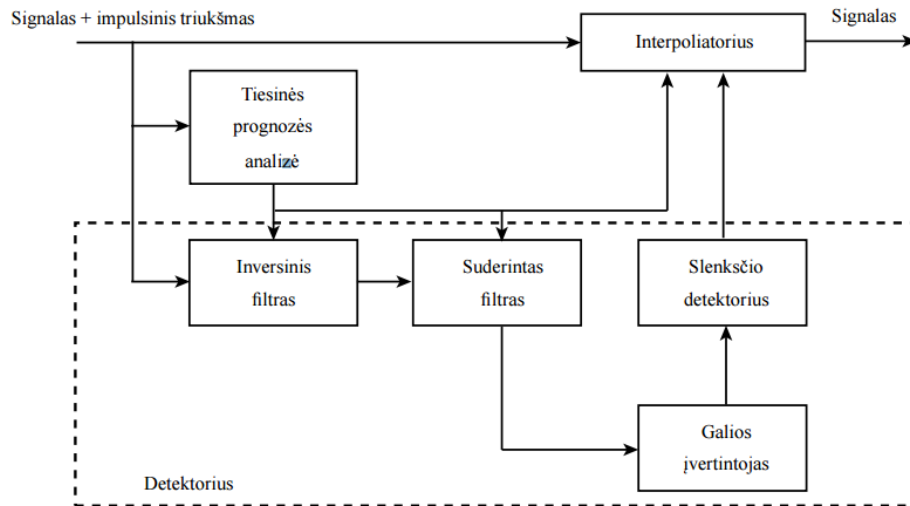
1.8. Impulsinio triukšmo pašalinimas naudojant tiesinę prognozę

Impulsinio triukšmo pašalinimo algoritmas (ITP). Paprastai impulsinis triukšmas paveikia nedidelę dalį visų signalo atskaitų, o didžioji atskaitų dalis lieka nepakitusi. Taigi, naudojant tiesinę prognozę išvengiama nereikalingo signalo apdorojimo, iškraipymo surandant tik atskirus triukšmo impulsus ir juos pašalinant.

Impulsinio triukšmo aptikimo bei pašalinimo metodai gali būti modifikuojami į garso charakteristikų išskyrimo algoritmus.

Įprastai detektorius nustato kiekvieno triukšmo impulso vietą, o interpoliatorius pakeičia iškraipytas signalo atskaitas naujomis, kurios randamos pasinaudojant esančiomis prieš or po trukdžio atskaitomis (žr. 10 pav.).

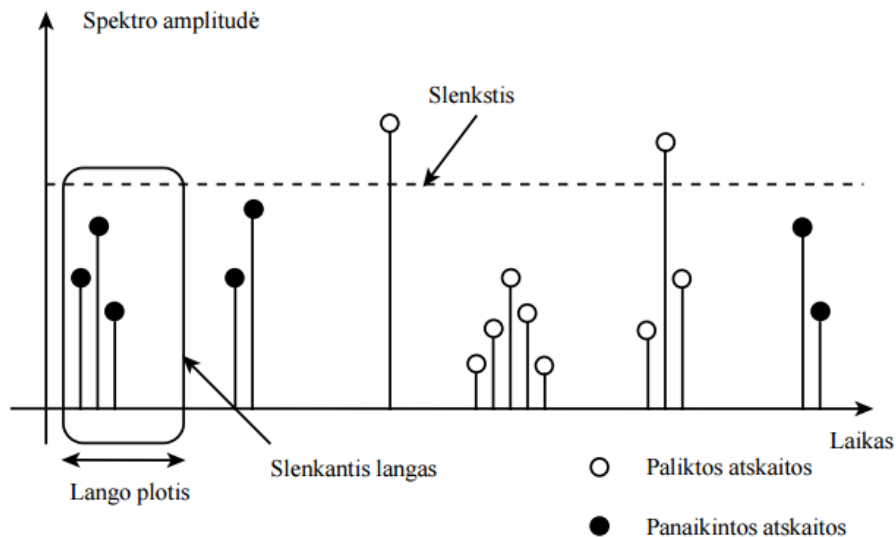
Šis filtras yra pažangesnis už Medianos filtravimą, todėl verta ištirti tiesinės prognozės svarbą garso charakteristikų išskyrimui.



10 pav. Impulsinio triukšmo pašalinimo sistemos struktūra

1.9. Paprastas iškraipymų filtravimas kalbos įrašė

Kalbos signalas yra sudarytas iš nestacionarių akustinių sekų, kurios atsiranda, egzistuoja tam tikrą laiką ir išnyksta, o po to vėl atsiranda, bet turi kitą intensyvumą ir spektro sudėtį. Dažniausia yra žymus skirtumas tarp kalbos signalo charakteristikų ir triukšmo. Šie skirtumai gali būti panaudoti išskirti muzikinį triukšmą iš kalbos signalo. Pagrindinis šio triukšmo bruožas – trumpalaikiai ir siaurajuosčiai mažos amplitudės signalo protrūkiai. Jų trukmė paprastai neviršija 20 ms, tuo tarpu tikras signalas yra didesnės trukmės (žr. 11 pav.). Pavaizduotas tokio triukšmo identifikavimo metodas [23].



11 pav. Triukšmo suradimas ir panaikinimas [23]

Kiekviename GFT dažnių diapazone ieškoma trumpalaikių dažnio dedamųjų. Jei dažnio komponentės trukmė mažesnė už lango plotį ir amplitudė mažesnė už slenkstį, tada ji laikoma triukšmu ir yra pašalinama [23].

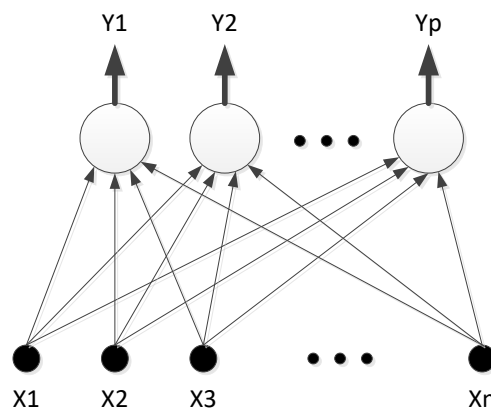
Algoritmas yra ganėtinai primitivus, tačiau kombinuojant su sudėtingesniais algoritmais gebėjimas filtruoti trumpalaikius triukšmo šuolius gali būti naudingas (pvz.: išfiltruoti kitų algoritmų sukurtą arba praleistą triukšmą, ar/ ir padėti realaus triukšmo filtravime kur sudėtingesni algoritmai praleidžia trumpalaikius triukšmo šuolius).

1.10. Neuroniniu tinklu pagrįsti algoritmai

Dirbtinis neuroninis tinklas (*angl. neural network*) – tarpusavyje sujungtų dirbtinių neuronų grupė. Ši technologija mėgdžioja žmogaus galvos smegenų darbą – tiksliau neuronų veikimą. Esama įvairių rūšių neuroninių tinklų. Vieni iš jų apmokomi su mokytoju (žinant reikiamas tinklo išėjimų reikšmes), kiti – be mokytojo (mokosi iš pavyzdinių duomenų) [5].

Neuroninio tinklo veikimo pagrindas yra neuronas. Neuronas yra skaičiavimo mazgas, kuris gali turėti keletą įėjimo reikšmių. Kiekviena neurono įėjimo jungtis turi savo svorį, dar vadinamą perdavimo koeficientu. Kiekvienas neuronas turi savo slenkstinę reikšmę ir neurono sužadinimo reikšmę. Sužadinimo reikšmė apskaičiuojama, iš įėjimo signalų svorių sumos atimant slenkstinę reikšmę. Neurono išėjimo reikšmė apskaičiuojama pagal neurono perdavimo koeficientą. Perdavimo koeficientas yra funkcija, kurios argumentas yra sužadinimo reikšmė. Bendra įdėja yra ganėtinai paprasta, neuroniniai tinklai pagal sudėtingumą prasideda nuo paprasčiausio – viena-sluoksnio neuroninio tinklo (žr. 12 pav.).

Tinkamo tipo, apmokytas neuroninis tinklas gebėtų iš akustinio triukšmo išskirti žmogaus balsus. Kas leistų išspręsti specifines akustinio triukšmo situacijas.



12 pav. Viena-sluoksnio neuroninio tinklo modelis

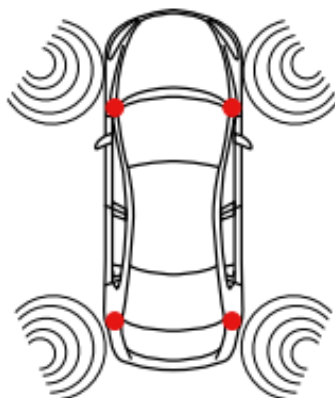
Didžiausias neuroninių tinklų algoritmų trūkumas – jų greیتaveika ir apmokymo laikas. Gali būti, jog tinkamam neuroninio tinklo algoritmui veikti realiu laiku nepakaks įrenginio resursų. Šaltiniuose yra mažai informacijos, tyrimų susijusių su greیتaveika. Tačiau matomas potencialas, todėl verta apsvarstyti neuroninių tinklų galimybes.

Gilusis mokymasis kalbos atskyrimui (GMKA) (*angl. DLMSS - deep learning for monaural speech separation*) algoritmas aprašytas šaltinyje [25] sėkmingai įgyvendina kalbos (dainavimo) atskyrimą nuo muzikos, bei triukšmo filtravimą įrašė, kur kalba žmogus [26]. Šaltiniuose nėra pateikta algoritmo greitaveikos duomenų, be to neužsiminta apie pritaikymus realaus laiko duomenų apdorojimui.

Algoritmas yra sudėtingas, jo realizacija paremta daugeliu bibliotekų, bei kitais algoritmais skirtais garsų apdorojimui, algoritmo pavyzdžiai šaltinyje [27].

Šaltinyje „A Neural Network for Real-Time Signal Processing; Donald B. Malkoff“ [28] aprašoma galimybė pritaikyti neuroninį tinklą realaus laiko signalų apdorojimui. Aprašomos galimybės neuroninį tinklą panaudoti triukšmo filtravimui.

Mokslinis straipsnis pavadinimu „Įspėjimo ir pavojaus ženklai – Urbanistinio triukšmo filtravimas geresniam vairuotojo darbui“ [31] aprašo neuroninio tinklo panaudojimą kelio triukšmui perteikti automobilio vairuotojui bet kurį triukšmą atpažįstant (prilyginant atitinkamai garsui duomenų bazėje) ir perdurbant į malonesnę bei informatyvesnę garsą. Darbe siūloma išdėstyti 4 sensorius aplink mašiną tam jog tiksliai atkurti erdvinį garsą dviejose dimensijose (žr. 13 pav.).



13 pav. „Įspėjimo ir pavojaus ženklai – Urbanistinio triukšmo filtravimas geresniam vairuotojo darbui“ mokslinio straipsnio pasiūlytas sensorių išdėstymas [31]

Šie sprendimai gali būti tinkami akustinio triukšmo problemai spręsti.

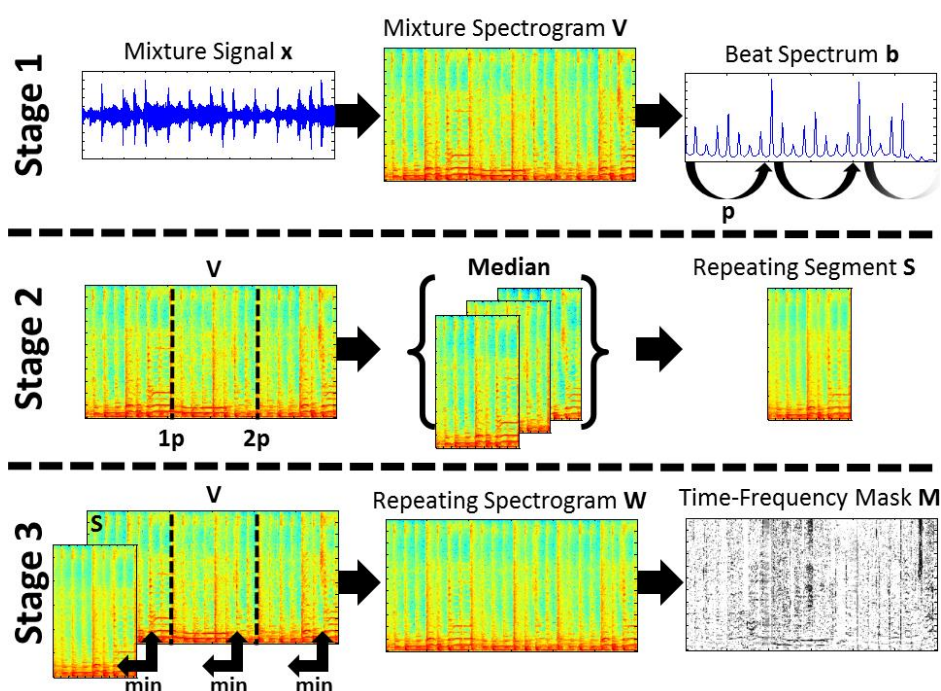
1.11. Pasikartojančio modelio išgavimo technikos algoritmas

Pasikartojančio modelio išgavimo technikos (PMIT) (*angl. REPET – repeating pattern extraction technique*) algoritmas yra skirtas muzikos garso duomenų, kurioje groja grupė bei dainuoja vokalas dekompozicijai į muziką ir vokalą atskirai, turint omenyje jog muzika yra pasikartojanti. Akivaizdu, jog šis algoritmas tinka nedaugeliui dainų, tačiau gali būti idealus būdas pasikartojančiam akustiniam triukšmui filtruoti.

Algoritmas yra suskirstytas į 3 stadijas. Šaltinyje [24] stadijos detalios aprašytos (žr. 14 pav.) ir pateikiami detalūs tyrimai, bandymai ir keletas algoritmo versijų. Algoritmo pagrindinis filtras yra muzikos ritmo išgavimo technika, kuria remiasi sekančios stadijos. Šią techniką pritaikius išgauti pasikartojantį triukšmą algoritmas galėtų filtruoti triukšmą.

Didžiausias algoritmo trūkumas – jis reikalauja daug skaičiavimo resursų ir gali būti sunkiai pritaikomas realaus laiko skaičiavimams. Akustinio triukšmo problemos atžvilgiu algoritmas nėra pritaikytas modeliuoti kintamą akustinį triukšmą (sunkiai aptinka ir nekintamo režius).

Neskaitant trūkumų, šis algoritmas geba ganėtinai tiksliai modeliuoti statinį (nekintamą) akustinį triukšmą ir gali būti panaudojamas problemos sprendimui.



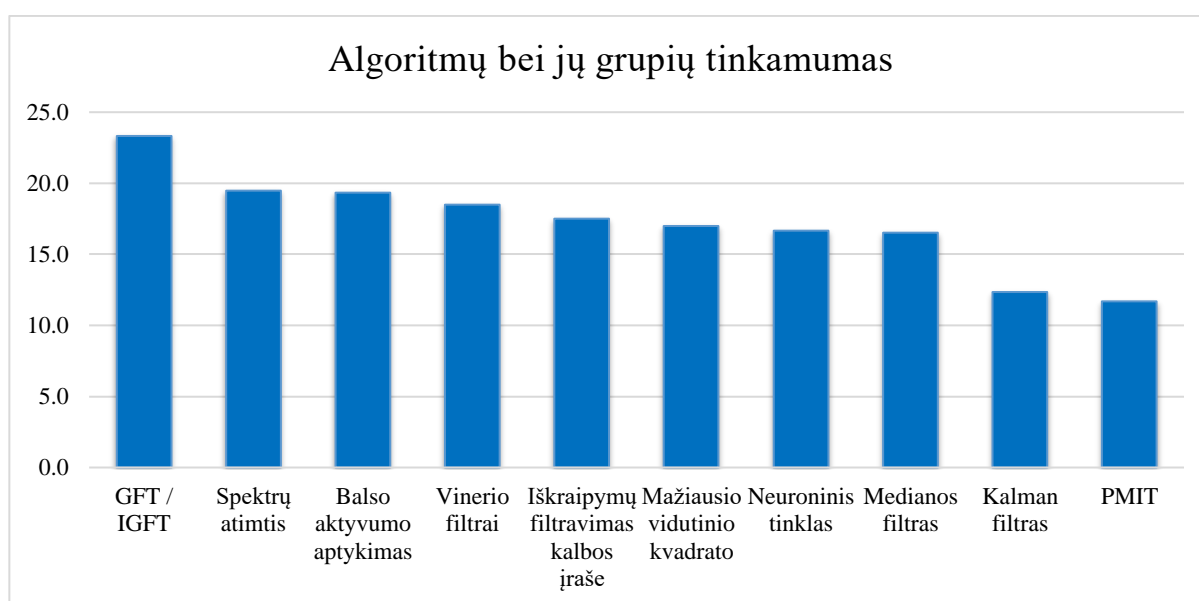
14 pav. Pasikartojančio modelio išgavimo technikos algoritmo etapai [24]

1.12. Algoritmų tinkamumo įvertinimas

Visi prieš tai aprašyti skyriai (algoritmai arba jų grupės) įvertinti (subjektyviai) pagal formulę (1.1) ir surūšiuoti pagal balą. Rezultatai pateikiami 1-moje lentelėje bei grafiškai atvaizduojami (žr. 15 pav.).

1 lentelė. Algoritmų tinkamumo vertinimo duomenys.

Algoritmo (-ų grupės) pavadinimas	Algoritmo paprastumas (10 – paprastas, 1 – sudėtingas)	Algoritmo tinkamumas (10 – tinkamas, 1 – netinkamas)	Algoritmo greیتaveika (10 – greitas, 0 – lėtas)	Tyrimų informacijos kiekis (10 – daug, 1 – mažai)	Balas
GFT / IGFT	10	10	10	10	23.3
Spektrų atimtis	8	8	9	9	19.5
Balso aktyvumo aptykimas	8	9	8	7	19.3
Vinerio filtrai	9	7	8	9	18.5
Iškraipymų filtravimas kalbos įrašė	10	5	9	9	17.5
Mažiausio vidutinio kvadrato	7	6	9	9	17.0
Neuroninis tinklas	3	9	7	8	16.7
Medianos filtras	10	4	9	9	16.5
Kalman filtras	1	7	5	7	12.3
PMIT	7	8	5	7	16.3



15 pav. Algoritmų tinkamumo vertinimų grafikas

Sprendimų tyrimo metu surasta, jog GFT / IGFT algoritmai yra tinkamiausi, o Kalman filtras - netinkamiausias algoritmas akustinio triukšmo filtravimo atžvilgiu.

Nerasta algoritmų skirtų bendram (visose aplinkose) sveikų žmonių akustinio triukšmo mažinimui ir filtravimui, daugelis algoritmų specializuojasi į žmogaus kalbos apdorojimą.

Tinkami (garso filtravimo) algoritmai yra naudojami klausos prietaisų gamintojų. Dažniausia jų realizuojami algoritmai yra komercinė paslaptis, tačiau egzistuoja bendrų apžvalgų kas naudojama ir pasiūlymų kaip juos patobulinti [33].

Klausos prietaisų gamintojų naudojami algoritmai:

- BAA – labai platus spektras balso aptikimo ar/ir spektro išskyrimo algoritmu;
- Spektrinės atimties algoritmai;
- Vinerio filtrai;
- Kalmano filtrai.

Pastebėta, jog klausos prietaisų gamintojai orientuojasi tik į žmogaus kalbėsenos aptikimą, dinaminio diapazono sumažinimą (žmonės turintys problemų su klausa yra jautresni dideliame garsumo dinaminiam diapazonui). Daugelis naudojamų filtravimo technikų yra tiesiogiai priklausomos nuo balso aptikimo algoritmo. Triukšmo pavyzdinis spektras atnaujinamas tik neaptikus žmogaus kalbos.

Akustinio triukšmo bendrojo atvejo sprendimui klausos prietaisų naudojami triukšmo filtravimo modeliai yra netinkami, nes jie priklauso nuo balso aptikimo, arba yra pritaikomi tik balso išgavimo optimizavimui. Taip pat naudojamas erdvinis atskyrimas (turint 2 ir daugiau mikrofonų vienoje ausyje) dažnai yra kryptinis (girdima tik tai kas priekyje užbrėžtoje vietoje) ir yra netinkamas šiai problemai spręsti. Sprendžiama platesnio spektro akustinio triukšmo problema, tačiau šie garso filtravimo algoritmai gali būti įkomponuoti.

Nepaisant klausos prietaisų naudojamų algoritmų trūkumų, daugeliu atveju triukšmingam signalui išfiltruoti (aprašę balso aptikimo algoritmus (BAA)) jie naudoja Vinerio, Kalmano, spektrų atimties filtrus. Tikėtina, jog pastarieji filtrai tikėtų bendrojo akustinio triukšmo filtravimo tikslais.

Ištyrus šaltinius nustatyta, jog bendro akustinio triukšmo spektro filtravimo problema retai nagrinėjama, apeinama gilinantis į siauresnio spektro problemas. Yra daug adaptyvių ir paprastesnių algoritmų gebančių išvalyti signalą nuo triukšmo, tačiau nėra apibrėžtas bendras akustinio triukšmo algoritmas gebantis aprašyti (klasifikuoti) tąjį akustinį triukšmą.

Adaptyviųjų filtrų bei adaptyviosios spektrinės atimties ir pan. apmokymui reikia atskirti reikšmingus garsus nuo triukšmingų (akustinio triukšmo). Toliau bus tiriamas būtent šis algoritmas gebantis identifikuoti akustinį triukšmą arba reikšmingus garsus skirtinguose dažnių diapazonuose. Toks algoritmas arba jo dalys gali būti pritaikomi daugeliui garso filtravimo algoritmų, juos apmokant, tobulinant ir pan.

2. AKUSTINIO TRIUKŠMO FILTRAVIMO ALGORITMO PROJEKTAVIMAS

Šiame skyriuje aprašomas akustinio triukšmo filtravimo algoritmo projektavimo (tyrimo) planas. Apibrėžiami reikalavimai, testavimas, pateikiamas abstraktus algoritmo modelis. Pilnas algoritmas pasiūlomas tyrimo metu (žr. 3.3 skyrių).

2.1. Akustinio triukšmo filtravimo algoritmo reikalavimai

Siekama sukurti universalų, patikimą akustinio triukšmo filtravimo algoritmą. Algoritmas gali būti taikomas padidinto pavojaus aplinkoje, dėl to jis neturi degraduoti svarbiausios garso informacijos, turi leisti ją perteikti greitai ir efektyviai, tačiau sumažinant akustinio triukšmo kiekį.

2.1.1. Reikalavimai algoritmui

- 2.1.1.1. Algoritmas turi gebėti apdoroti duomenis realiu laiku (neturi atsižvelgti į turimus ateities duomenis testavimo įrašuose);
- 2.1.1.2. Algoritmas turi gebėti atskirti akustinį triukšmą nuo reikšmingų garsų;
- 2.1.1.3. Algoritmo konfigūracija turi būti lanksti, t.y. norint patobulinti atpažinimą arba pritaikyti specifinei panaudai nekeičiant charakteristikų analizių ar kodo;
- 2.1.1.4. Algoritmas turi gebėti išfiltruoti bent 40% akustinio triukšmo kai tai įmanoma (kai reikalinga informacija neperdengia akustinio triukšmo);
- 2.1.1.5. Algoritmas turi išlaikyti bent 80% reikiamos informacijos, žmogus turi greitai ir tiksliai suprasti kas tai per garsai;
- 2.1.1.6. Algoritmas turi sumažinti plataus dažnių spektro informacijos dažnių diapazoną kai tai įmanoma (pvz.: miesto triukšmas, šlapiu keliu pravažiuojanti mašina skleidžia plataus diapazono garsus);
- 2.1.1.7. Akivaizdu, jog algoritmas turės savo būseną aprašomą perduodamų parametrų, kurią įtakos netolima praeities analizė. Ši būsena atspindi filtruojamą aplinką ir netūrėtų apimti daugiau nei 30 praeities sekundžių, tam jog algoritmas būtų lankstus filtruojant dinaminį akustinį triukšmą;
- 2.1.1.8. Algoritmas turi sumažinti plataus dažnių spektro triukšmą. Tokio triukšmo pamatavimas gali būti labai paprastas – 1024 kvadratinių dažnių reikšmių (GFT) vidurkis.

2.1.2. Algoritmo testavimo planas

Algoritmų testavimams naudojami 14 įrašų:

- 2.1.2.1. Miesto aplinkos įrašai, užteršti 4 miesto aplinkos triukšmo lygiais;

2.1.2.2. Gamyklos aplinkos įrašai, užteršti 4 gamyklos aplinkos triukšmo lygiais;

2.1.2.3. Ofiso aplinkos įrašai, užteršti 6 ofiso aplinkos triukšmo lygiais.

Visi garso įrašai yra kombinuojami iš tikrų aplinkos įrašų (reikšmingo garso, bei triukšmo atskirų įrašų), tam jog turėti tikslią, statinę originalaus ir triukšmo įrašų informaciją atskirai. Algoritmo veiksmingumas pamatuojamas koreliacijos koeficientu. Koreliacijos koeficientai turi atitikti reikalavimus (žr. 16 pav., 2.1.1 skyrių).

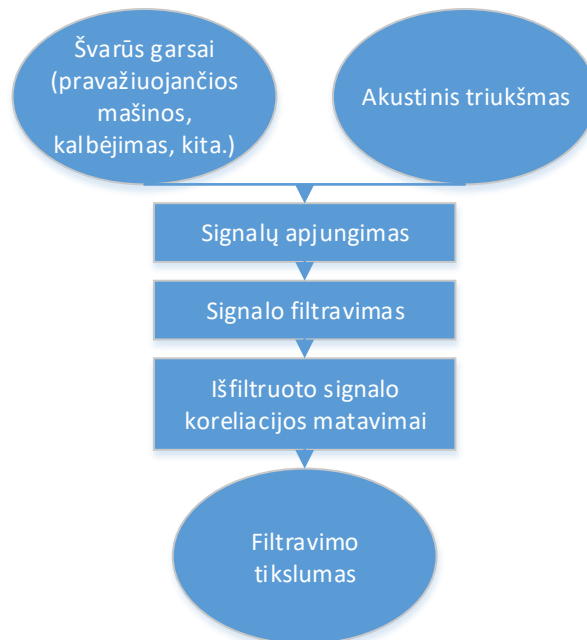
Tiksliai išmatuojamas algoritmo tikslumas žinant kurios garso įrašų dalys turi būti filtruojamos ir atpažintos. Signalų koreliacijos matavimams gali būti naudojamos:

- Pirsono koreliacijos formulė. Pirsono koreliacijos koeficientas įvertina tiesinio ryšio stiprumą ir yra naudojamas kai stebimų atsitiktinių dydžių reikšmės yra išmatuotos intervalų arba santykių skalėje. Naudojama Pirsono formulė (2.1):

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}} \quad (2.1)$$

- Spirmeno koreliacijos formulė. Spirmeno (Spearman) ranginės koreliacijos koeficientas apskaičiuojamas pagal formulę (2.2).

$$r_s = \frac{\sum_{i=1}^n (R_{x_i} - (n+1)/2)(R_{y_i} - (n+1)/2)}{(\sum_{i=1}^n (R_{x_i} - (n+1)/2)^2 \sum_{i=1}^n (R_{y_i} - (n+1)/2)^2)^{1/2}} \quad (2.2)$$



16 pav. Testavimo sistemos modelis

Kartu su koreliacijos garsų atkūrimo testavimais, galima pamatuoti bendros garso jėgos pernešimo kiekį (kiek jis sumažėjo) kaip papildomą algoritmo veiksmingumo matą. Matuojama

sudedant atskirų dažnių amplitudes ir palyginant įeinančio bei išeinančio įrašo, normalizavus pagal jų didžiausią amplitudę analoginėje išraiškoje.

Papildomai, kokybės matavimams, naudojamas STI (*angl. Speech transmission Index*) indekso modelis gebantis įvertinti kalbos atkūrimo kokybę. STI modelis naudojamas transmisijų, kodavimo programų balso persiuntimo įvertinimui ir yra apibrėžtas amerikos standartų institucijos[39]. Darbo eigoje pastebėta, jog STI modelis reikšmingus pasikeitimus įvertina labai mažu kokybės skirtumu, todėl papildomai pasitelktas įrašo kokybės matavimo modelis STOI (*angl. Short time objective intelligibility measure*). Šis modelis skirtas įvertinti kalbos suprantamumą (*angl. Intelligibility*), lyginant švarų ir filtruotą įrašus[40][41].

2.2. Akustinio triukšmo filtravimo algoritmo modeliavimas

Pasirinkta naudoti MATLAB įrankį kodo realizacijai. MATLAB įrankis pateikia daugelį populiariausių filtrų realizacijų, jas lengva pritaikyti, palyginti.

Šiame skyriuje gali būti pateikta MATLAB įrankio kodo pavyzdžių, kaip algoritmo dalių paaiškinimo įrankis. MATLAB kodas išskirtas atskiroje pastraipoje. Kodo dalys naudojamos matematinėse formulėse. MATLAB kodas pateiktas 2 priedų skyriuje.

2.2.1. Algoritmo modeliavimas

Garsai turi būti apdorojami realiu laiku, dėl to algoritmas turi dirbti su apibrėžto laiko informacijos kadrais. Taip pat filtruojami garsai neturi vėlinti, dėl to tikslinga algoritmo architektūrą pritaikyti kuo greitesniam garso apdorojimui.

Siūlomo algoritmo veikimo principas susideda iš šių etapų:

2.2.1.1. Garso įvestis;

2.2.1.2. Pradinis garso signalo filtravimas (būtinai tik sistemiškai užterštam įrašui (įrašymo įrenginio tarša));

2.2.1.3. Garso signalo kadravimas;

2.2.1.4. Kadruoto garso signalas filtruojamas etapais:

2.1.1.4.1. Etapų charakteristikų išgavimas;

2.1.1.4.2. Garso filtravimas po kiekvieno etapo.

2.2.1.5. Informacijos išsaugojimas, nustatymų atnaujinimas ir perdavimas sekančiam kadrai;

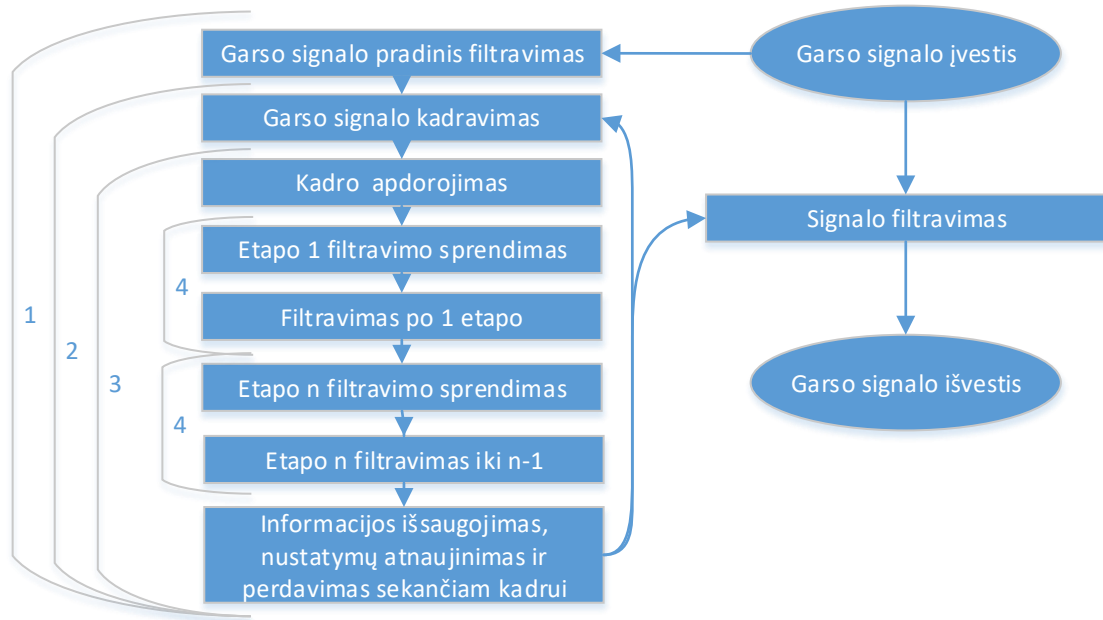
2.2.1.6. Kartojama 2.2.3 kol yra duomenų;

2.2.1.7. Realus laiko filtravimo atveju garsas filtruojamas nuosekliai pagal 2.2.5 etapo sprendimus, testavimo atveju etapų filtravimo rezultatai gali būti tiesiog apjungiami;

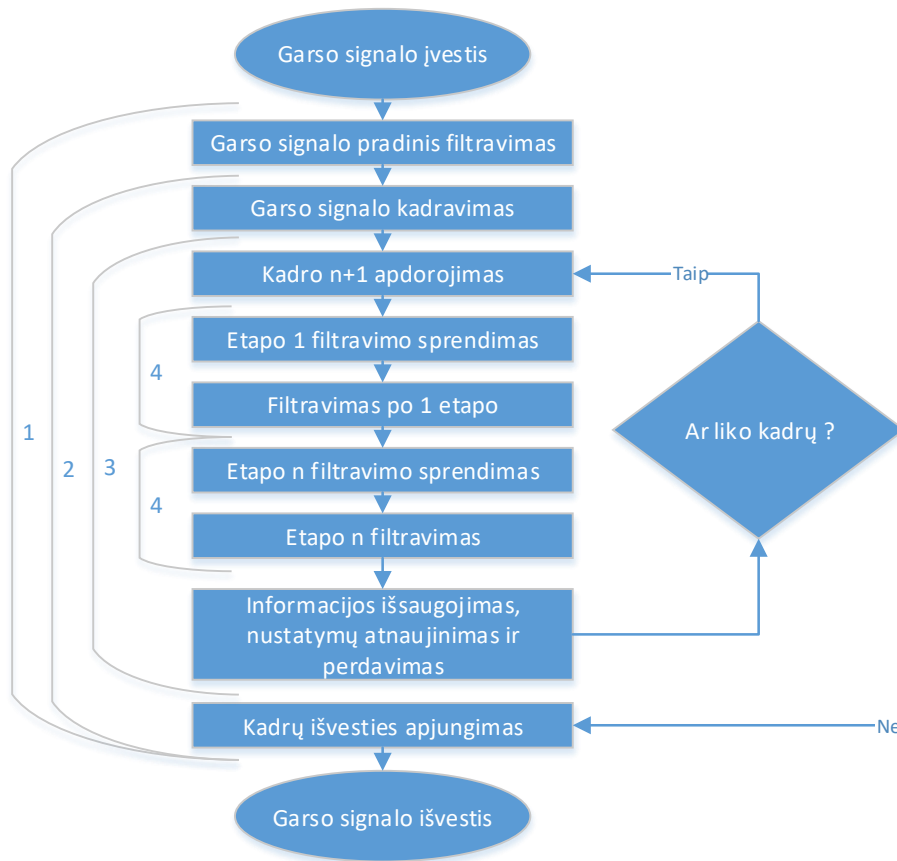
2.2.1.8. Garso išvestis.

Realaus laiko apdorojimo atveju garsas turi būti filtruojamas nepertrūkstamai, norint sumažinti garso vėlinimą filtravimo sprendimai daromi atskirai nuo realaus filtravimo (žr. 17 pav.). Tokiu atveju vėluoja tik filtravimo sprendimai, o ne realaus laiko garsas, tačiau tai neturi neigiamos įtakos sistemos funkcionalumui.

Testavimo tikslu naudojami statiniai švaraus bei triukšmo garsų įrašai ir jie apjungiami. Filtravimo metu įrašai suskaidomi į kadrus, o kadru etapų sprendimai apjungiami (žr. 18 pav.)



17 pav. Garso filtravimo modelis realaus laiko panaudos atveju



18 pav. Garso filtravimo modelis testavimo tikslu

Algoritmo etapus galima apjungti į modelius:

1. Signalų filtravimo modelis
2. Signalų kadravimo modelis;
3. Kadro apdorojimo modelis;
4. Kadro etapo modelis;

Algoritmas turi savo būseną, kurią aprašo perduodami parametrai. Parametrus įtakoja netolima praeities analizė. Ši būseną atspindi filtruojamą aplinką ir netūrėtų apimti daugiau nei 30 praeities sekundžių, tam jog algoritmas būtų kuo lankstesnis vis kintančiai žmogaus aplinkai.

2.2.2. Signalų filtravimo modelis

Signalas paruošiamas filtruoti, galimai papildomai filtruojamas naudojant atsitiktinio sisteminio triukšmo filtrus (medianos).

2.2.3. Signalų kadravimo modelis

Signalas paruošiamas kadruoti į nustatytus dydžius, paruošiami nustatymai, duomenų struktūros siunčiamos tarp kadrų.

Testavimo procese signalo kadravimo modelis yra pritaikomas jau turimiems garso duomenims, tačiau realiu algoritmo panaudojimo atveju kadrai būtų sukaupiami iš realaus laiko duomenų. Šis skirtumas neturi jokios įtakos algoritmo veikimui

2.2.4. Kadro apdorojimo modelis

Kadro apdorojimo modelis tvarko informaciją, kuri perduodama tarp kadro, taip pat modelis filtruoja duomenis pagal kadro etapų sprendimus

Tarp kiekvieno kadro apdorojant signalą yra koreguojami ir pritaikomi svertų parametrai. Pagal šiuos parametrus yra apjungiami charakteristikų sprendimai. Būtina perduoti vientisą parametru struktūrą kiekvienam kadrai iš eilės norint gauti tolygų sprendimų pasiskirstymą.

2.2.5. Kadro etapo modelis

Kadro etapus sudaro visiškai atskiri charakteristikų išskyrimai ir signalo filtravimas prieš perduodant kitiems kadrams (pvz.: pirmajame etape filtruojami užtikrintai užteršti spektrai tam jog antrasis etapas gautų švaresnį signalą).

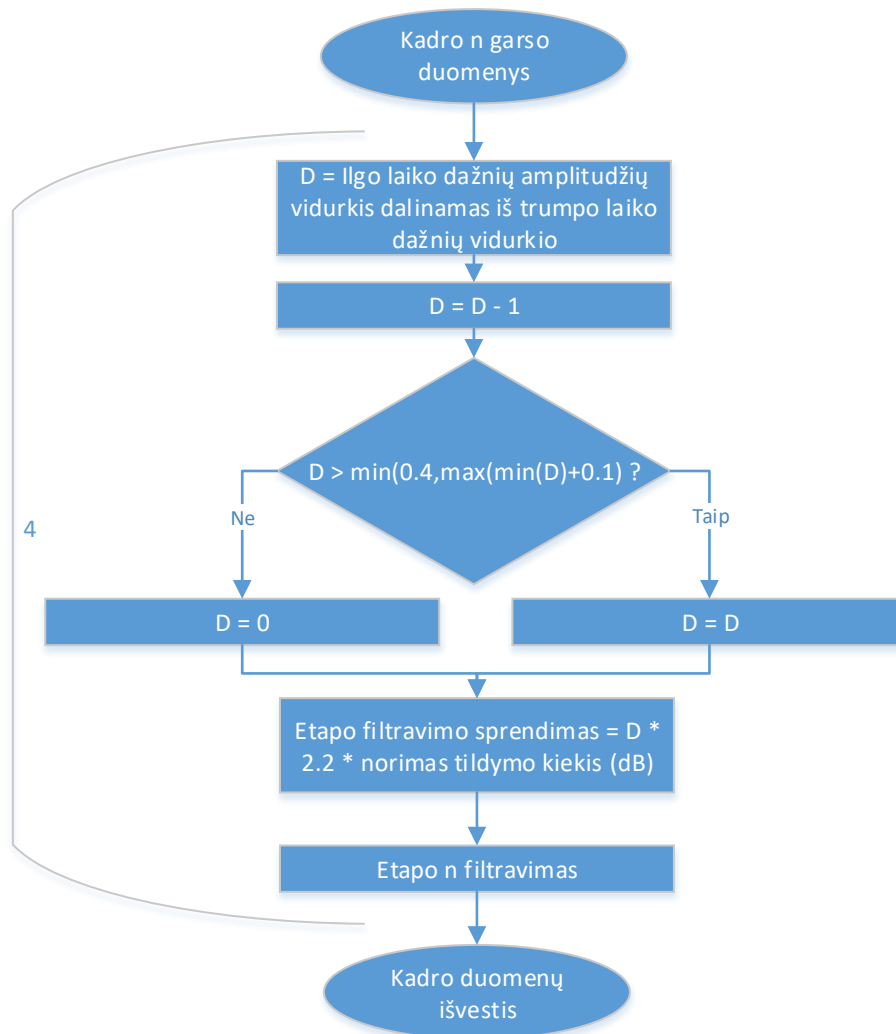
2.2.6. Algoritmo tyrimas

Pasirinkta tyrimo eiga, atsižvelgiant į akustinio triukšmo klasifikatorių trūkumą ir siauresnio spektro triukšmo filtravimų algoritmų perteklių:

- Pasiūlomas, bei tiriamas akustinio triukšmo filtravimo algoritmas, paremtas akustinio triukšmo klasifikavimu ir pasirinktų dažnių režijų filtravimu;
- Pasiūlytas algoritmas kombinuojamas su egzistuojančiais sprendimais, taip aprašant kombinavimo principus bei surandant efektyvų algoritmą, tinkamą akustinio triukšmo problemai spręsti.

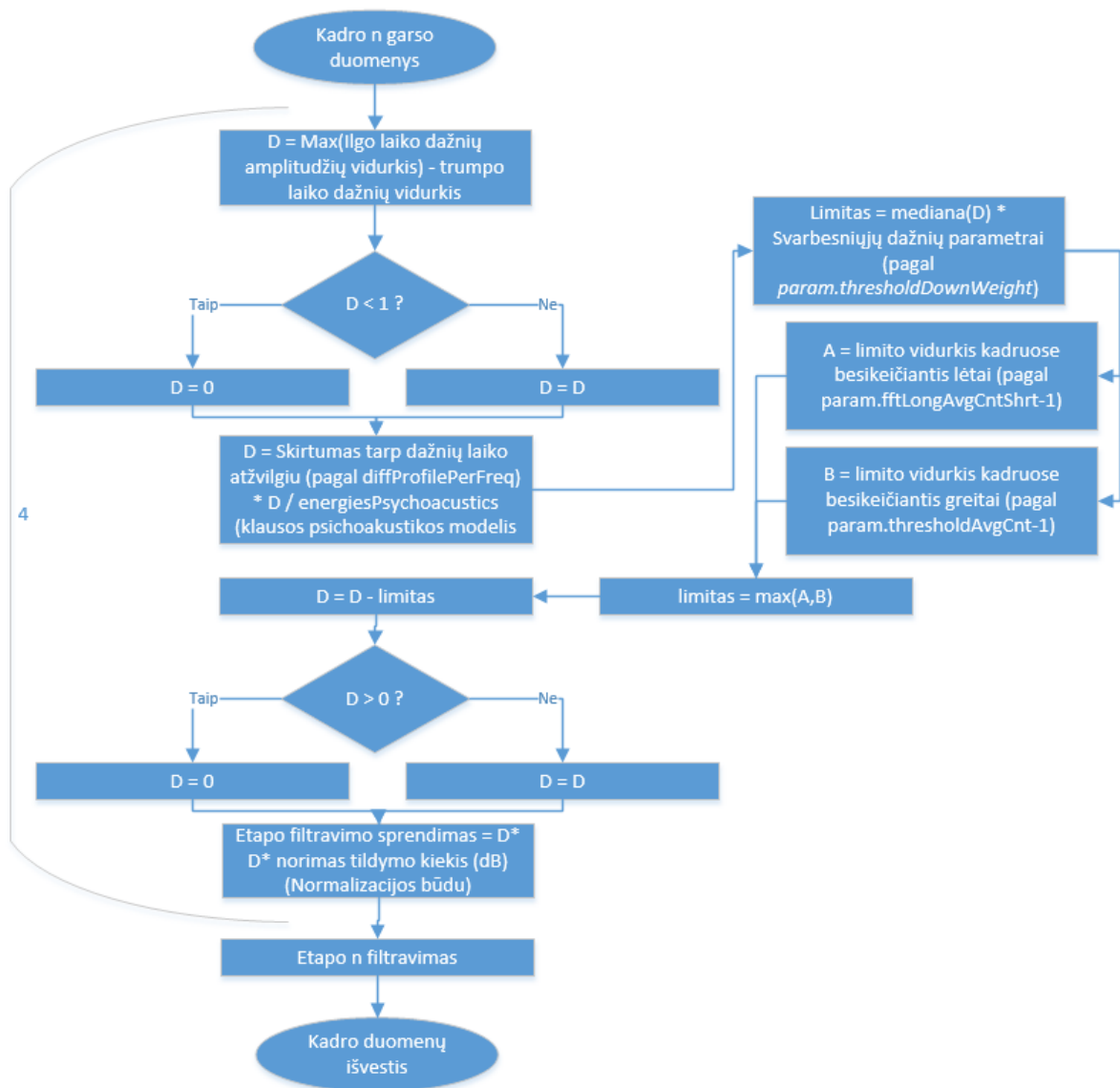
Tyrimo metu pasiūlytas algoritmas (žr. 3.3.1 skyrių) apjungia dvi charakteristikų kombinacijas:

- Monotoniškų spektrų filtras (19 pav.) (žr. 3.2.1.5 skyrių).
- Adaptyvaus limitu, maksimalios jėgos filtras (20 pav.) (žr. 3.2.1.6 skyrių).



19 pav. Monotoniškų spektrų filtras (žr. 3.2.1.5 skyrių)

Monotoniškų spektrų filtro (žr. 19 pav., 3.2.1.5 skyrių) tikslas - identifikuoti monotoniško spektro dažnius pasirinktame laiko tarpe. Filtras negeba identifikuoti reikšmingų garsų, nors pagal dizainą reikšmingi garsai visados išjungia filtravimą. Aptikus reikšmingus garsus pagal nustatymus filtravimas nutraukiamas tam tikram laiko tarpui. Siekiant tinkamai identifikuoti monotoniško triukšmo garsus - optimalus nutraukimo laiko tarpas nustatomas ilgas. Dėl to būtina šį filtravimo būdą kombinuoti su kitais filtrais.



20 pav. Adaptyvaus limito, maksimalios jėgos filtras (žr. 3.2.1.6 skyrių)

Adaptyvaus limito, maksimalios jėgos filtro tikslas - identifikuoti reikšmingą informaciją ir palaikyti sprendimų vientisumą išfiltruojant visus kitus dažnius (žr. 20 pav., 3.2.1.6 skyrių). Filtras neatskiria monotoniškų spektrų (sąlyginai nekintančio triukšmo) nuo reikšmingos informacijos. Pagal dizainą, esant tik akustiniam triukšmui, filtro išvestis artėja link sprendimo, jog visi garsai yra reikšmingi. Tyrimų metu, nustatyta jog toks dizainas yra būtinas, norint lanksčiai aptikti įvairius reikšmingus garsus. Dėl šio trūkumo šį filtravimo būdą būtina kombinuoti su kitais.

3. AKUSTINIO TRIUKŠMO FILTRAVIMO EKSPERIMENTINĖ DALIS

Siekiant pasiūlyti veiksmingą akustinio triukšmo filtravimo algoritmą, atlikti garso charakteristikų, jų apjungimo, kombinacijų, nustatymų tyrimai.

Šiame skyriuje:

- Ištiriamos garso charakteristikos (žr. 3.1 skyrių).
- Pasiūlomos ištirtų charakteristikų kombinacijos (žr. 3.2 skyrių).
- Pasiūlomas akustinio triukšmo filtravimo algoritmas (žr. 3.2.1 skyrių).
- Pasiūlomi egzistuojančių triukšmo analizės ir filtravimo algoritmų patobulinimai (žr. 3.3.5 skyrių).

Visi algoritmai tiriami naudojant 14 testavimo įrašų (žr. 2.1.2 skyrių), lyginant jų išeitį koreliacijos arba kitais būdais.

3.1. Akustinio triukšmo charakteristikų analizė

Siekama surasti reikšmingiausias akustinio triukšmo charakteristikas, norint pasiūlyti ir ištirti jų kombinacijas. Pasinaudojant ištirtais analizės ir filtravimo algoritmais (žr. 1 skyrių), tiriamos galimai reikšmingiausių (šio tyrimo tikslui pasiekti) garso charakteristikų veiksmingumas.

Daugelis garso charakteristikų analizių yra paremtos garso skaidymu kadrais. Siekiant garsą analizuoti realiu laiku, tenka analizuoti tik mažas (naujausias) jo dalis vienu metu, pvz.: 25 milisekundžių kadrais.

Prieš išrenkant garso charakteristikas patartina signalą filtruoti (žr. 3.1.1 skyrių), atsitiktinis triukšmas gali stipriai paveikti charakteristikų algoritmus apmokymo ar/ir analizės metu.

Aprašomos charakteristikos bei įvertinama jų išeitis kadruotą informaciją koreliuojant su akustinio triukšmo duomenimis.

3.1.1. Slenkančio vidurkio filtras

Daugelis charakteristikų išskyrimo algoritmų yra jautrūs triukšmui duomenyse, kadangi norima filtruoti garsus girdimus žmogui, prieš charakteristikų išskyrimą verta filtruoti duomenis. Tam naudojamas medianos filtras (žr. 21 pav.).

Tai yra slenkančio vidurkio filtras, tinkamas signalo formos glodinimui bei atsitiktinio triukšmo mažinimui.

Filtro išėjimo reikšmė gaunama skaičiuojant įėjimo reikšmių vidurkį pagal formulę (3.1), kur $x[i]$ yra įėjimo signalas, $MED[i]$ – išėjimo signalas, M vidurkiui skaičiuoti reikalingų reikšmių skaičius:

$$MED[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \quad (3.1)$$

Skaičiavimo pavyzdys (3.2), kai $M = 5$, $i = 50$:

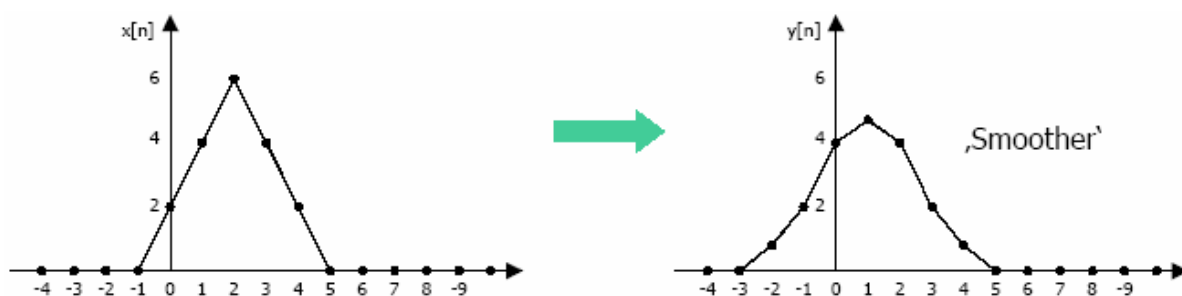
$$MED[50] = \frac{x[50]+x[51]+x[52]+x[53]+x[54]}{5} \quad (3.2)$$

Iš (3.2) formulės matyti, jog išėjimo reikšmė skaičiuojama tik iš ateities duomenų, verta pataisyti formulę jog išėjimo reikšmė būtų skaičiuojama simetriškai vidurkį apskaičiuojant naudojant praeities bei ateities duomenis. Nauja formulė (3.3):

$$MED[i] = \frac{1}{M} \sum_{j=-(M-1)/2}^{(M-1)/2} x[i+j] \quad (3.3)$$

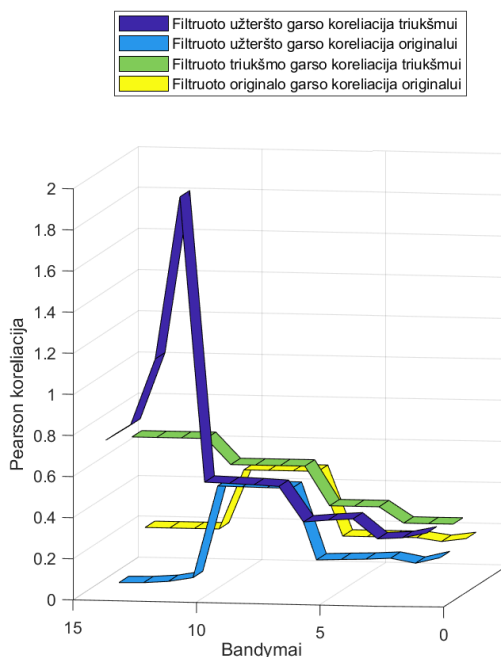
Skaičiavimo pavyzdys (3.4), kai $M = 5$, $i = 50$:

$$MED[50] = \frac{x[48]+x[49]+x[50]+x[51]+x[52]}{5} \quad (3.4)$$



21 pav. Signalų glodinimo pavyzdys

Tiriant filtro pritaikymo galimybes pastebėta, jog filtras iškraipo švarų garso įrašą. Filtras tinkamas naudoti tik charakteristikų duomenų filtravimui (svyravimų sumažinimui). Pateikiamas tyrimų koreliacinis grafikas (žr. 22 pav.).



22 pav. Slenkančio vidurkio filtro duomenų koreliacijos rezultatų grafikas

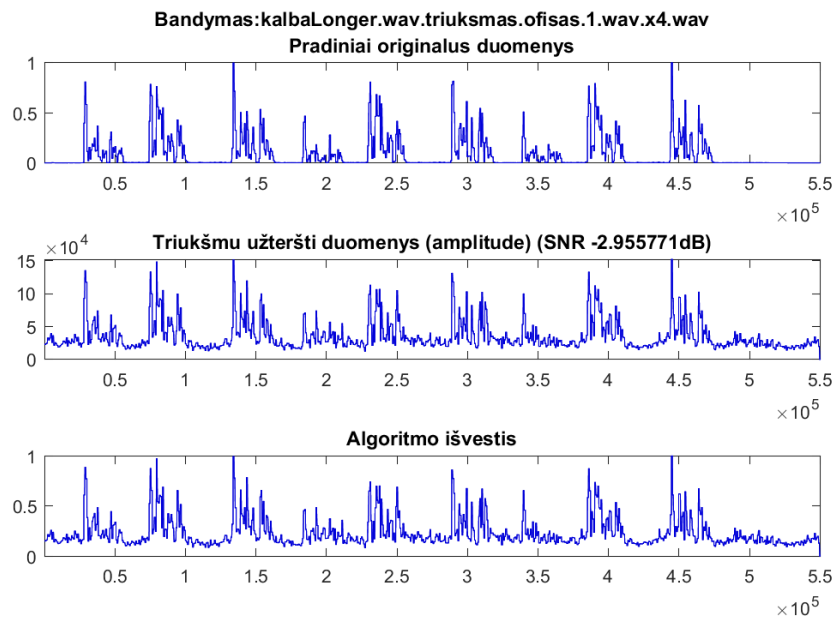
3.1.2. Dažnių energija

Garsas yra slėgio svyravimai terpėje, garso bangos perneša energiją. Akivaizdu, jog energija yra viena pagrindinių garso charakteristikų naudojama daugelyje algoritmų.

Garso energija kadre n apskaičiuojama pagal formulę (3.5), kur $|s(n + i)|$ yra garso kadras ir M yra duomenų kiekis kadre.

$$E(n) = \sum_{i=-M/2}^{M/2-1} |s(n + i)| \quad (3.5)$$

Tiriant pastebėta, jog garso energija reikšmingai padidėja pasireiškus neužterštiems garsams. Galima daryti prielaidą jog net ir plataus spektro triukšmas neperneša tiek daug energijos. Dėl šių charakteristikų garso energija yra viena tinkamiausių charakteristikų tirti kitas charakteristikas, lyginant kitų charakteristikų koreliaciją su garso energijos (žr. 23 pav.).



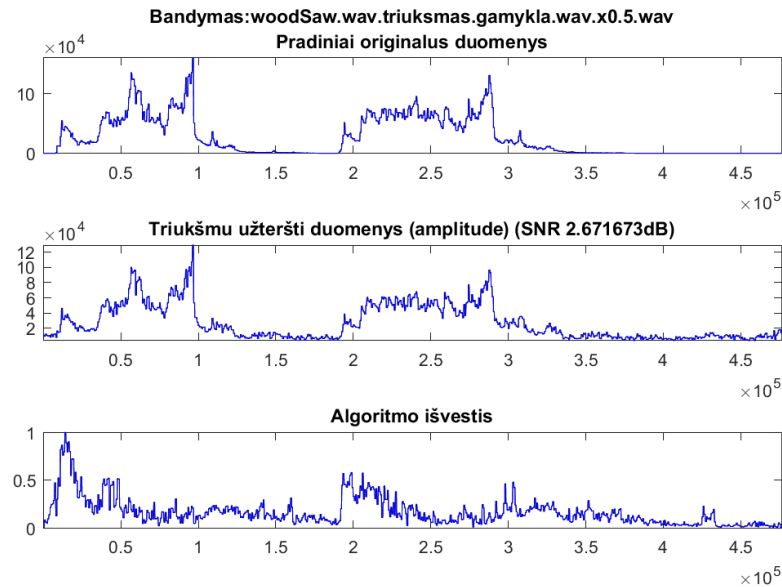
23 pav. Energijos charakteristikos išskyrimo palyginimas su duomenimis

3.1.3. Pilnos juostos energija (naudojama G.729 BAA)

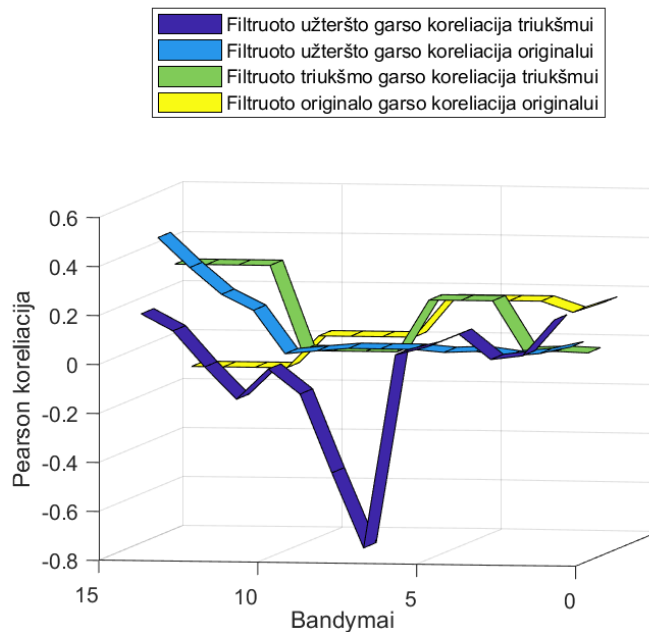
Garso energijos charakteristika naudojama ITU G.729 BAA algoritme.

Pilnos juostos energija yra apskaičiuojama pagal formulę (3.6), kur $R(0)$ yra pirmi 13 autokoreliacijos koeficientų, o N yra lango dydis naudojamas autokoreliacijos apskaičiavimams:

$$E_f = 10 \cdot \log \left[\frac{1}{N} R(0) \right] \quad (3.6)$$



24 pav. Pilnos juostos energijos charakteristikos palyginimas su duomenų amplitude



25 pav. Pilnos juostos energijos charakteristikos išėjties duomenų koreliacijos rezultatų grafikas

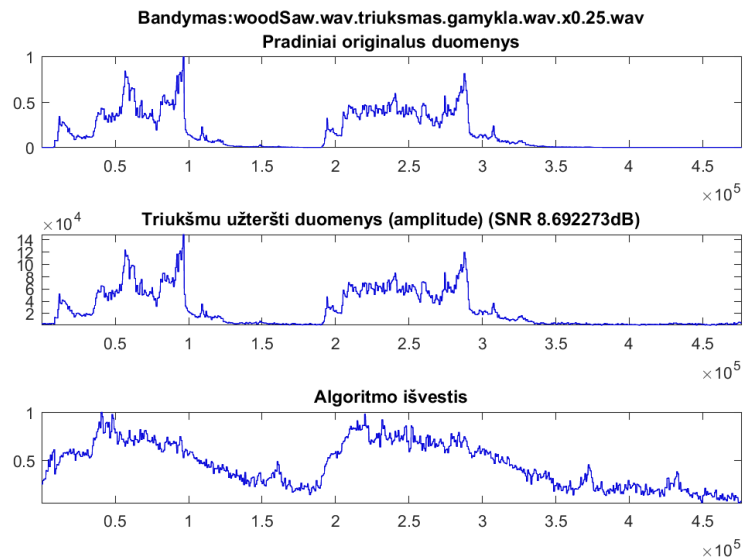
Atlikus tyrimus (žr. 24 pav., 25 pav.) nustatyta jog ši charakteristika stipriau koreliuoja su triukšmu nei su norimais garsais. Galimai charakteristika gali būti panaudota triukšmui nustatyti.

3.1.4. Nulio kirtimo dažnis (naudojamas G.729 BAA)

NKD (*Angl. ZCR – zero crossing rate*) garso charakteristika yra naudojama silpnų žmogaus balso friktyvų aptikimui nes jie yra sunkiai aptinkami naudojant paprastus energijos algoritmus, tačiau gali identifikuoti ir daugelį kitų similiarų garsų.

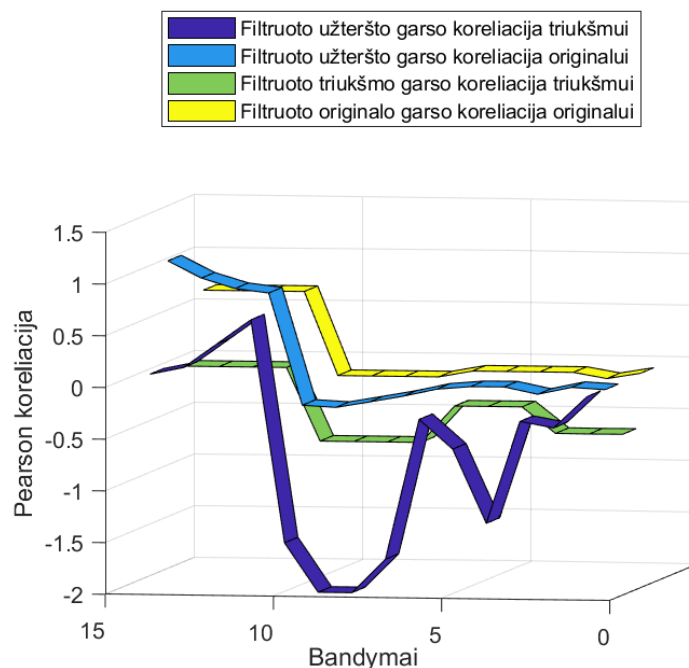
NKD algoritmas apskaičiuojamas pagal formulę (3.7), kuri patikrina ar ženklas keitėsi dviejų tarp dviejų gretimų elementų, analoginės garso išraiškos duomenyse.

$$NKD = \frac{1}{M} \sum_{i=1}^M \frac{[|sgn[s(i)] - sgn[s(i-1)]|]}{2} \quad (3.7)$$



26 pav. Nulio kirtimo dažnio charakteristikos išskyrimo pavyzdys

Ištyrus nulio kirtimo dažnio charakteristiką pastebėta, jog ji reikšmingai neigiamai koreliuoja su akustinę triukšmą perteikiančiais garsais ir esant pastoviam triukšmui nekinta, tačiau ne visuose garsų spektruose pateikia norimą rezultatą. Pastebėta, jog nulio kirtimo parodymai šokteli ties reikšminga informacija (žr. 26 pav., 27 pav.).



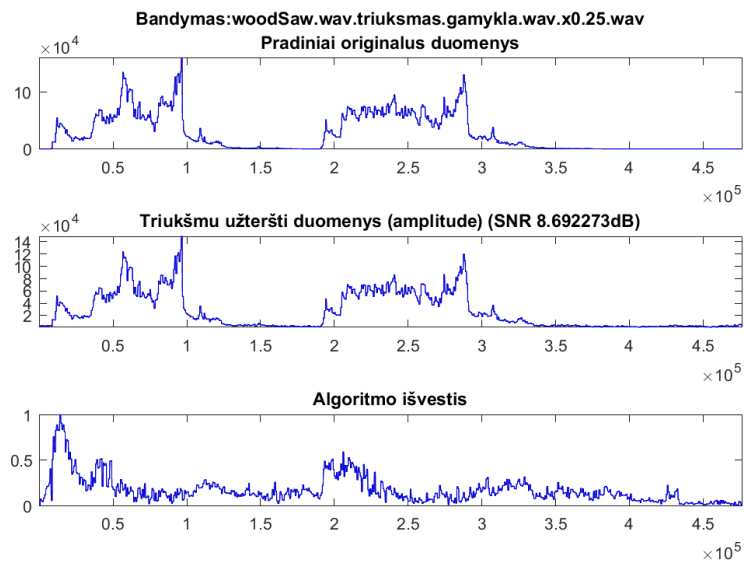
27 pav. Nulio kirtimo dažnio charakteristikos išeities duomenų koreliacijos rezultatų grafikas

3.1.5. Spektrinė analizė (G.729 BAA)

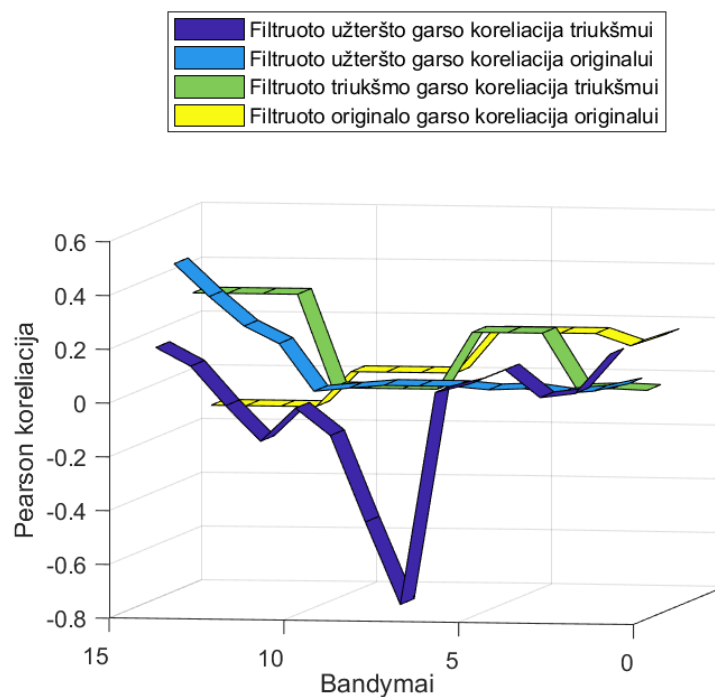
Spektrinė analizė leidžia aptikti tam tikrų dažnių svyravimus garse. G.729 BAA algoritmas dažnių analizei naudoja spektrinių dažnių (*Angl. spectral frequencies (LSF)*) charakteristikos algoritmą, taip pat vadinamą spektrinių linijinių porų (*Angl. line spectral pairs (LSP)*) algoritmu.

Algoritmas remiasi linijiniu prognozavimu, pagal formulę (3.8), kur $x(n)$ prognozuota signalo vertė, $x(n - i)$ yra praeitos vertės ir a_i praėję koeficientai:

$$x(n) = \sum_{i=1}^p a_i x(n - i) \quad (3.8)$$



28 pav. Spektrinės analizės palyginimas su akustiniu triukšmu užterštų duomenų amplitude



29 pav. Spektrinės analizės charakteristikos išėities duomenų koreliacijos rezultatų grafikas

Matyti, jog spektrinės analizės išeitis apskaičiuojama šiuo būdu yra mažai naudinga atskiriant reikiamus garsus nuo triukšmo. Pastebėta, jog spektrinė analizė yra daug vientisesnė kai duomenys daugiau užteršti (žr. 28 pav., 29 pav.).

3.1.6. Dominuojančios dažnių komponentės

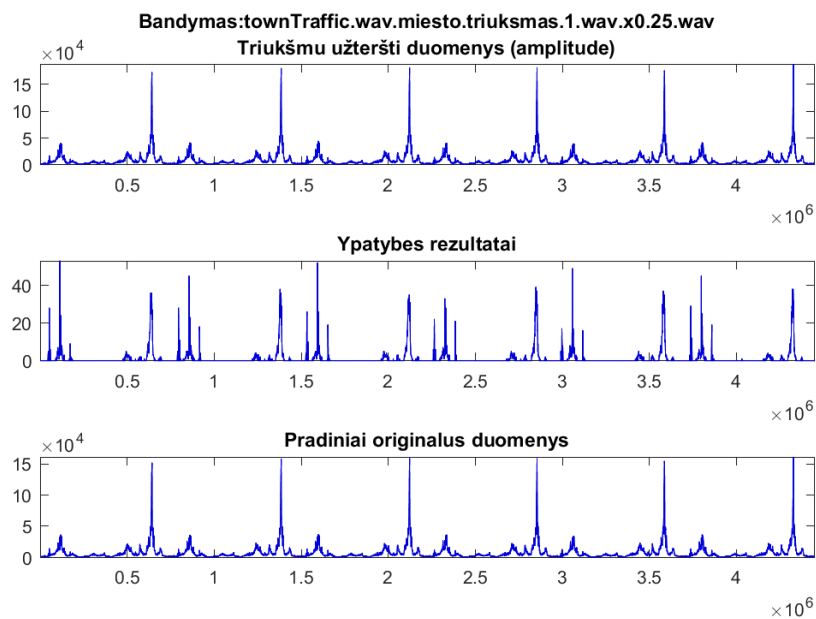
Dominuojančių dažnių komponentių paieška, nors ir gali atrodyti primityvi, tačiau tinkamai optimizavus šios charakteristikos išgautus duomenis galima efektyviai atskirti daugelį garsų, remiantis dažnių duomenimis.

Pritaikius GFT galima rasti n dominuojančių dažnių pagal jų amplitudę, naudojant formulę (3.9), kur N yra GFT dažnių spektras, $N(i)$ yra randamas dominuojantis dažnis ir tuomet jis bei jo j kaimynų kiekis yra panaikinami iš GFT amplitudžių paieškos spektro:

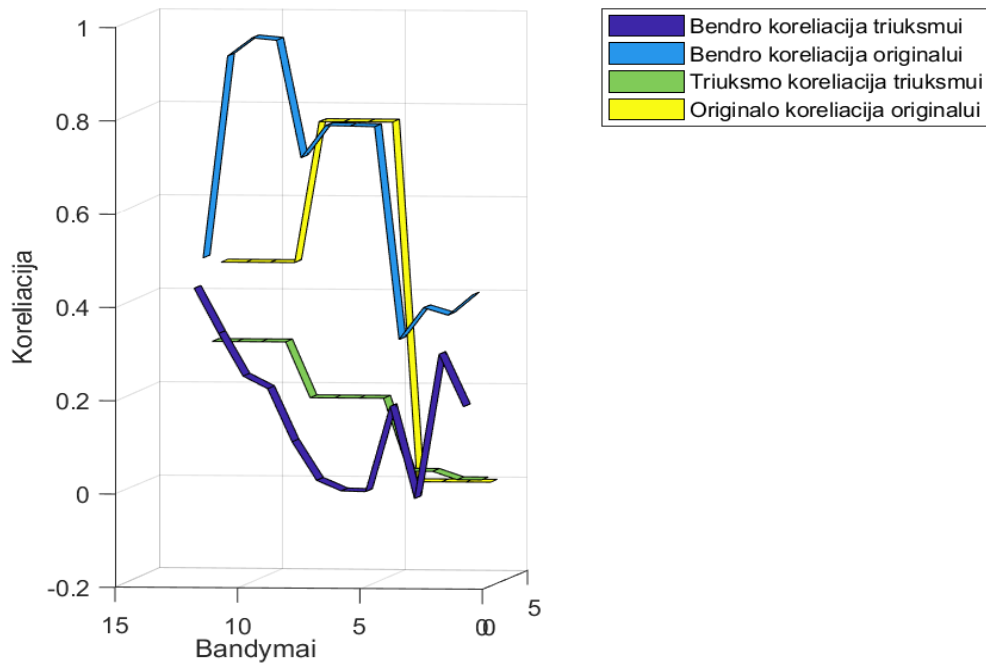
$$N(i) = \max(N), N(i - 1) = 0 \dots N(i - j) = 0, N(i) = 0, N(i + 1) = 0 \dots N(i + j) = 0 \quad (3.9)$$

Papildomai, galima paiešką optimizuoti išskiriant minimalų amplitudės padidėjimą, nuo praicitų amplitudžių esančių už j GFT spektro kaimynų kiekio.

Suradus dominuojančių dažnių amplitudžių kiekį ir dominuojančias amplitudes filtruojant pagal minimalų amplitudės padidėjimą (50%) gaunama labai koreliatyvi išeitis neužterštajam signalui. Ši charakteristika geba atskirti mums įprastą akustinį triukšmą (žr. 30 pav., 31 pav.).



30 pav. Dominuojančios dažnių komponentių palyginimas su duomenų amplitude



31 pav. Dominuojančių dažnių komponentių charakteristikos išeities duomenų koreliacijos rezultatų grafikas

3.1.7. Spektro tiesiškumas

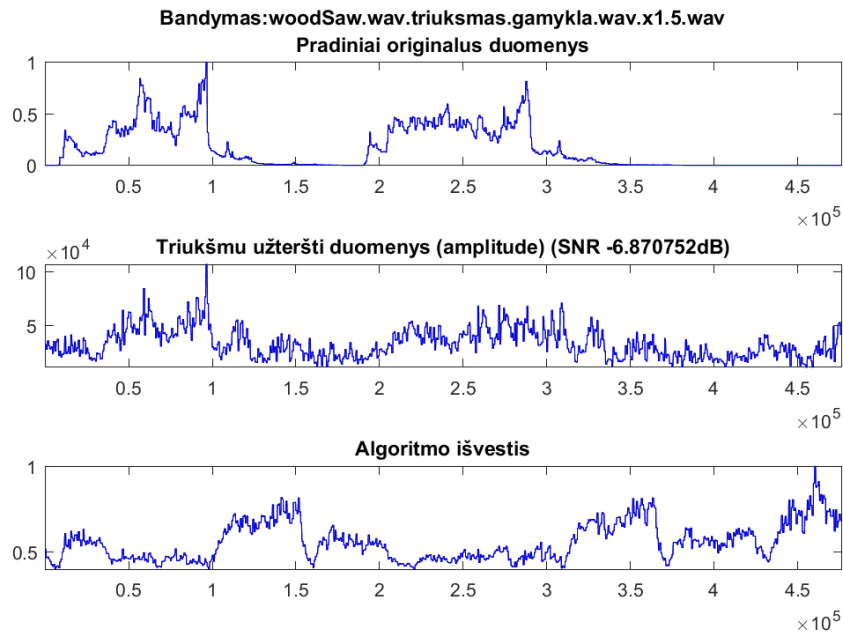
Spektro tiesiškumas apibūdina spektro triukšmingumą (tiesiškumą, dekoreliaciją). Jis apskaičiuojamas, apskaičiuojant santykį tarp geometrinio bei aritmetiniu spektro reikšmių vidurkių pagal formulę (3.10), kur b yra dažnio juostų kiekis, $SFM(b)$ yra spektro tiesiškumas tose juostose, $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a :

$$SFM(b) = \frac{(\prod_{k \in b} a(k))^{\frac{1}{k}}}{\frac{1}{k} \sum_{k \in b} a(k)} \quad (3.10)$$

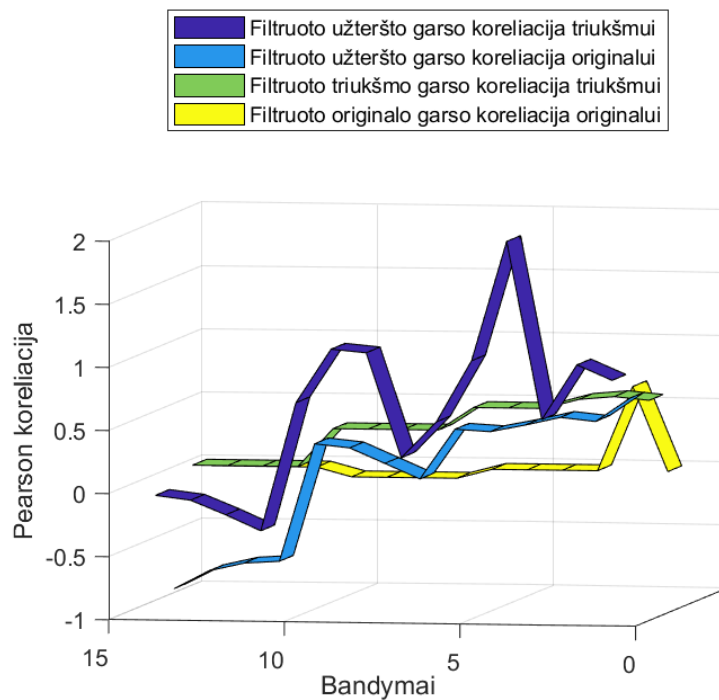
Toningiems signalams SFM turėtų būti arti 0, triukšmingiems signalams SFM turėtų būti arti 1. Praktiškesnis SFM skaičiavimas gali būti atliekamas pagal formulę (3.11):

$$SFM_{ab} = 10 * \frac{1}{N} \sum_k (\log_{10} a(k) - \log_{10} u) \quad (3.11)$$

Tyrimų metu paaiškėjo jog įprastas spektro tiesiškumas nesikoreliuoja su duomenimis, tačiau stebint pateiktus charakteristikos rezultatų grafikus galima teigti jog charakteristika geba atpažinti kada spektras yra triukšmingas ir kada girdimi norimieji garsai. Ši charakteristika geba atskirti plataus spektro akustinį triukšmą nuo reikšmingų garsų (žr. 32 pav., 33 pav.).



32 pav. Spektro tiesiškumo charakteristikos išėities pavyzdys



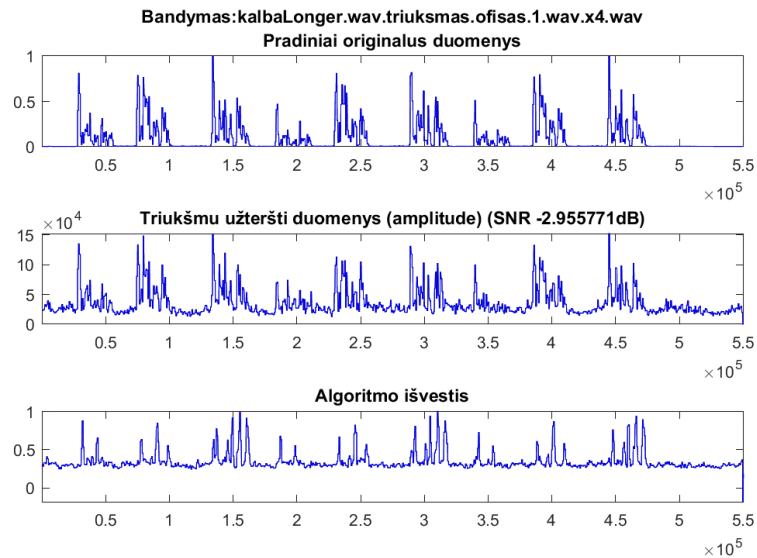
33 pav. Spektro tiesiškumo charakteristikos išėities duomenų koreliacijos rezultatų grafikas

3.1.8. Spektro geometrinis tiesiškumas

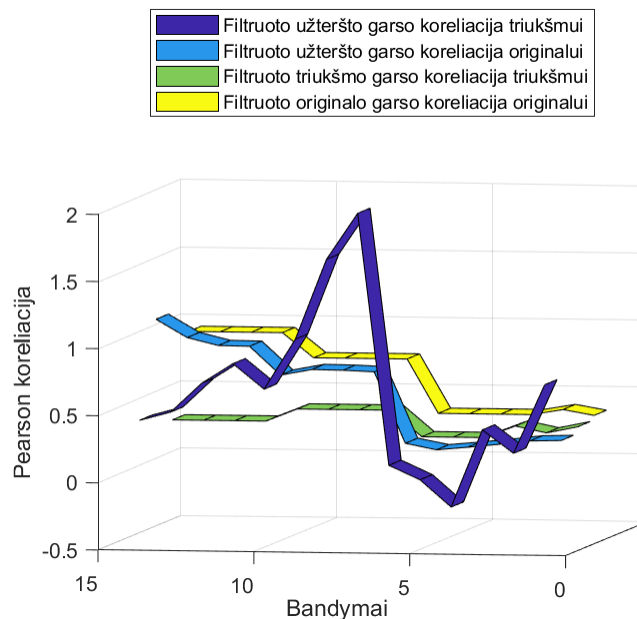
Spektro tiesiškumas apibūdina spektro triukšmingumą. Geometrinis tiesiškumas apskaičiuojamas pagal praeito skyriaus(3.1.7) formulę (3.11), kur b yra dažnio juostų kiekis, $SGT(b)$ yra spektro tiesiškumas tose juostose, $a(k)$ yra amplitudė k dažnio juostoje:

$$SGT(b) = \left(\prod_{k \in b} a(k) \right)^{\frac{1}{k}} \tag{3.12}$$

Tyrimų metu pastebėta, jog charakteristikos išėitis stipriai koreliuoja su originaliu garso įrašu ir pasižymi mažesne koreliacija triukšmui. Stebint pateiktus charakteristikos rezultatų grafikus galima teigti, jog charakteristika geba atskirti akustinį triukšmą nuo reikšmingų garsų. Panašu, jog charakteristika pateikia gerus rezultatus esant plataus spektro triukšmui (žr. 34 pav., 35 pav.).



34 pav. Spektro geometrinio tiesiškumo išėitis palyginus su įrašo amplitude



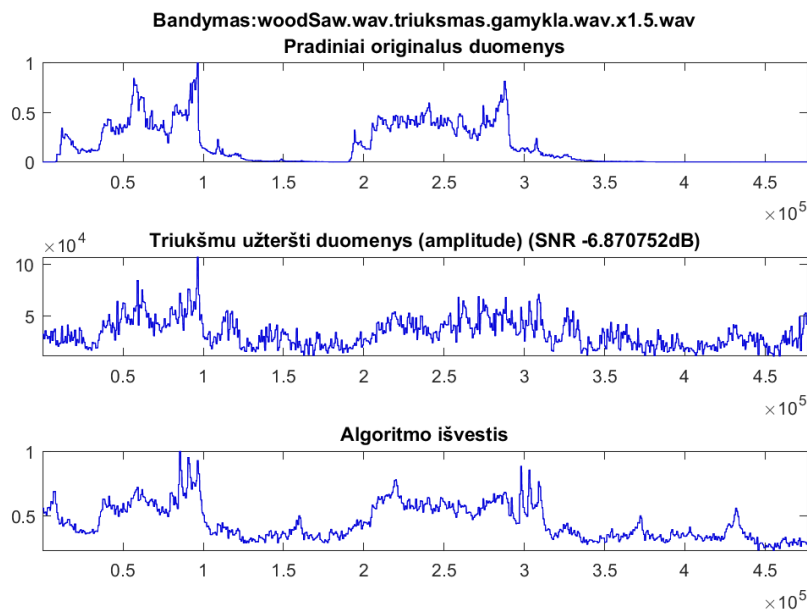
35 pav. Spektro geometrinio tiesiškumo charakteristikos išėities duomenų koreliacijos rezultatų grafikas

3.1.9. Spektro aritmetinis tiesiškumas

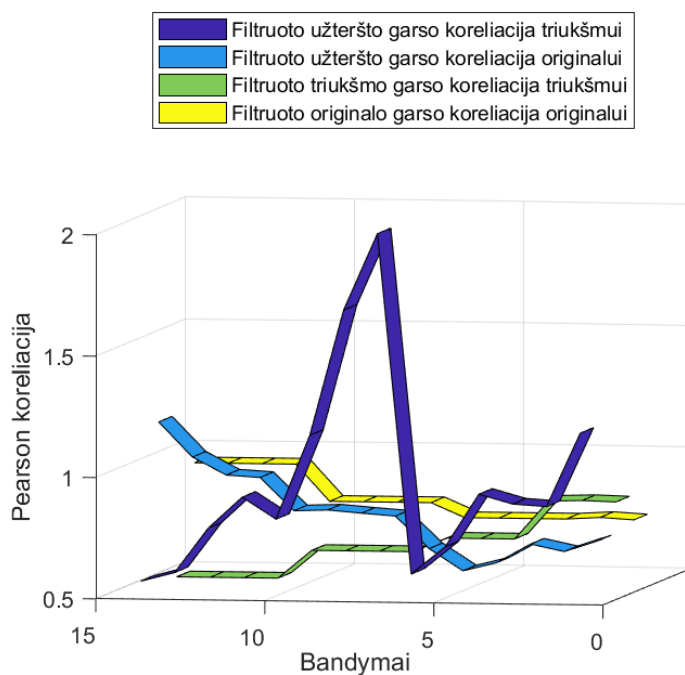
Spektro tiesiškumas apibūdina spektro triukšmingumą. Geometrinis tiesiškumas apskaičiuojamas pagal skyriaus(3.1.7) formulę (3.11), kur b yra dažnio juostų kiekis, $SAT(b)$ yra spektro tiesiškumas tose juostose, $a(k)$ yra amplitudė k dažnio juostoje:

$$SAT(b) = \frac{1}{K} \sum_{k \in b} a(k) \quad (3.13)$$

Tyrimų metu pastebėta jog charakteristikos išeitis stipriai koreliuoja su originaliu garso įrašu ir pasižymi mažesne koreliacija triukšmui. Stebint pateiktus charakteristikos rezultatų grafikus galima teigti jog charakteristika geba atpažinti kada spektras yra triukšmingas ir kada girdimi norimieji garsai. Ši charakteristika būtų veiksminga esant plataus spektro triukšmui (žr. 36 pav., 37 pav.).



36 pav. Spekro aritmetinio tiesiškumo išeitis palyginus su įrašo amplitude

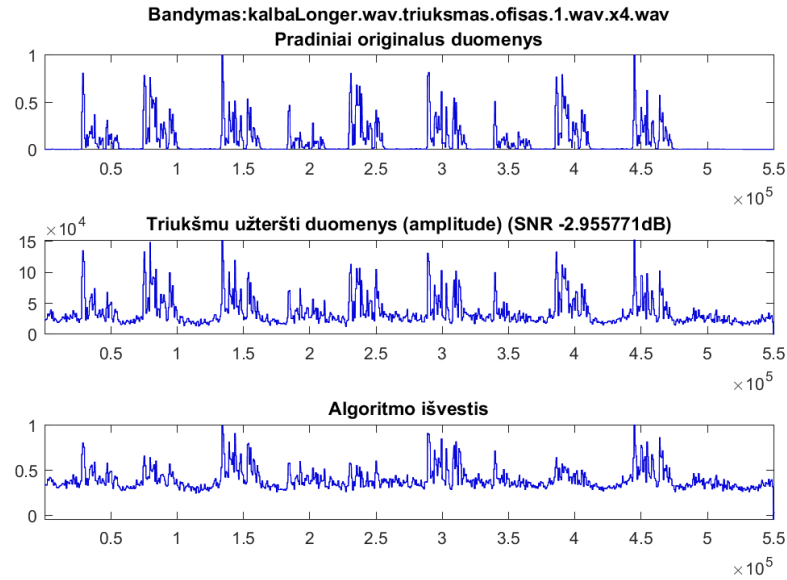


37 pav. Spekro aritmetinio tiesiškumo charakteristikos išeities duomenų koreliacijos rezultatų grafikas

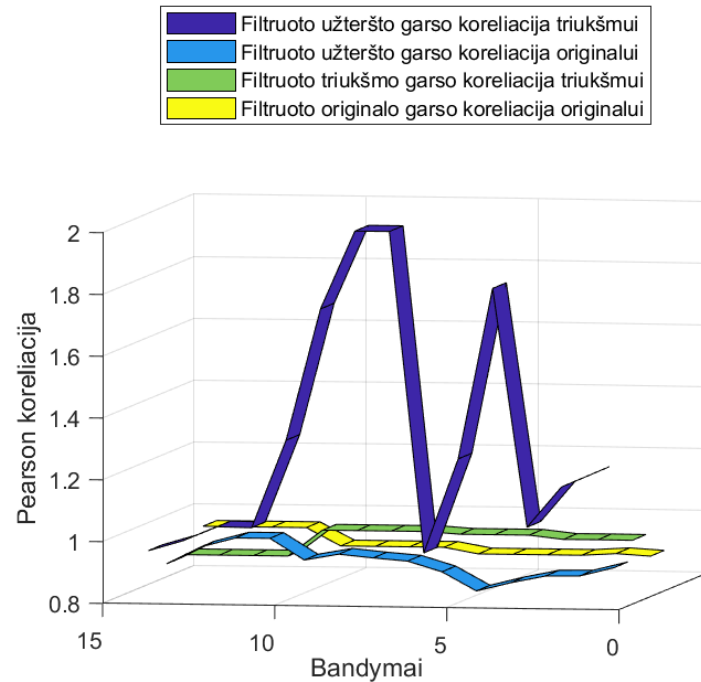
3.1.10. Vidutinė kvadratinė vertė

Vidutinė kvadratinė vertė (angl. *RMS – root mean square*) apskaičiuojama pagal formulę (3.14) kur b yra dažnio juostų kiekis, $RMS(b)$ yra spektro tiesiškumas tose juostose, $a(k)$ yra amplitudė k dažnio juostoje

$$RMS(b) = \sqrt{1/k(a(k)_1^2 + a(k)_2^2 + \dots + a(k)_n^2)} \quad (3.14)$$



38 pav. Vidutinės kvadratinės vertės išėities palyginimas su garso įrašo amplitude



39 pav. Vidutinės kvadratinės vertės charakteristikos išėities duomenų koreliacijos rezultatų grafikas

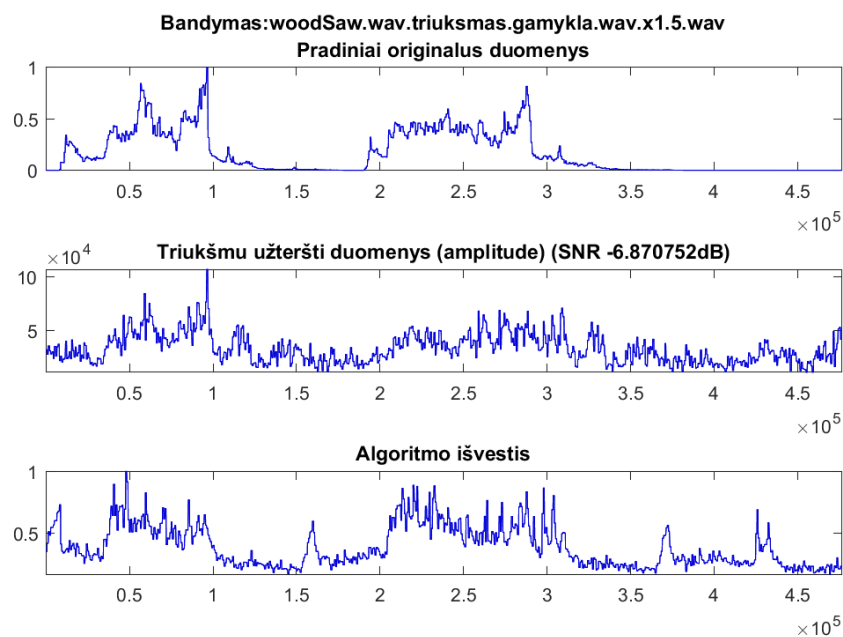
Tyrimų metu pastebėta, jog charakteristikos išeitis stipriai koreliuoja su originaliu garso įrašu, tačiau pasižymi didesne koreliacija triukšmui. Charakteristika geriau koreliuoja su triukšmu nei originaliu garso įrašu, pasitelkus papildomą apdorojimą, toks koreliacijos skirtumas, leistų atskirti triukšmingą garsą (žr. 38 pav., 39 pav.).

3.1.11. Spektro centroidė

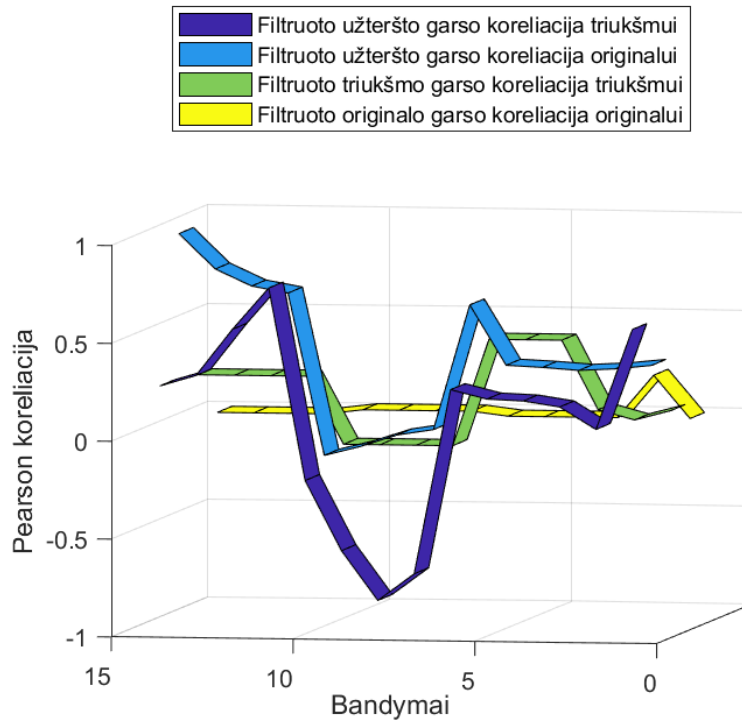
Spektro centroidė yra randama kaip įprasto duomenų pasiskirstymo centroidė 2D erdvėje pagal formulę (3.16), kur x yra dažnis, o $p(x)$ yra galimybė aptikti x : $p(x)$:

$$u = \int x \cdot p(x) \quad (3.15)$$

$$p(x) = \frac{\text{amplitudė}(x)}{\sum_x \text{amplitudė}(x)} \quad (3.16)$$



40 pav. Spektro centroidės kitimas pravažiuojančių mašinų įraše (aukštesni dažniai virsta žemesniais)



41 pav. Spektro centroidės charakteristikos išėities duomenų koreliacijos rezultatų grafikas

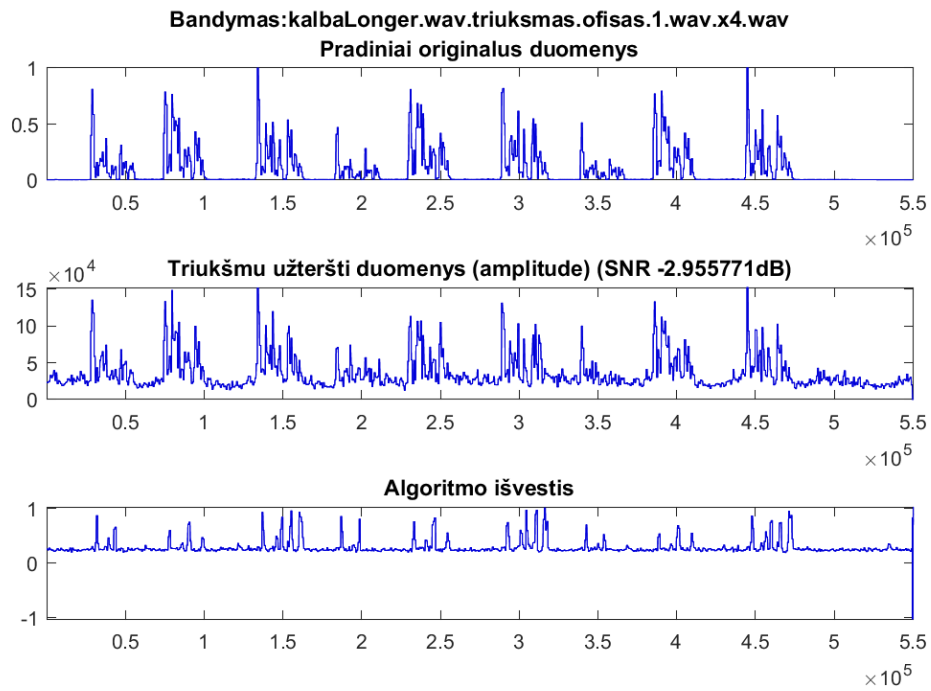
Matyti, jog centroidė nekoreliuoja su reikšmingų garsų amplitude, tačiau esant neužterštiems garsams centroidė išlaiko savo poziciją. Centroidė galėtų būti panaudota triukšmo aptikimui bei atskyrimui (žr. 40 pav., 41 pav.).

3.1.12. Spektro išsiskaidymas

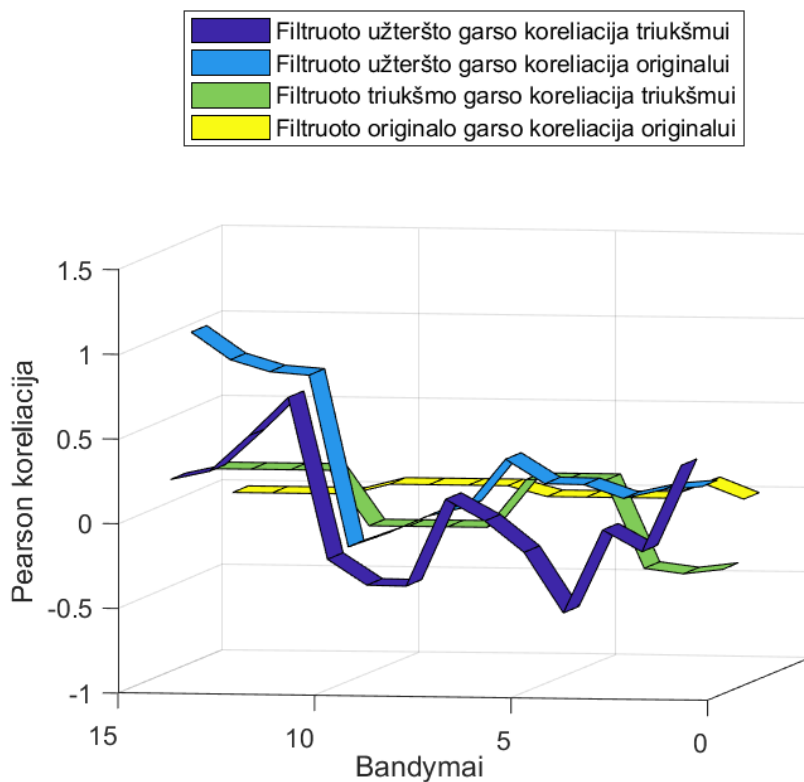
Spektro išsiskaidymas yra apskaičiuojamas remiantis pastarojo skyrio skaičiavimais (3.16). Spektro išsiskaidymas apskaičiuojamas kaip išsiskaidymas aplink spektro vidurkį (centroidę), t.y. distribuciją. Skaičiuojama pagal formulę (3.17), kur x yra dažnis, o $p(x)$ yra galimybė aptikti x :

$$\sigma^2 = \int (x - u)^2 \cdot p(x) \quad (3.17)$$

Spektro išsiskaidymas, atitinkamai spektro centroidei pateikia panašius (proporcingus) rezultatus lyginant kadruotą informaciją. Spektro išsiskaidymas koreliuoja tik su reikšmingais garsais su mažu triukšmo kiekiu (mažu SNR), tačiau nepateikia jokios reikšmingos informacijos (neskaitant triukšmo lygio) kituose įrašuose (žr. 42 pav., 43 pav.).



42 pav. Spektro išsiskaidymo palyginimas su užteršto garso įrašo amplitude



43 pav. Spektro išsiskaidymo charakteristikos išėities duomenų koreliacijos rezultatų grafikas

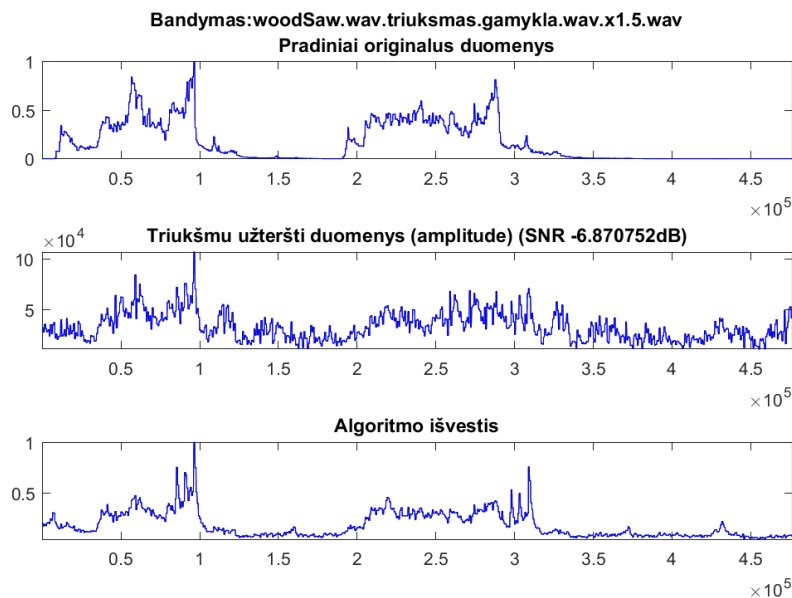
3.1.13. Koreliacija tarp kanalų naudojant kryžminę koreliaciją

Charakteristikos veikimo principas yra paremtas dviejų kanalų garso įrašu. Bandoma ištirti, ar koreliacija tarp kanalų gali koreliuoti su reikšmingais garsais arba triukšmu. Apskaičiuojama

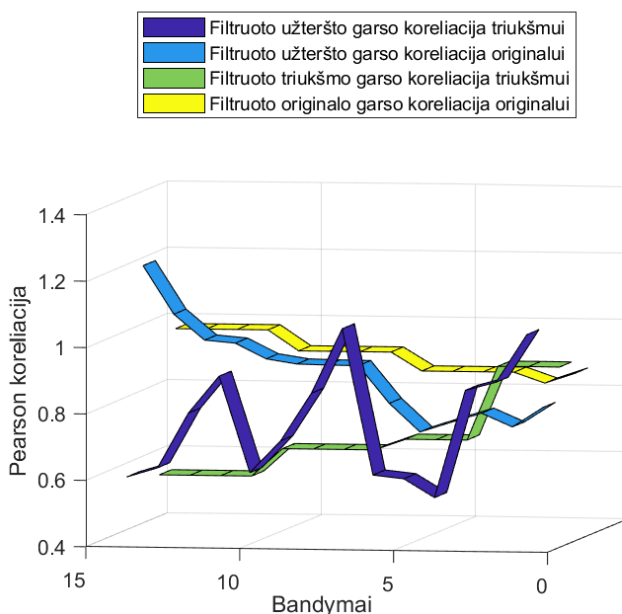
koreliacija tarp kairio ir dešinio įrašo kanalų naudojant kryžminės koreliacijos formulę (3.18), kur f^* yra m kompleksinis konjugatas, o n yra pasiskirstymas (vėlinimas):

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f * [m]g[m + n] \quad (3.18)$$

Tyrimų metu nustatyta jog ši charakteristika pasiteisino ir reikšmingai koreliuoja su originaliais garsais, atitinkamai mažai koreliuoja su akustinės taršos garsais (žr. 44 pav., 45 pav.).



44 pav. Charakteristikos išvedamų duomenų palyginimo pavyzdys



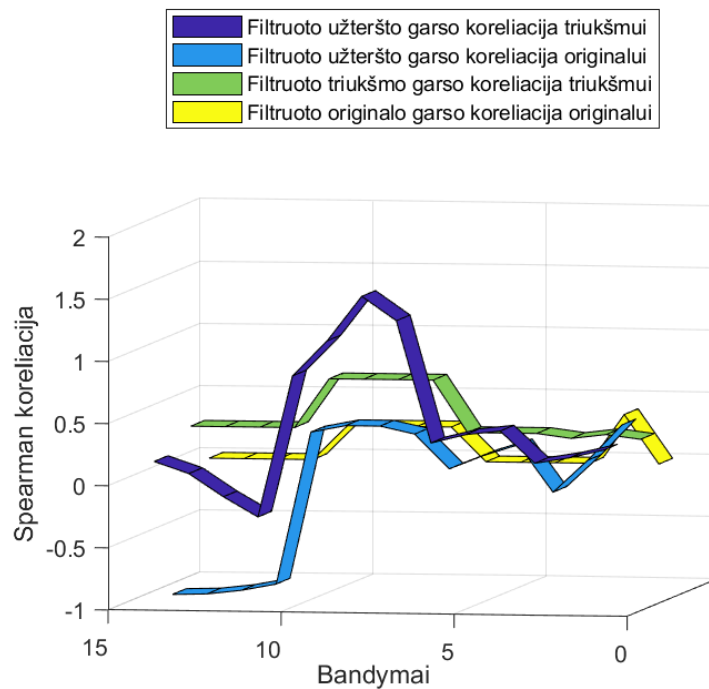
45 pav. Koreliacijos tarp kanalų naudojant kryžminę koreliaciją charakteristikos išėjties duomenų koreliacijos rezultatų grafikas

3.1.14. Koreliacija tarp kanalų pagal Pirsono koreliaciją

Charakteristikos veikimo principas yra paremtas dviejų kanalų garso įrašu. Bandoma ištirti, ar koreliacija tarp kanalų gali koreliuoti su reikšmingais garsais arba triukšmu. Apskaičiuojama koreliacija tarp kairio ir dešinio įrašo kanalų pagal Pirsono koreliacijos formulę (3.19):

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}} \quad (3.19)$$

Tyrimų metu įrodyta jog ši charakteristika nepasizymi teigiama koreliacija nei triukšmui nei norimiems garsams (žr. 46 pav.).



46 pav. Koreliacijos tarp kanalų charakteristikos išėjties duomenų koreliacijos rezultatų grafikas

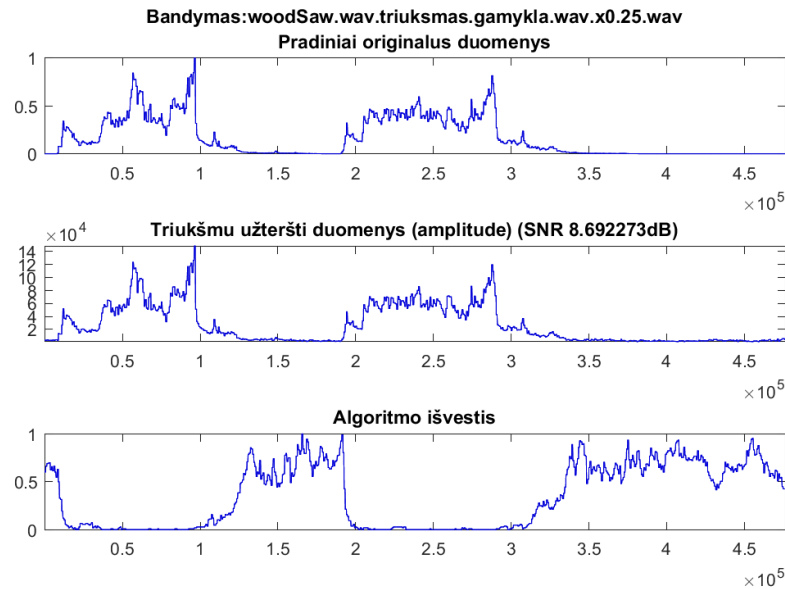
3.1.15. Mažos jėgos spektro kiekis

Charakteristika skirta atpažinti triukšmo kiekį duomenyse, turint omenyje jog mažo SNR signalas savo informaciją pateikia siauro dažnio spektre.

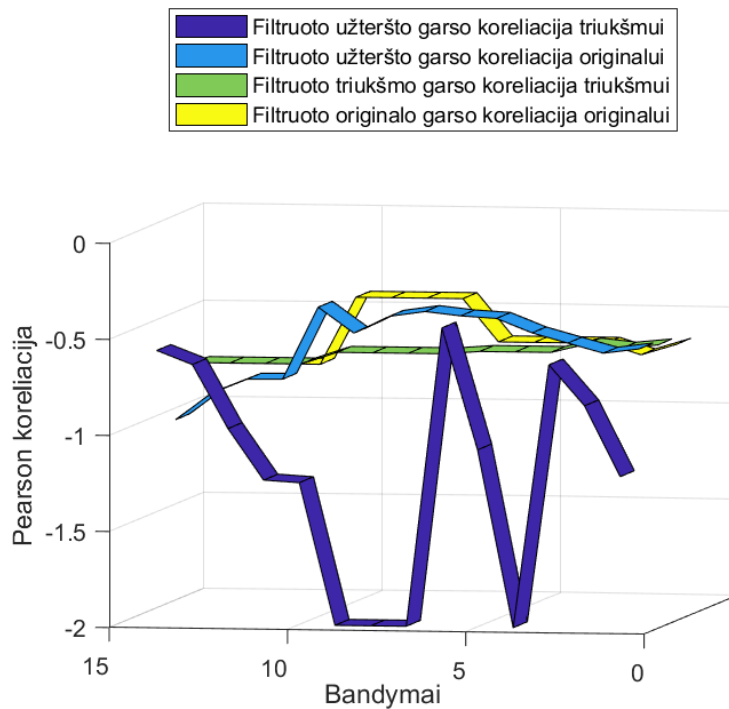
Mažos jėgos spektro kiekis paskaičiuojamas pagal formulę MJS , kur $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a (3.20)

$$MJS = \sum_{k=1}^n [a(k) > (\bar{a} * std(a) * 2)] \quad (3.20)$$

Tyrimų metu pastebėta, jog ši charakteristika reikšmingai koreliuoja su įrašo triukšminga dalimi (žr. 47 pav., 48 pav.).



47 pav. Mažos jėgos spektro kiekio charakteristikos išeities duomenų pavyzdys

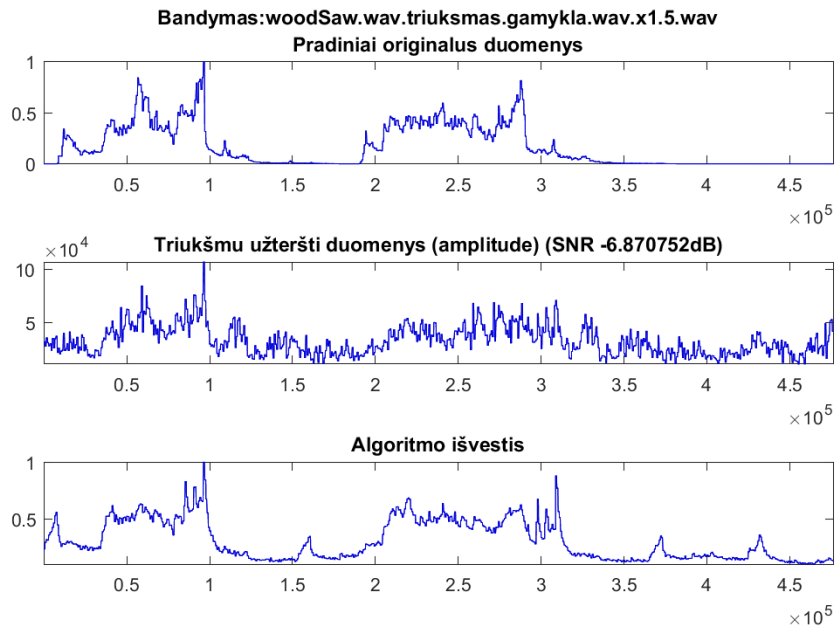


48 pav. Mažos jėgos spektro kiekio charakteristikos išeities duomenų koreliacijos rezultatų grafikas

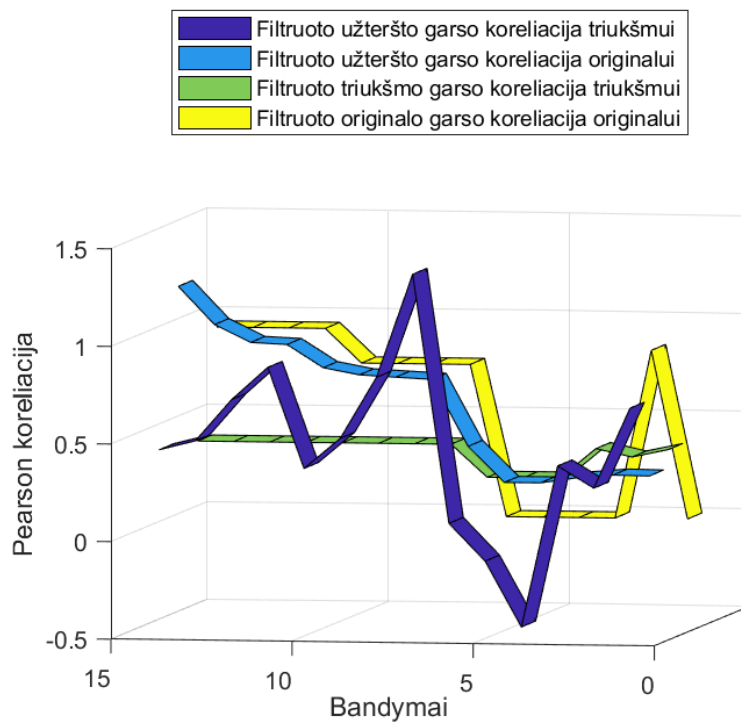
3.1.16. Dažnių triukšmingumas

Dažnių triukšmingumas $DT(k)$ nustatomas pagal formulę (3.21), kur k yra tam tikras dažnis, kur $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a , $d(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje d :

$$DT(k) = \frac{a(k)}{d(k)} \quad (3.21)$$



49 pav. Dažnių triukšmingumo charakteristikos rezultatų pavyzdys



50 pav. Dažnių triukšmingumo charakteristikos išeities duomenų koreliacijos rezultatų grafikas

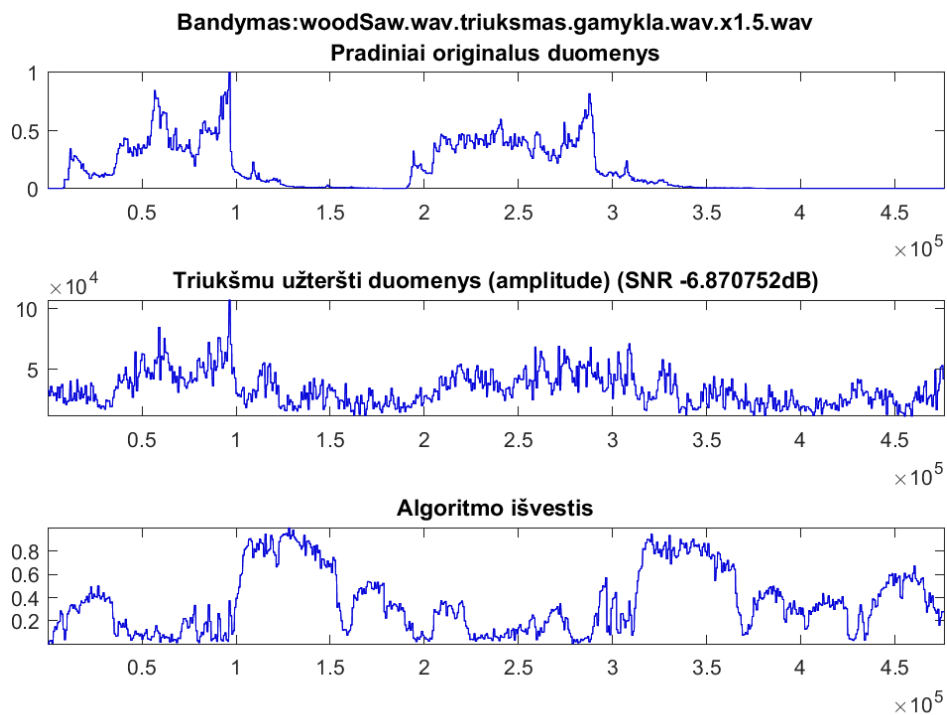
Tiriant pastebėta, jog dažnių triukšmingumo charakteristikos išėjimas stipriai koreliuoja su norimais garsais, bei atitinkamai - mažiau su triukšmu daugelyje testavimo atvejų. Ši charakteristika gali būti naudojama išskirti norimus garsus, arba aptikti dominuojančius reikšmingų garsų dažnius. Darant prielaidą, jog esant $SNR > 1$ dominuojantys dažniai daugelyje atvejų būtų norimieji garsai,

atvirkščiai,- kai $SNR < 1$. Tačiau triukšmo bei norimų garsų spektrai gali būti įvairiai pasiskirstę ir tai apskaičiuoti yra sunku (žr. 49 pav., 50 pav.).

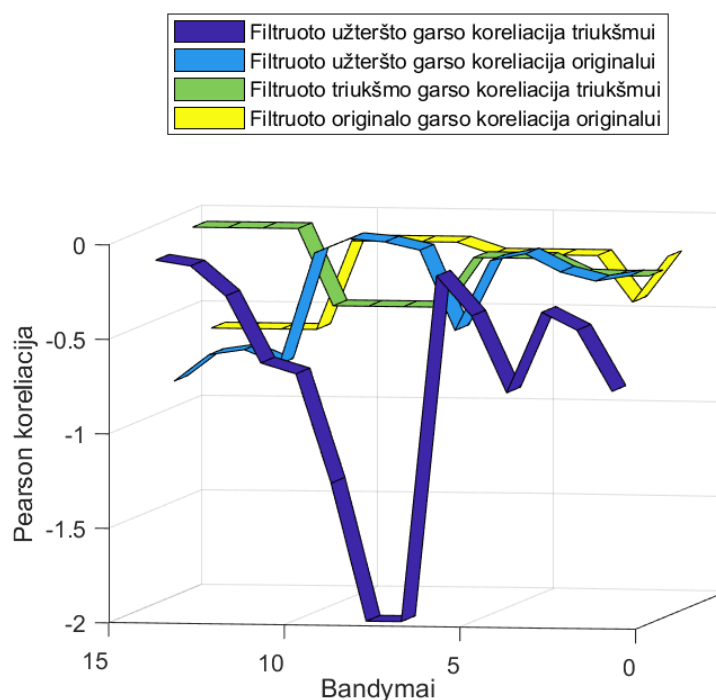
3.1.17. Spektrų atstumas

Spektrų atstumo palyginimo charakteristika lygina spektrų jėgas, toks palyginimas pasiūlytas S. F. Boll [36] spektrų atimties algoritme [36]. Spektrų atstumas lyginamas pagal formulę SPBA (3.22), kur $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a , $b(k)$ yra k amplitudė dažnių amplitudžių juostoje b (b išskaičiuojama iš (20 kadru) vidurkio).

$$SPBA = 1/n \sum_{i=1}^n 20 * (\log_{10}(a(k)) - \log_{10}(b(k))) \quad (3.22)$$



51 pav. Spektrų atstumo charakteristikos išeities duomenų pavyzdys



52 pav. Spektrų atstumo charakteristikos išėities duomenų koreliacijos rezultatų grafikas

Pastebėta, jog charakteristikos išvestis vidutiniškai nekoreliuoja nei su triukšmu nei su originaliu garsu, tačiau stebint grafikus keletose situacijose ji stipriai neigiamai koreliavo su triukšmo garsu. Galima teigti, jog charakteristiką įmanoma pritaikyti tam tikrose situacijose (žr. 51 pav., 52 pav.).

3.1.18. Akustinio triukšmo charakteristikų veiksmingumo vertinimas

Tiriant akustinio triukšmo charakteristikas, siekiant matematiškai palyginti jų išėitį, charakteristikų išėitys buvo išradingai koreliuojamos su signalais (pvz.: tik amplitudės, Spearman arba Pearson koreliacijos būdais ir pan.). Įprastai charakteristikos išėitis tiesiogiai nesikoreliuoja su signalu, be to ne visoms charakteristikoms koreliacinis palyginimas buvo veiksmingiausias,- šiais atvejais duomenis teko interpretuoti.

Pastebėta, jog mažai pavienių garso charakteristikų išgavimo algoritmų padeda apibrėžti bendrą akustinio triukšmo situaciją. Siekiant filtruoti akustinį triukšmą būtina remtis amplitudžių manipuliacijomis, kurios reikšmingai koreliuoja su norimais garsais kol $SNR > 1$. Ištyrus testuojamus įrašus nustatyta, jog akustiniu triukšmu užterštas garsas nėra mažesnio nei 2 SNR normaliomis sąlygomis. Galima teigti, jog amplitudžių manipuliacijomis paremtų charakteristikų turi užtekti norint išspręsti akustinio triukšmo problemą bendru atveju. Pabrėžtina, jog papildomas apdorojimas gali padėti atpažinti norimus garsus ir su mažesniu SNR nei 1, jeigu akustinis triukšmas yra pakankamai pastovus.

Išskirtos šios garso charakteristikos tiksliausiai apibūdinančios akustinį triukšmą arba norimus garsus (žr. 2 lentelę, 53 pav.):

- Dominuojantys dažniai;
- Dažnių energija;
- Spektro geometrinis tiesiškumas;
- Dažnių triukšmingumas;
- Mažos jėgos spektro kiekis;
- Vidutinė kvadratinė vertė;
- Spektrų atstumas.

Charakteristikų apžvalga:

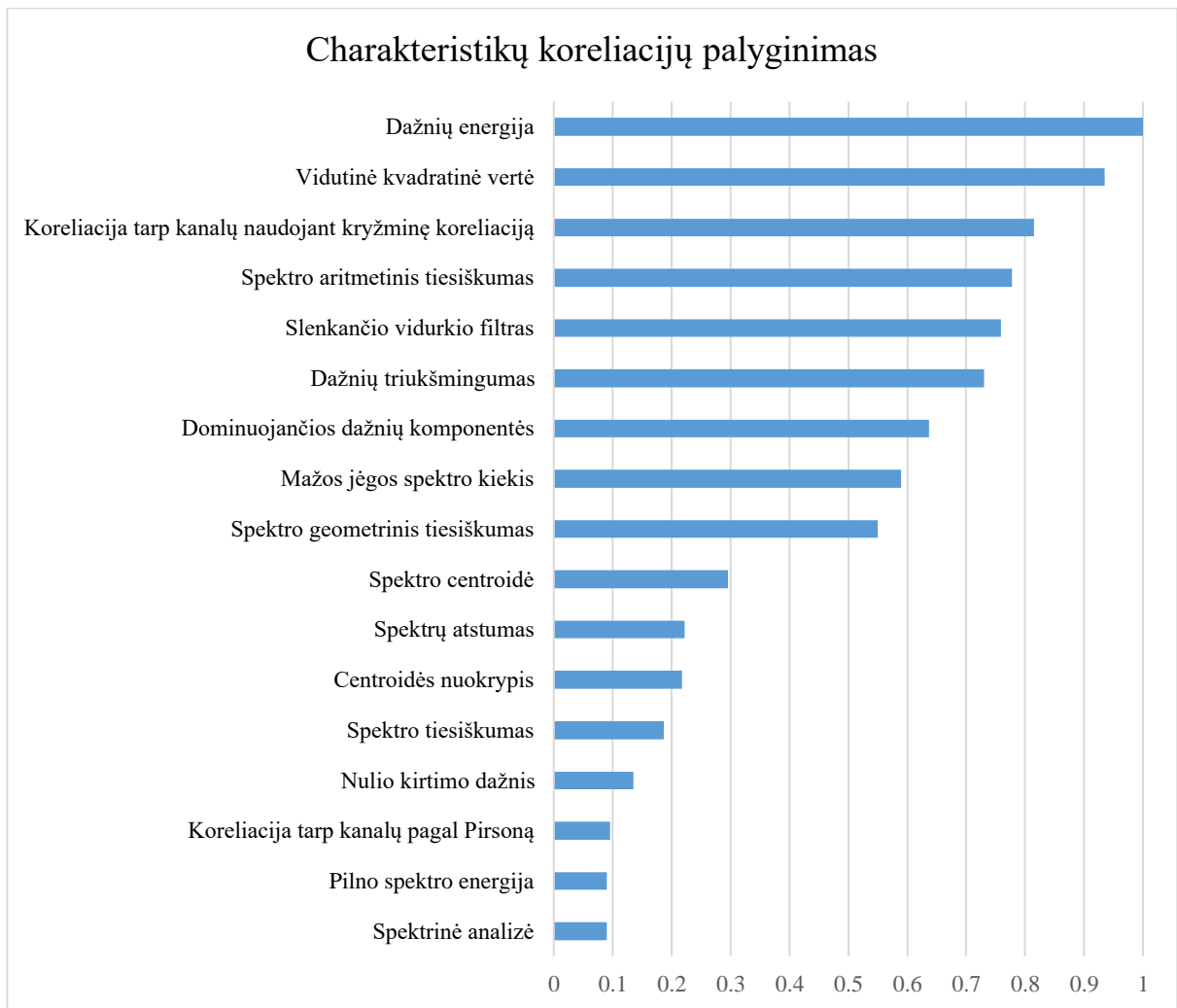
- Dominuojantys dažniai - tiriant pastebėta, jog garso energija reikšmingai padidėja pasireiškus reikšmingiems garsams. Galima daryti prielaidą jog net ir plataus spektro triukšmas neperneša tiek daug energijos. Dėl šių charakteristikų garso energija yra viena tinkamiausių charakteristikų tirti kitas charakteristikas, lyginant kitų charakteristikų koreliaciją su švaraus garso energija;
- Dažnių energija – pastebėta, jog suradus dominuojančių dažnių amplitudžių kiekį ir dominuojančias amplitudes filtruojant pagal minimalų amplitudės padidėjimą (50%) gaunama labai koreliatyvi išėitis neužterštajam signalui. Ši charakteristika geba atskirti žmogui įprastą akustinį triukšmą;
- Dažnių triukšmingumas - tiriant pastebėta, jog charakteristikos išėitis stipriai koreliuoja su norimais garsais, bei atitinkamai - mažiau su triukšmu, daugelyje testavimo atvejų. Ši charakteristika gali būti naudojama išskirti norimus garsus, arba aptikti dominuojančius reikšmingų garsų dažnius. Darant prielaidą, jog esant $SNR > 1$ dominuojantys dažniai daugelyje atvejų būtų norimieji garsai, atvirkščiai,- kai $SNR < 1$. Tačiau triukšmo bei norimų garsų spektrai gali būti įvairiai pasiskirstę ir tai apskaičiuoti yra sunku.
- Spektro geometrinis tiesiškumas - tyrimų metu pastebėta, jog charakteristikos išėitis stipriai koreliuoja su reikšmingais garsais ir pasižymi mažesne koreliacija triukšmui. Stebint pateiktus charakteristikos rezultatų grafikus galima teigti, jog charakteristika geba atskirti akustinį triukšmą nuo reikšmingų garsų. Panašu, jog charakteristika pateikia gerus rezultatus esant plataus spektro triukšmui;
- Mažos jėgos spektro kiekis - tyrimų metu pastebėta, jog ši charakteristika reikšmingai koreliuoja su įrašo triukšminga dalimi;
- Vidutinė kvadratinė vertė - tyrimų metu pastebėta, jog charakteristikos išėitis stipriai koreliuoja su originaliu garso įrašu, tačiau pasižymi didesne koreliacija triukšmui.

Charakteristika geriau koreliuoja su triukšmu nei originaliu garso įrašu, pasitelkus papildomą apdorojimą, toks koreliacijos skirtumas, leistų atskirti triukšmingą garsą;

- Spektrų atstumas – Pastebėta jog charakteristikos išvestis vidutiniškai nekoreliuoja nei su triukšmu nei su originaliu garsu. Tačiau, išanalizavus grafikus pastebėta, jog tam tikrose situacijose ji stipriai neigiamai koreliuoja su triukšmo garsu. Galima teigti, jog charakteristiką įmanoma pritaikyti tam tikrose situacijose. Charakteristika pasirinkta, nes ji bus iširta kartu su S. F. Boll [36] spektrų atimties algoritmu (jame ji naudojama).

2 lentelė. Visų charakteristikų bendrų koreliacijos nuokrypių palyginimas

Pavadinimas	Bendras koreliacijų nuokrypis (filtruoto garso amplitudės koreliacijos tarp triukšmo bei originalo)	Koreliacijos tipas lyginant charakteristikos išvesties duomenis su įvesties duomenimis.
Dažnių triukšmingumas	0.7305	Spearman
Mažos jėgos spektro kiekis	-0.5900	Pearson
Koreliacija tarp kanalų naudojant kryžminę koreliaciją	0.8150	Pearson
Koreliacija tarp kanalų pagal Pirsoną	-0.0951	Pearson
Dažnių energija	1	Pearson
Nulio kirtimo dažnis	-0.1347	Pearson
Spektrinė analizė	0.0896	Pearson
Pilno spektro energija	0.0896	Pearson
Spekto centroidė	0.296	Pearson
Centroidės nuokrypis	0.2174	Pearson
Dominuojančios dažnių komponentės	0.6367	Pearson
Spekto tiesiškumas	0.1872	Pearson
Spekto geometrinis tiesiškumas	0.5498	Pearson
Spekto aritmetinis tiesiškumas	0.7775	Pearson
Vidutinė kvadratinė vertė	0.9354	Pearson
Slenkančio vidurkio filtras	0.7593	Pearson
Spektrų atstumas	-0.2224	Pearson



53 pav. charakteristikų koreliacijų palyginimas

3.2. Akustinio triukšmo charakteristikų kombinacijų tyrimas

Kombinuojamos reikšmingiausios akustinio triukšmo charakteristikos, siekiant pasiūlyti veiksmingiausias charakteristikų kombinacijas, kuriomis remiantis akustinio triukšmo filtravimo algoritmas gebėtų veiksmingai filtruoti garsus.

Norint sukurti algoritmą gebantį identifikuoti akustinio triukšmo spektrą ir jį atskirti nuo reikšmingų garsų - reikalingos charakteristikos identifikuojančios nereikšmingus bei reikšmingus garsus. Būtina išrinkti bei apjungti charakteristikų išvestis, jas gali tekti papildomai filtruoti, nes kiekviena charakteristika turi limitų bei trūkumų.

Išskirtos šios charakteristikos išskiriančios reikšmingiausią informaciją, bei papildančios viena kitą:

- Dominuojantys dažniai
- Dažnių energija
- Spektro geometrinis tiesiškumas
- Dažnių triukšmingumas
- Mažos jėgos spektro kiekis
- Vidutinė kvadratinė vertė
- Spektro atstumas

Taip pat naudojamas ir slenkančių vidurkių filtras.

Dirbama su charakteristikomis, kurios manipuliuoja dažnių amplitudėmis, tad logiška, jog filtravimas irgi vyktų dažnių spektro atžvilgiu.

Algoritmo sprendimų tikslumui matuoti, greitaveikos bei problemos analizės matmenų suspaudimo tikslais dažnių spektras turėtų būti suskaidomas į 10 dalių ir kiekviena dalis atitinkamai filtruojama.

Apjungtos charakteristikos toliau vadinamos - charakteristikomis.

3.2.1. Garso charakteristikų išskyrimo algoritmų pritaikymas

Tiriant charakteristikas, jų apjungimą ištirta bendra amplitudžių koreliacija pilname dažnių spektre. Akustinio garso filtravimo problema prideda dar vieną filtravimo dimensiją t.y. patį dažnių spektrą. Atpažįstant triukšmą bei norimus girdėti garsus neužtenka pritildyti arba pagarsinti visus garsus, kadangi tokiu atveju būtų įleista pernelyg daug akustinio triukšmo visame dažnių diapazone. Dažnių spektrą galima suskaidyti į pasirinktą dažnių kiekį ir šias dalis filtruoti atskirai. Tokiu būdu išrenkamos dažnių dalys pernešančios reikšmingiausią informaciją.

Eksperimentų būdu pasirinkta dažnių spektrą skaidyti į 10 dalių (MATLAB kodas):

$frequencies = [20\ 70\ 160\ 350\ 600\ 900\ 1800\ 3500\ 6000\ 10500]$; %in Hz

Suskaidyto dažnio spektro manipuliacijai pasirinkta naudoti MATLAB įrankį - parametrinį ekvalaizerį gebantį dirbti su 10 dažnių juostų.

Sukadravus informaciją, kadru etapuose panaudojamos charakteristikos, kurių išvados aprašo kaip turėtų būti filtruojami išrinktieji dažniai. Tai yra bene paprasčiausias būdas įrodyti algoritmo veiksmingumą, nenaudojant jokių kitų algoritmų dalių.

3.2.1.1. Duomenų rinkimas

Kiekvieno kadro metu atnaujinami šie pagrindiniai duomenys. Visi duomenys atnaujinami dvejiems (angl. *stereo*) kanalams atskirai, todėl kanalų kiekis nebus įkomponuotas į formules.

Formulių aprašymuose minimas sutrumpinimas „*param*“, reiškia jog tai yra išsaugomas parametras ir perduodamas sekančiam kadru, be limitavimų jį keisti ir naudoti esamo, sekančio kadro eigoje.

Naudojamos reikšmės:

- EqF – pasirinktų 10 ties dažnių indeksas.
- k yra viso dažnių spektro indeksas, arba atitinkamai amplitudžių spektro indeksas.
- $Param.x$ – parametras, kuris atnaujinamas ir dalinamas tarp kadru
- $targetDbToLower$ – norimas decibelų kiekis, kiek reiktų mažinti akustinį triukšmą, kuris nustatytas 85dB.

3.2.1.2. Kadro GFT reikšmės

GFT raiška kadre priklauso nuo kadro dydžio (laiko tarpo paskirto kadru). GFT raiška, bei atitinkami GFT dažnių atitikmenys (sumažintoje raiškoje) skaičiuojami pagal formules:

$$frequencyGap = sampleRate/frameSize/2 \quad (3.23)$$

$$midFrameFreq(EqF) = round(frequencies/frequencyGap) \quad (3.24)$$

GFT duomenys skaičiuojami pagal formules, kur $a(k)$ yra k dažnio amplitudė a dažnių amplitudžių juostoje:

$$fftCurrent(k) = |a(k)| \quad (3.25)$$

$$L = param.previousLowestFftAvgCnt \quad (3.26)$$

Suskaičiuojami senesni GFT duomenys, juos atnaujinant pagal L greitį:

$$param.previousFft(k) = \frac{param.previousFft(k)*(L-1)+fftCurrent(k)}{L} \quad (3.27)$$

$$t1(k) = \frac{param.previousLowestFft(k)*(L-1)+fftCurrent(k)}{L} \quad (3.28)$$

Suskaičiuojami GFT duomenys, juos atnaujinant pagal L greitį arba dydį, jeigu naujausi duomenys yra mažesni:

$$param.previousLowestFft(k) = \min\{fftCurrent(k), t1(k)\} \quad (3.29)$$

Išvedami praeitų GFT skaičiavimų vidurkiai dažnių skirsniams (EqF):

$$previousLowestFftPerFreq(EqF) = 1/ \sum_{i=(midFrameFreq(EqF-1)+midFrameFreq(EqF))/2}^{(midFrameFreq(EqF+1)+midFrameFreq(EqF))/2} param.previousLowestFft(n) \quad (3.30)$$

$$energyLevelPerFreq(EqF) = 1/ \sum_{i=(midFrameFreq(EqF-1)+midFrameFreq(EqF))/2}^{(midFrameFreq(EqF+1)+midFrameFreq(EqF))/2} param.previousFft(n) \quad (3.31)$$

$$fftShortCnt = param.fftShortAvgCnt \quad (3.32)$$

$$fftLongCnt = param.fftLongAvgCnt \quad (3.33)$$

$$fftCurrentMed(k) = MED(fftCurrent) \quad (3.34)$$

$$B(k) = \frac{param.fftShort(k)*(fftShortCnt - 1) + fftCurrentMed(k)}{fftShortCnt} \quad (3.35)$$

$$param.fftShort(k) = \max\{fftCurrentMed(k), B(k)\} \quad (3.36)$$

$$param.fftLong(k) = \frac{param.fftLong(k)*(fftLongCnt - 1) + fftCurrentMed(k)}{fftLongCnt} \quad (3.37)$$

$$fftPerFreqShort(EqF) = \sum_{i=(midFrameFreq(EqF-1)+midFrameFreq(EqF))/2}^{(midFrameFreq(EqF+1)+midFrameFreq(EqF))/2} RMS(param.fftShort(n)) \quad (3.38)$$

$$fftPerFreqLong(EqF) = \sum_{i=(midFrameFreq(EqF-1)+midFrameFreq(EqF))/2}^{(midFrameFreq(EqF+1)+midFrameFreq(EqF))/2} RMS(param.fftLong(n)) \quad (3.39)$$

$$diffProfilePerFreq(EqF) = |fftPerFreqLong(EqF) - fftPerFreqShort(EqF)| \quad (3.40)$$

3.2.1.3. Dažnių triukšmingumas

Dažnių triukšmingumo savybę aprašo *currDiffProfile* (3.41), kur $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a , $d(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje d .

$$currDiffProfile(k) = \frac{a(k)}{d(k)} \quad (3.41)$$

$$param.diffProfile(k) = \frac{param.diffProfile(k)*(param.diffProfileAvgCnt - 1) + currDiffProfile(k)}{param.diffProfileAvgCnt} \quad (3.42)$$

$$t1(k) = \frac{param.diffProfileMax(k)*(param.diffProfileMaxAvgCnt-1)+diffProfileMax(k)}{param.diffProfileMaxAvgCnt} \quad (3.43)$$

$$param.diffProfileMax(k) = \max\{currDiffProfile(k), t1(k)\} \quad (3.44)$$

$$diffProfilePerFreq(EqF) = 1/n \sum_{i=(midFrameFreq(EqF-1)+midFrameFreq(EqF))/2}^{(midFrameFreq(EqF+1)+midFrameFreq(EqF))/2} param.diffProfile(n) \quad (3.45)$$

$$diffProfilePerFreqMax(EqF) = 1/n \sum_{i=(midFrameFreq(EqF-1)+midFrameFreq(EqF))/2}^{(midFrameFreq(EqF+1)+midFrameFreq(EqF))/2} param.diffProfileMax(n) \quad (3.46)$$

3.2.1.4. Koreliacija tarp kanalų naudojant kryžminę koreliaciją

Koreliacijos tarp kanalų duomenys išrenkami pagal formulę $cor(n)$.

$$cor(n) = xcorr(param.diffProfileMax(n)_{kairys}, param.diffProfileMax(n)_{desinys}) \quad (3.47)$$

$$diffProfilePerFreqMax(EqF) = 1/n \sum_{i=(midFrameFreq(EqF-1)+midFrameFreq(EqF))/2}^{(midFrameFreq(EqF+1)+midFrameFreq(EqF))/2} cor(n) \quad (3.48)$$

3.2.1. Charakteristikų kombinavimo bandymai

Kombinuojamos pasirinktos charakteristikos ir tiriamas kombinacijų veiksmingumas. Dalis charakteristikų tiriama pasitelkiant jų išvedimą ir koreliuojant su garso amplitude, likusieji tyrimai atliekami filtruojant garsus ir koreliuojat su garsais.

3.2.1.1. Vidutinės aukštos energijos dažniai

Apjungiamos Dominuojančių dažnių, dažnių energijos, spektro aritmetinio tiesiškumo charakteristikos.

Vidutinės aukštos energijos dažnių spektras $VME(k)$ kiekvienam dažniui k apskaičiuojamas pagal formulę (3.60).

Pirma būtina apskaičiuoti palaipsniui kintančius svetus $LIM(k)$, $LMT(k)$ kur $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a, $avgCnt = 13$:

$$LIMv(k) = \frac{LIM(k)*(1000-1)+a(k)}{1000} \quad (3.49)$$

$$LIM(k) = \max\{a(k), LIMv(k)\} \quad (3.50)$$

$$LIMT(k) = \frac{LIMT(k)*(3-1)+a(k)}{3} \quad (3.51)$$

$$LIMmax = \max(LIM)_1 \quad (3.52)$$

$$LIMTmax = \max(LIMT)_2 \quad (3.53)$$

$$EFI = a(k) * LIMmax/0.75 \quad (3.54)$$

$$EFC = a(k) * LIMTmax/0.75 \quad (3.55)$$

$$EFI(k) = \max(EFI(k) - 1, 0) \quad (3.56)$$

$$EFC(k) = \max(EFC(k) - 1, 0) \quad (3.57)$$

$$VME_m(k) = \frac{VME(k)*(avgCnt-5)+EFI(k)*5}{avgCnt} \quad (3.58)$$

$$VME_a(k) = \frac{VME(k)*(avgCnt-1)+EFI(k)}{avgCnt} \quad (3.59)$$

$$VME(k) = \min\{VME_m(k), VME_a(k)\} \quad (3.60)$$

Charakteristikos atsakymas išgaunamas pasveriant $VME(k)$ ir suskaičiuojant kiekį C , taip jog C yra lygus visiems VME elementams, kurie yra > 0 , kur n yra lygus dažnių juostų kiekiui. SGT – spektro geometrinis tiesiškumas.

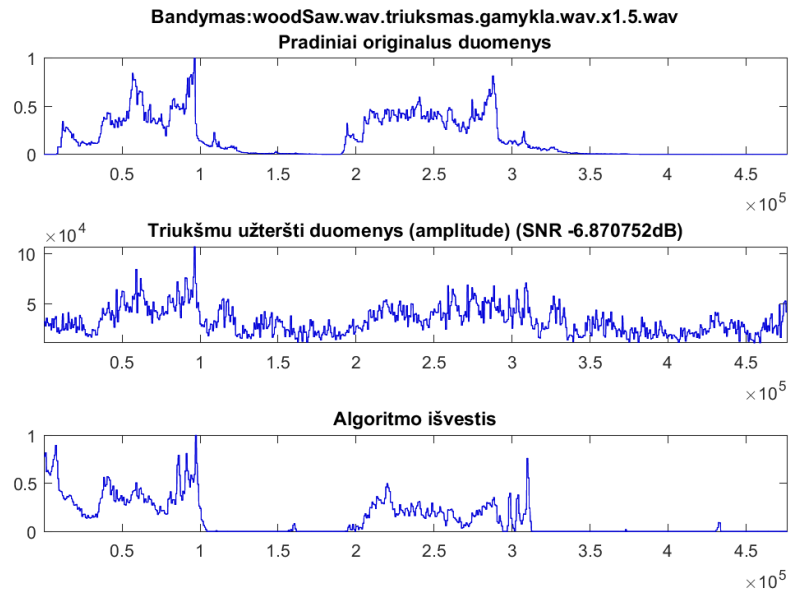
$$SGT = (\prod_{k \in b} a(k))^{\frac{1}{k}} \quad (3.61)$$

$$C = \sum_{k=1}^n [VME(k) > 0] * SGT \quad (3.62)$$

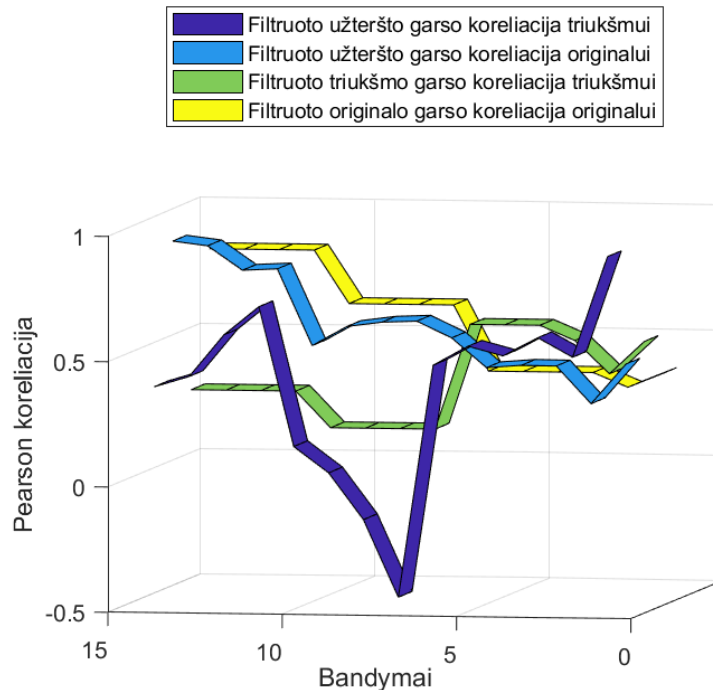
Gaunami atsakymai rodo žymią koreliaciją su užterštu bei originaliu garsu, tačiau ši koreliacija kinta bandymų metu. Neskaitant nepastovios koreliacijos, ši charakteristika geba išskirti žymios aukštos energijos dažnius iš triukšmo spektro ir daugelyje atvejų gražina norimą rezultatą. Koreliacijos matavimui, dažniai virš ribos buvo sumuojami, tačiau algoritme gali būti išrenkami tik žymiausi dažniai ir taip padėtų identifikuoti $SNR > 1$ informacijos dažnius (žr. 54 pav., 55 pav.).

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – 61.0%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 38.2%;
- STI rodiklis – netaikoma, koreliuota amplitudės išvestis;
- STOI rodiklis – netaikoma, koreliuota amplitudės išvestis.



54 pav. Vidutinės aukštos energijos dažnių charakteristikos išėities pavyzdys



55 pav. Vidutinės aukštos energijos dažnių charakteristikos koreliacijos bandymų grafikas

3.2.1.2. Limituotas maksimalus vidutinis amplitudės triukšmo limitas

Limituotas maksimalus vidutinis amplitudės pokyčio proporcijos skirtumas gaunamas panaudojant dažnių triukšmingumo charakteristiką, kuri naudoja ir dažnių energijos charakteristiką, $LMVAPP(k)$ kiekvienai amplitudei k dažnių juostoje apskaičiuojamas pagal formulę (3.68).

Amplitudžių pokyčio proporcija $P(k)$ kiekvienai amplitudei k dažnių juostoje apskaičiuojama pagal formulę (3.63), kur $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a , $d(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje d , kur $avgCnt = 20$:

$$P(k) = \frac{a(k)}{d(k)} \quad (3.63)$$

$$B(k) = \frac{LMVAPP(k) * (avgCnt - 1) + P(k)}{avgCnt} \quad (3.64)$$

$$LMVAPP(k) = \max\{P(k), B(k)\} \quad (3.65)$$

Charakteristikos atsakymas išgaunamas pasveriant $LMVAPP(k)$ ir suskaičiuojant kiekį C , taip jog C yra lygus visiems $LMVAPP$ elementams, kurie yra > 3 , kur n yra lygus dažnių juostų kiekiui.

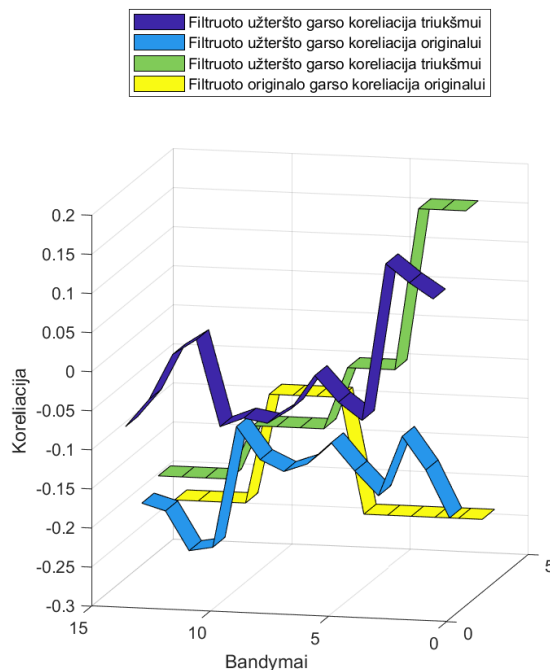
$$C = \sum_{k=1}^n [LMVAPP(k) > 3] \quad (3.66)$$

Gaunami atsakymai rodo nežymią koreliaciją su neužterštu garsu ir mažesnę koreliaciją su užterštais garsais (žr. 56 pav.).

$LMVAPP(EqF)$ kiekvienai EqF dažnio amplitudei apskaičiuojamas pagal formulę (3.68), kur $avgCnt = 20$.

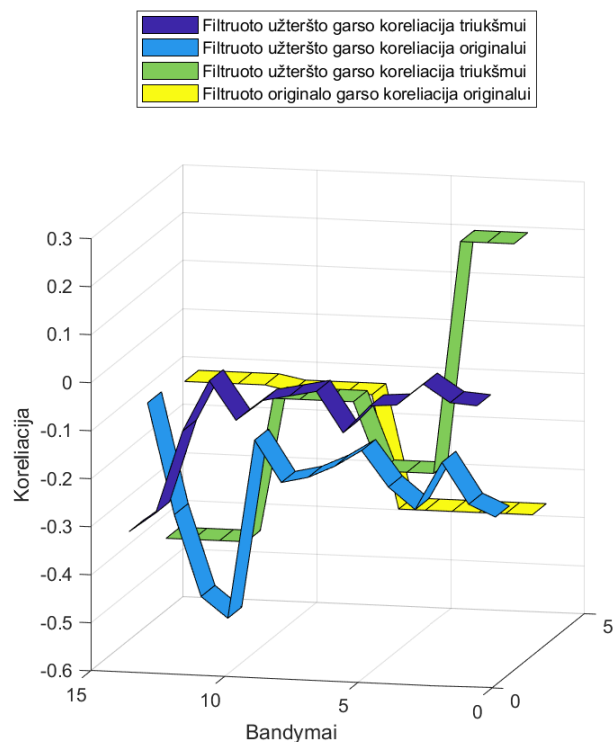
$$B(EqF) = \frac{LMVAPP(EqF) * (avgCnt - 1) + diffProfilePerFreq(EqF)}{avgCnt} \quad (3.67)$$

$$LMVAPP(EqF) = \max\{diffProfilePerFreq(EqF), B(EqF)\} \quad (3.68)$$



56 pav. LMVAPP koreliacija garso įrašams

Pastebėta, jog pritaikius medianos filtrą $P(k)$ gaunami pastovesni ir tikslesni rezultatai (žr. 57 pav.).



57 pav. LMVAPP koreliacija garso įrašams pritaikius medianos filtrą

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – -31.7%;
- Koreliacija triukšmo signalui (santykinė pradinei) – -40.5%;
- STI rodiklis – netaikoma, koreliuota amplitudės išvestis;
- STOI rodiklis – netaikoma, koreliuota amplitudės išvestis.

3.2.1.3. Minimalios energijos nustatymas ir filtravimas

Minimalios energijos filtravimas atliekamas pagal formulę (3.74), kur $minimumEnergyAllowed = 1$:

$$ME(EqF) = energyLevelPerFreq(EqF)/minimumEnergyAllowed \quad (3.69)$$

$$ME(EqF) = 0, \text{ kai } ME(EqF) > 1 \quad (3.70)$$

$$ME(EqF) = |ME(EqF) - 1| * targetDbToLower, \text{ kai } ME(EqF) > 0 \quad (3.71)$$

$$EML = param.enforceMinimumEnergy \quad (3.72)$$

$$EML_a(EqF) = \frac{EML(EqF)*(minimumEnergyAllowed - 1) + ME(EqF)}{minimumEnergyAllowed} \quad (3.73)$$

$$EML(EqF) = \min\{enforceMinimumEnergy(EqF), EML_a(EqF)\} \quad (3.74)$$

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – 94.3%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 91.6%;
- STI rodiklis – 0.136;
- STOI rodiklis – 0.257.

3.2.1.4. Koreliacijos tarp kanalų charakteristikos įkomponavimas

Tiriant pastebėta jog koreliacijos tarp kanalų charakteristikos įkomponavimas neduoda norimų rezultatų, įkomponavus koreliaciją tarp kanalų kaip svertą daliai sprendimų, gerokai sumažėja išfiltruoto triukšmo kiekis ir nukenčia algoritmo veiksnumas. Pvz.: filtravimo veiksnumas (išfiltruoto triukšmo kiekio prieaugio skirtumas išfiltruotam originaliam garsui) su koreliacija tarp kanalų yra 0.1851, be koreliacijos tarp kanalų 0.2190, kombinuojant kartu su vidutinės aukštos energijos, LMVAPP algoritmais.

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – 99.4%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 93.8%;
- STI rodiklis – 0.141;
- STOI rodiklis – 0.259.

3.2.1.5. Monotoniškų spektrų filtras

Monotoniškų spektrų filtro (žr. 3 figūrą) tikslas - identifikuoti monotoniško spektro dažnius pasirinktame laiko tarpe. Filtras negeba identifikuoti reikšmingų garsų, nors pagal dizainą reikšmingi garsai visados išjungia filtravimą. Aptikus reikšmingus garsus pagal nustatymus filtravimas nutraukiamas tam tikram laiko tarpui. Siekiant tinkamai identifikuoti monotoniško triukšmo garsus - optimalus nutraukimo laiko tarpas nustatomas ilgas. Dėl to būtina šį filtravimo būdą kombinuoti su kitais filtrais.

Monotoniški spektrai surandami pagal formulę MS (3.79), kur $a(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje a , $d(k)$ yra dažnio k amplitudė dažnių amplitudžių juostoje d .

$$MS = \text{fftPerFreqLong}(EqF) / \text{fftPerFreqShort}(EqF) , \text{ visiems } EqF \quad (3.75)$$

$$MS = |MS(EqF) - 1|, \text{ visiems } EqF \quad (3.76)$$

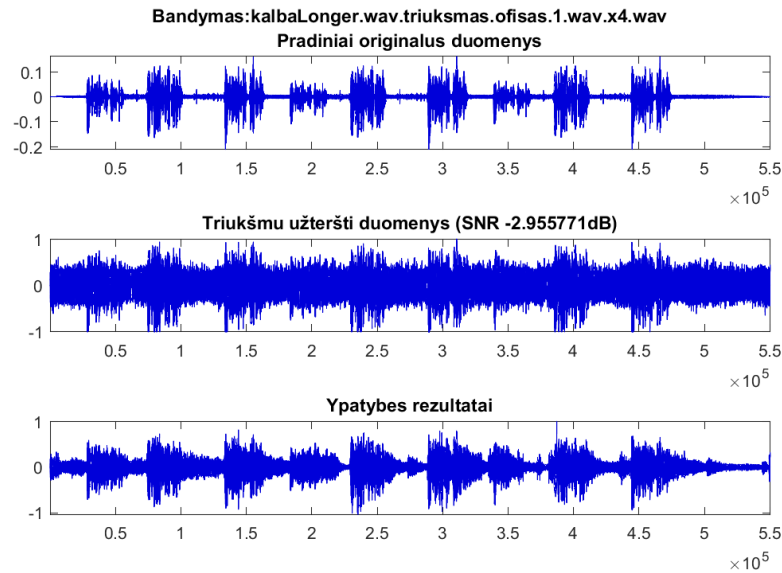
$$\text{minOfMon} = \min\{0.4, \max(\min(MS) + 0.1)\} \quad (3.77)$$

$$MS = \begin{cases} MS = 0 & , \text{ kur } MS > \text{minOfMon} \\ MS = MS, & \text{ kur } MS \leq \text{minOfMon} \end{cases} \quad (3.78)$$

$$MS = MS(EqF) * \text{targetDbToLower} * 2.2, \text{ visiems } EqF \quad (3.79)$$

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus (žr. 58 pav.):

- Koreliacija originaliam signalui (santykinė pradinei) – 100%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 70.2%;
- STI rodiklis – 0.168;
- STOI rodiklis – 0.263.



58 pav. Monotoniškų spektrų filtro išėities pavyzdys

3.2.1.6. Adaptyvaus limito, maksimalios jėgos filtras

Adaptyvaus limito, maksimalios jėgos filtro tikslas - identifikuoti reikšmingą informaciją ir palaikyti sprendimų vientisumą išfiltruojant visus kitus dažnius (žr. 4 figūrą). Filtras neatskiria monotoniškų spektrų (sąlyginai nekintančio triukšmo) nuo reikšmingos informacijos. Pagal dizainą, esant tik akustiniam triukšmui, filtro išvestis artėja link sprendimo, jog visi garsai yra reikšmingi. Tyrimų metu, nustatyta jog toks dizainas yra būtinas, norint lanksčiai aptikti įvairius reikšmingus garsus. Dėl šio trūkumo šį filtravimo būdą būtina kombinuoti su kitais.

Šis filtras yra vidutinės aukštos energijos filtro patobulina versija. Filtras veikia naudodamasis adaptyviojo limito pagalba išfiltruojant didžiausios laikinos jėgos dažnių spektrus.

$$dat = \max(fftPerFreqLong) - fftPerFreqShort \quad (3.80)$$

$$dat = 1, \text{ kur } dat < 1 \quad (3.81)$$

$$DP = diffProfilePerFreq \quad (3.82)$$

Kintamasis *energiesPsychoacustics* aprašomas 3.3.2 skyriuje.

$$DP = DP * dat / energiesPsychoacustics \quad (3.83)$$

Apskaičiuojant medianos vidurkį būtina transponuoti $DP(EqF)$ matricą, po apskaičiavimo reikia transponuoti dar kartą.

$$limitas = \text{med}(DP^T)^T * \text{param.thresholdDownWeight} \quad (3.84)$$

$$A = \frac{\text{param.threshold}(k) * (\text{param.fftLongAvgCntShrt} - 1) + limitas}{\text{param.fftLongAvgCntShrt}} \quad (3.85)$$

$$B = \frac{\text{param.threshold}(k) * (\text{param.thresholdAvgCnt} - 1) + limitas}{\text{param.thresholdAvgCnt}} \quad (3.86)$$

$$\text{param.threshold} = \max(A, B) \quad (3.87)$$

$$DP = DP - \text{param.threshold} \quad (3.88)$$

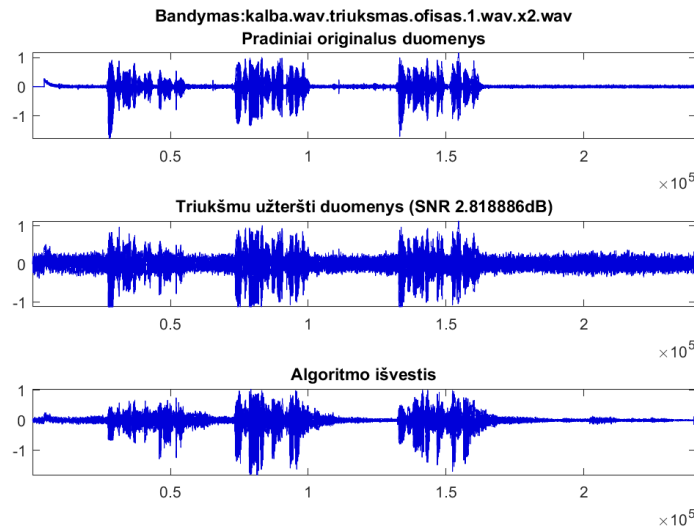
$$DP = 0, \text{ kur } DP > 0 \quad (3.89)$$

$$DP = |DP| * |DP| \quad (3.90)$$

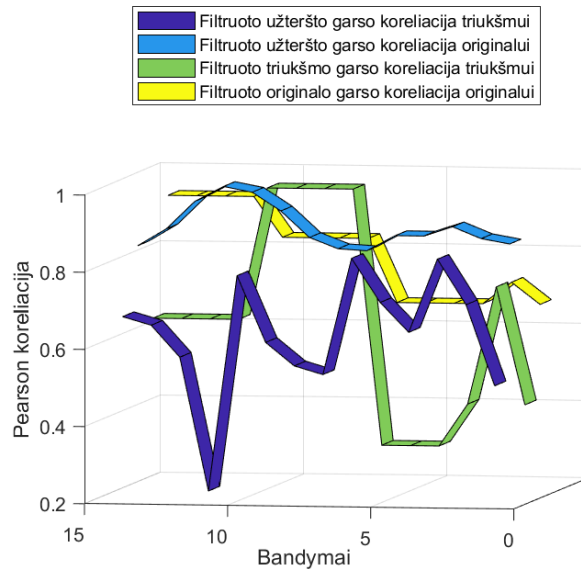
Paskutiniame žingsnyje atliekama DP normalizacija iki užsibrėžto $targetDbToLower$ filtravimo limito.

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus (žr. 60 pav., 59 pav.):

- Koreliacija originaliam signalui (santykinė pradinei) – 98.7%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 63.3%;
- STI rodiklis – 0.184;
- STOI rodiklis – 0.246.



59 pav. Adaptyvaus limito, maksimalios jėgos filtravimo pavyzdys



60 pav. Pasvertų filtravimo korelacių grafikas

3.2.1.7. S. F. Boll spektrų atimties filtras

S. F. Boll [36] pasiūlyto spektrų atimties filtro apžvalga kombinavimo su charakteristikomis tikslu [36].

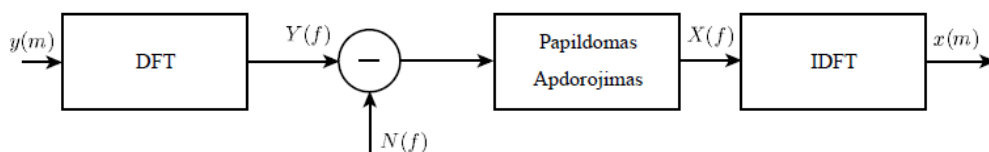
Signalas filtruojamas iš jo dažnių spektro atimant pradinio triukšmo spektrą. Pradžioje randamas triukšmo spektras, triukšmo spektrų duomenys dažniausia atnaujinami iš signalo periodų kuriuose nėra atpažinto norimo garso.

Filtro formulė (3.91), kur $|X(f)|^b$ yra originalaus signalo $|Y(f)|^b$ įvertis atimant $|N(f)|^b$ vidutinį triukšmo spektrą:

$$|X(f)|^b = |Y(f)|^b - a|N(f)|^b \quad (3.91)$$

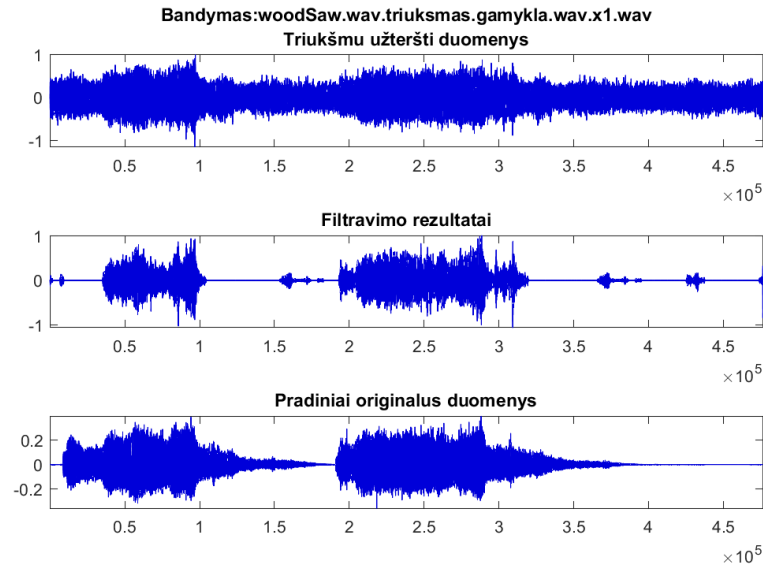
Turima omenyje jog triukšmas yra stacionarus atsitiktinis procesas. Jei vykdoma amplitudės atimtis, tai $b=1$, jei vykdoma galios spektrų atimtis tai $b=2$. Parametras a valdo atimamo triukšmo amplitudę, pilnam triukšmo atėmimui $a=1$. Laike normalizuotas triukšmo spektras gaunamas iš periodų, kai neaptinkamas norimas garsas pagal formulę (3.92), kur $|N_i(f)|$ yra i -tojo lango triukšmo spektras, K – langų skaičius.:

$$|N(f)|^b = \frac{1}{K} \sum_{i=0}^{K-1} |N_i(f)|^b \quad (3.92)$$



Spektrų atimties filtro blokinė diagrama

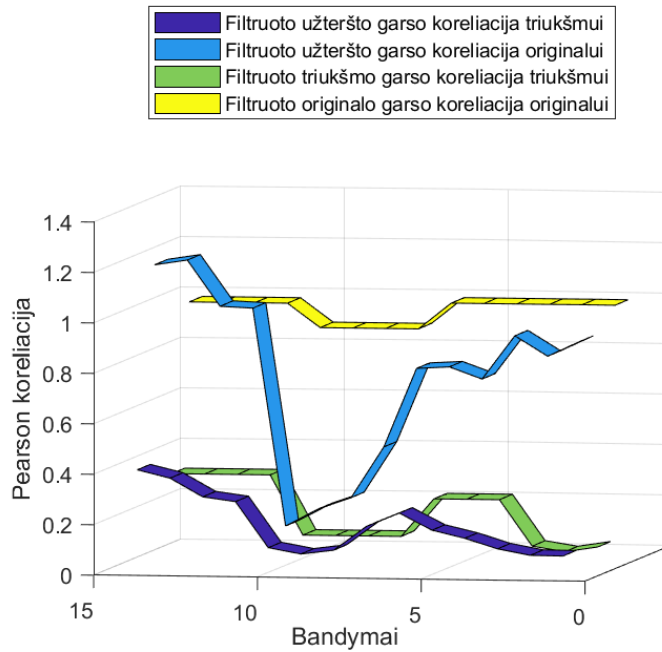
Testuojama plačiai naudojama ir tyrinėjama S. F. Boll [36] spektrų atimties teorema [36] balso įrašo švarinimo tikslais nedaug tesiskirianti nuo esamo pavyzdžio, kurią pritaikius atlikti koreliacijos testai. S. F. Boll [36] spektrų atimties metode triukšmo spektras atnaujinamas kadru informacija, kuriuose neaptikta žmonių kalba. Kalba aptinkama spektrų atstumo charakteristikos išskyrimo metodu (žr. 61 pav.).



61 pav. Spektrų atimties filtro išėities pavyzdys

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus (žr. 62 pav.):

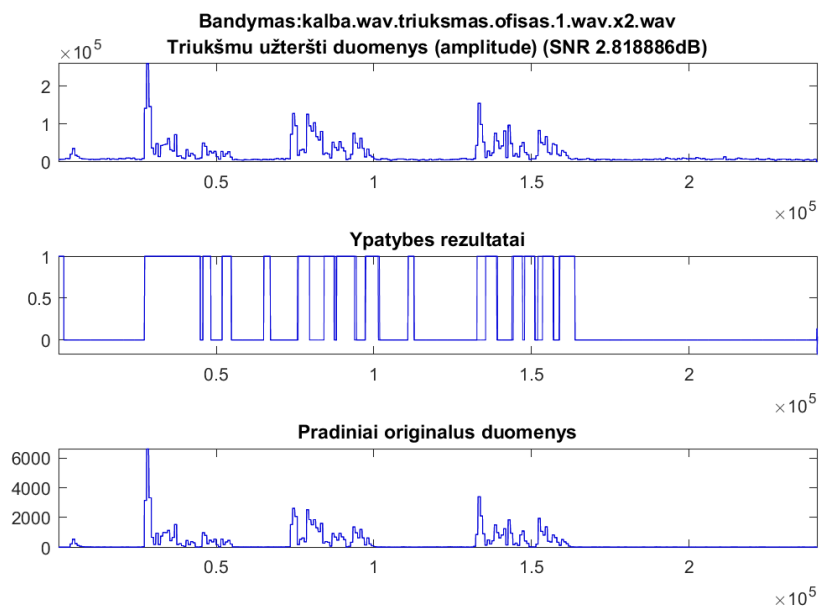
- Koreliacija originaliam signalui (santykinė pradinei) – 75.4%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 18.8%;
- STI rodiklis – 0.364;
- STOI rodiklis – 0.420.



62 pav. Spektrų atimties filtro koreliacijos grafikas

Tyrimų metu pastebėta, jog koreliacijos atžvilgiu spektrų atimties filtras geba aptikti triukšmą ir jį išfiltruoti, tačiau, nors koreliacija originaliam įrašui vidutiniškai siekia 75%, jis dažnu atveju žmogui pasidaro nebesuprantamas. Taip pat spektrų atimtis įneša papildomų (muzikinių) garsų (iškraipymų). Geriausiu atveju toks algoritmas (arba jo dalys) gali būti naudojamas triukšmui aptikti, tačiau šis algoritmas galėtų būti patobulinamas pastarųjų algoritmų pagalba (tiksliau apmokomas).

Su esamais apmokymo algoritmais galima palyginti spektrų atimties balso atpažinimo dalį (ši dalis apmoko S. F. Boll [36] pasiūlytą spektrų atimtį), kuri yra ganėtinai paprasta (žr. 63 pav.).



63 pav. S. F. Boll [36] balso atpažinimo algoritmo pavyzdys

Algoritmas yra labai ribotas net apdorojant beveik švarų įrašą su žmogaus balsu (kadangi kalbėjimo signalo mažai, triukšmo šiame signalo yra ganėtinai mažai (su SNR 2.81)). Akivaizdu, jog šis algoritmas nėra tinkamas miesto ar gamyklos triukšmui atpažinti (žr. 63 pav.).

3.2.1.8. Minimalios vidutinės kvadratinės paklaidos logaritminės-spektrinės amplitudės įvertinimo filtras

Minimalios vidutinės kvadratinės paklaidos logaritminės-spektrinės amplitudės įvertinimo - MMSE LSA [38] (*angl. MMSE LSA – Minimum mean square error with a log-spectral amplitude estimator*) algoritmas vis dar plačiai naudojamas ir studijuojamas. Algoritmas naudoja trumpo laiko GFT reikšmių įvertinimą, kur triukšmingas spektras yra padaugintas iš padidėjusios energijos, iš šių duomenų apskaičiuojant švarų signalą. Padidėjusios energijos funkcija dažnai yra BAA algoritmas, labai panašus į S. F. Boll [36] spektrų atimties siūlomą. Šis algoritmas apvežgliamas kombinavimo su charakteristikomis tikslu.

Formulėje (3.93) $Y(nT)$ aprašo triukšmingas kalbos imtis, kur T yra imties kadras, o n yra raktas. $x(nT)$ bei $d(nT)$ aprašo švarios kalbos ir pridėtinio triukšmo imtis, atitinkamai. $Y_k(m)$, $X_k(m)$ bei $D_k(m)$ aprašo k -tąjį spektro elementą, m -tajame analizės lange, atitinkamai – $y(nT)$, $x(nT)$, $d(nT)$ [38].

Kadangi švarus signalas nėra žinomas, X_k švarios kalbos komponentas apskaičiuojamas pagal MMSE LSA [38] algoritmą:

$$|X_k| = E_k / (1 + E_k) \exp\{1/2 \int_{v_k}^{\infty} e^{-t} / t * dt\} |Y_k| \quad (3.93)$$

Toliau E_k , y_k yra aprašomi formuliu:

$$E_k = l_x * (k) / l_d * (k) \quad (3.94)$$

$$y_k = |Y_k|^2 / l_d(k) \quad (3.95)$$

kur l_x bei l_d aprašo k -tuosius spektrinius komponentus, atitinkamai švarios kalbos ir triukšmą.

Kadangi švarios kalbos bei triukšmo kitimai nežinomi, jie randami pagal praeito kadro amplitudę. ML amplitudės apskaičiavimas yra paremtas signalo kitimu, randant maksimumą pagal formules:

$$l_x, ML(k) = \max\{p(Y_k(m) | l_x(k)), l_d(k)\} \quad (3.96)$$

$$E_{(k,ML)}(m) = \begin{cases} 1/L \sum_{i=0}^{L-1} y_k(m-i) - 1, & \text{kai rezultatas neneigiamas} \\ 0, & \text{kitu atveju} \end{cases} \quad (3.97)$$

Apskaičiavimai paremti L kadru iš eilės, kur

$$Y_k(m) = \{Y_k(m), Y_k(m-1), \dots, Y_k(m-L+1)\} \quad (3.98)$$

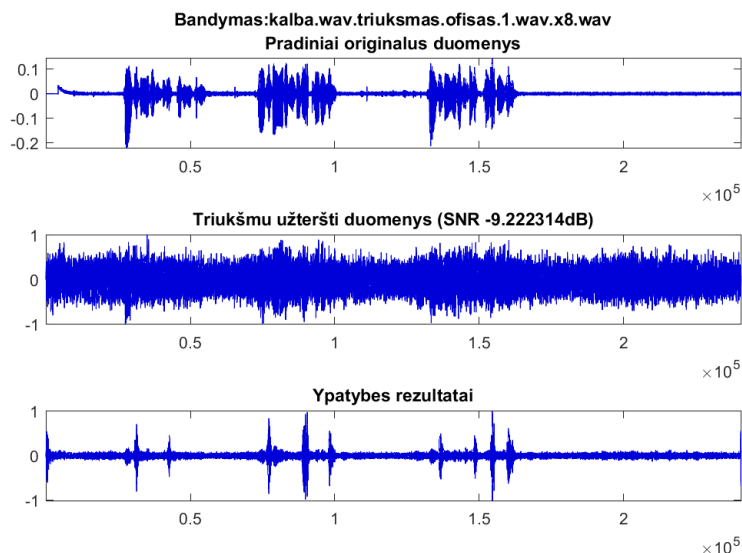
Implementacijai supaprastinti galima panaudoti rekursyviai apskaičiuojamą vidurkį, formuliu:

$$y_k(m) = ay_k(m - 1) + (1 - a) * (y_k(m)/B) \quad (3.99)$$

$$E_k(m) = P\{y_k(m) - 1\} \quad (3.100)$$

,kur a bei B atitinkamai lygūs 0.725 bei 2.

Įvertinus algoritmo veiksnumą, matyti jog jo išvestis labai panaši į S. F. Boll [36] spektrinės atimties algoritmo išvestį (žr. 64 pav.).



64 pav. MMSE LSA [38] algoritmo išvesties pavyzdys

Tyrimų metu pastebėta, jog algoritmas nesusidoroja su triukšmingais duomenimis, nes naudoja panašų BAA algoritmą į S. F. Boll [36] spektrų atimties (žr. 64 pav.). Pastebėta, jog algoritmas įterpia mažiau muzikinio triukšmo nei S. F. Boll [36] spektrų atimties algoritmas. Bet kuriuo atveju šis algoritmas gali būti panaudojamas tik patobulinant (pakeičiant) BAA atpažinimą.

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – 98.2%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 43%;
- STI rodiklis – 0.326;
- STOI rodiklis – 0.253.

3.2.2. Akustinio triukšmo charakteristikų kombinacijų apžvalga

Pateikta apjungtų charakteristikų statistika matuojant kiekvienos charakteristikos koreliacijas atskirai. Ši statistika neatspindi galutinio filtravimo rezultato ir yra sudaryta palyginimo bei papildomos informacijos tikslais. Kombinuotas charakteristikas sunku sukoreliuoti dėl tų pačių priežasčių, dėl kurių sunku sukoreliuoti pavienes charakteristikas (charakteristikos išeities duomenų skirtumai, lyginant su signalo įvestimi). Koreliuojama su pilna dažnių juosta o ne 10-imi (suskaitytu

dažnio juostos dalių), kurias apdorojus algoritmas daro atskirus sprendimus. Dėl šių priežasčių iš anksto sunku numatyti ar charakteristikų kombinacija bus veiksminga ar ne (žr. 3 lentelę, 65 pav.).

Pastebėta, jog koreliacijos atžvilgiu spektrų atimties filtras geba gerai išfiltruoti triukšmą, tačiau su įprastu balso aptikimo algoritmu visiškai sugadina girdimus balsus mažesnio nei 10 SNR įrašuose, o mašinų ar gamyklos garsų neaptinka. Taip pat spektrų atimtis įneša papildomų (muzikinių) garsų. Toks algoritmas (arba jo dalys) gali būti naudojamas triukšmui aptikti, arba jis turi būti patobulintas. MMSE LSA [38] algoritmo išeitis labai panaši į S. F. Boll [36] pasiūlytą spektrinę atimtį.

LMVAP filtro išvedami atsakymai rodo nežymią koreliaciją su neužterštu garsu ir mažesnę koreliaciją su užterštais garsais.

Vidutinės aukštos energijos dažnių filtro gaunami atsakymai rodo žymią koreliaciją su užterštu bei originaliu garsu, tačiau ši koreliacija kinta bandymų metu. Neskaitant nepastovios koreliacijos, ši charakteristika geba išskirti žymios aukštos energijos dažnius iš triukšmo spektro ir daugelyje atvejų gražina norimą rezultatą. Koreliacijos matavimui, dažniai virš ribos buvo sumuojami, tačiau algoritme gali būti išrenkami tik žymiausi dažniai ir taip padėtų identifikuoti $SNR > 1$ informacijos dažnius.

Minimalios energijos nustatymas ir filtravimas veikia ofiso, gamyklos aplinkose, tačiau pateiktas algoritmas neautomatizuoja minimalios energijos ir ją turėtų įvesti vartotojas arba pradiniai nustatymai, todėl šis nustatymas sukombinuotas į adaptyvaus limito maksimalios jėgos filtrą.

Koreliacija tarp kanalų neparodė reikšmingų rezultatų.

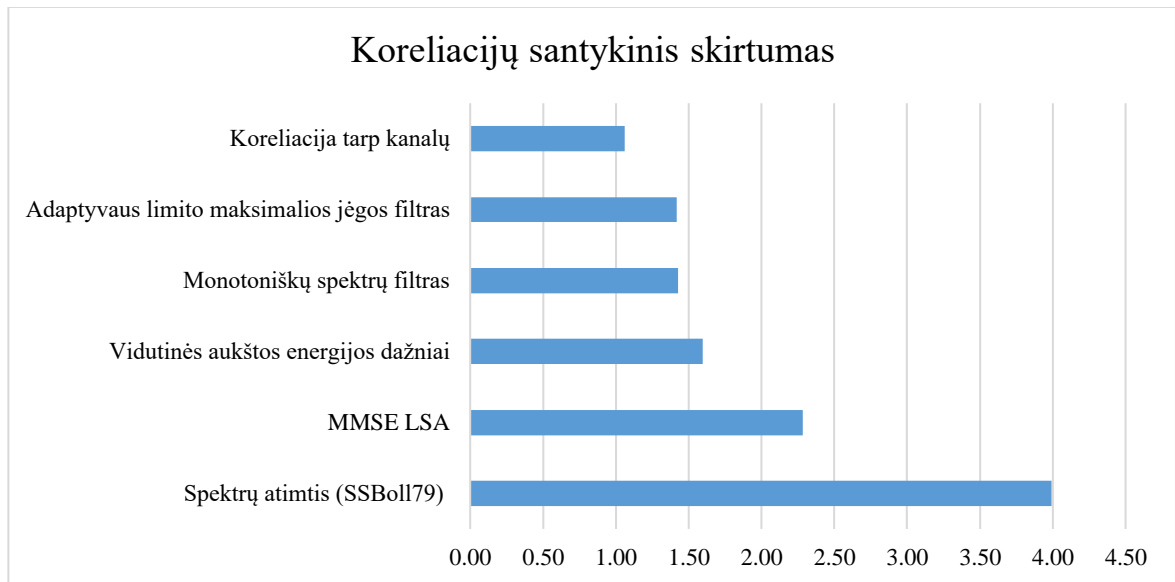
Monotoniškų spektrų filtras yra praeitų bandymų rezultatas ir pasitelkiant ganėtinai paprastus skaičiavimus geba aptikti dažnių spektrus, kurie yra monotoniškai triukšmingi.

Adaptyvaus limito maksimalios jėgos filtras apjungia LMVAP, Minimalios jėgos nustatymą ir kitus bandymus. Šis filtras naudoja adaptyvų limitą, kombinuojant su maksimalios ir minimalios jėgos skaičiavimais taip išskaičiuojant dažnių ribas, kuriose yra reikšmingiausia informacija, palyginus su kitais monotoniškais dažniais (tačiau nepanašus į Monotoniškų spektrų filtrą).

3 lentelė. Koreliacijų nuokrypių palyginimas

Pavadinimas	Originalo koreliacijos pokytis užterštam	Triukšmo koreliacijos pokytis užterštam	Koreliacijų santykinis skirtumas	Koreliuojamų duomenų tipas
Spektrų atimtis (S. F. Boll)	0.754700727	0.18899777	3.99	Garsas
MMSE LSA	0.982215062	0.43006079	2.28	Garsas
Vidutinės aukštos energijos dažniai	0.610727418	0.38285656	1.60	Amplitudės

Monotoniškų spektrų filtras	1.003176776	0.70254817	1.43	Garsas
Adaptyvaus limito maksimalios jėgos filtras	0.897082495	0.63308378	1.42	Garsas
Koreliacija tarp kanalų	0.9945272	0.93873522	1.06	Garsas
Minimalios energijos limitas	0.943168897	0.91674495	1.03	Garsas
Limituotas maksimalus vidutinis amplitudės pokyčio proporcijos skirtumas	-0.31738622	-0.405808	0.78	Amplitudės



65 pav. Kombinuojamų charakteristikų koreliacijų palyginimas

3.3. Akustinio triukšmo filtravimo algoritmo tyrimas

Tiriamos bei kombinuojamos reikšmingiausios akustinio triukšmo charakteristikų kombinacijos, siekiant pasiūlyti veiksmingą akustinio triukšmo filtravimo algoritmą.

Šiame poskyryje:

- Ištiriamos 2 apjungtų charakteristikų kombinacijos;
- Pasiūlomas akustinio triukšmo filtravimo algoritmas;
- Pasiūlomi egzistuojančių akustinio triukšmo filtravimo algoritmų patobulinimai.

3.3.1. Akustinio triukšmo filtravimo algoritmų kombinacijų bandymai

Aprašomos ir ištiriamos dvi reikšmingiausių akustinio triukšmo charakteristikų kombinacijų kombinacijos. Antroji kombinacija yra pirmosios tobulinimo rezultatas.

3.3.1.1. Pirmosios kombinacijos apžvalga

Ištirtų algoritmų atsakymai apjungiami į bendrą akustinio triukšmo filtravimo atsakymą pagal formulę (3.101):

$$filtruoti(EqF) = LMVAPP(EqF) + EFI(EqF) + EML(EqF) \quad (3.101)$$

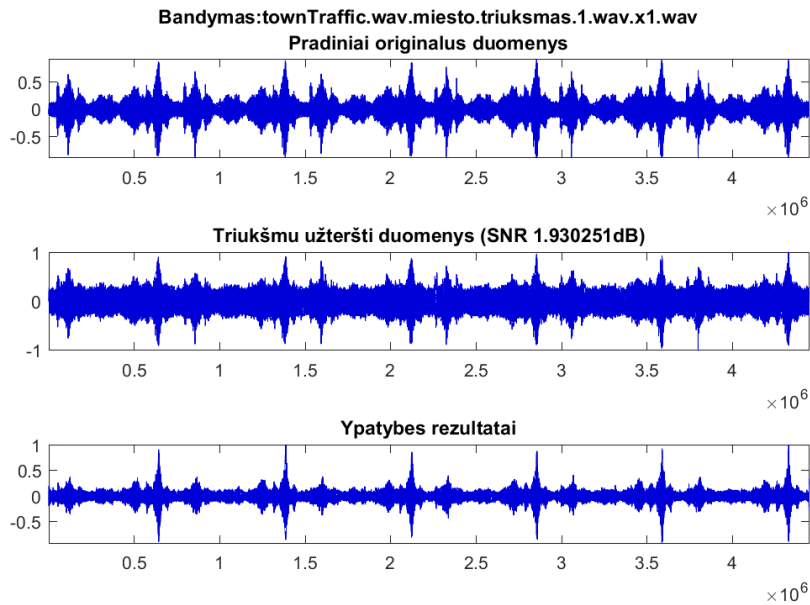
Panaudoti algoritmai:

- Vidutinės aukštos energijos dažniai
- Limituotas maksimalus vidutinis amplitudės triukšmo limitas
- Minimalios energijos nustatymas ir filtravimas

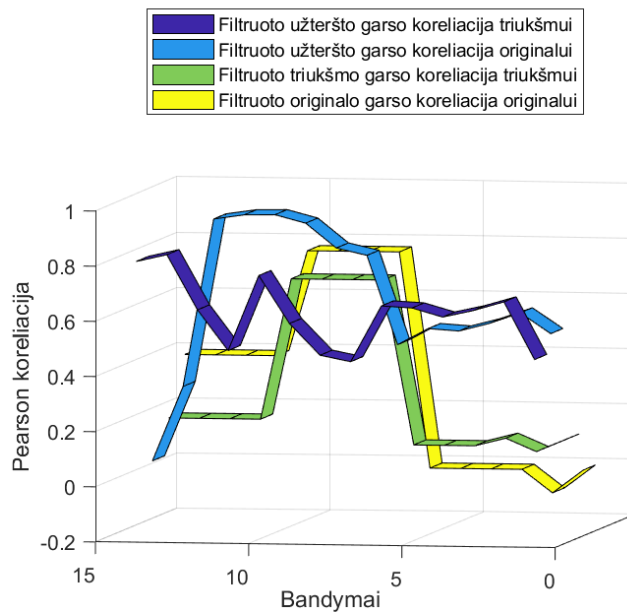
Ši kombinacija pateikė patenkinamus rezultatus daugelyje akustinio triukšmo filtravimo bandymų, tačiau buvo sunkiai optimizuojama plačiam akustinio triukšmo spektrui. Rezultatai buvo pernelyg mažai dinamiški, atsižvelgiant į triukšmingo įrašo SNR (esant 10 SNR įrašas būdavo prastai filtruojamas, palyginus su panašiu filtravimu esant 0 SNR) (žr. 66 pav., 67 pav.).

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – 64.1%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 55.1%;
- STI rodiklis – 0.234 (vidutinis), pradinis vidutinis rodiklis – 0.139;
- STOI rodiklis – 0.232 (vidutinis), pradinis vidutinis rodiklis – 0.259;
- Garso galia – 5.042 (vidutinė), pradinė vidutinė garso galia – 6.793.



66 pav. Pirmosios kombinacijos bandymo pavyzdys



67 pav. Pirmosios kombinacijos bandymų grafikas

3.3.1.2. Antrosios kombinacijos apžvalga

Ištirtų algoritmų atsakymai apjungiami į bendrą akustinio triukšmo filtravimo atsakymą, pagal formulę (3.102):

$$filtruoti(EqF) = DP(EqF) + MS(EqF) \quad (3.102)$$

DP aprašoma 3.2.1.6 skyriuje, 3.90 formulė, MS aprašoma 3.2.1.5 skyriuje, 3.79 formulė.

Panaudoti algoritmai:

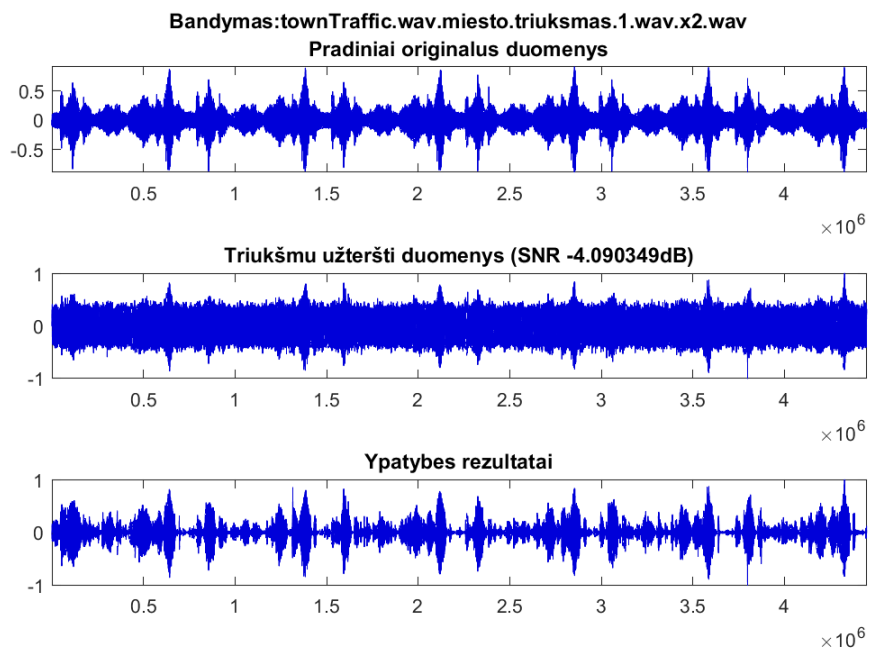
- Monotoniškų spektrų filtras

- Adaptyvaus limito maksimalios jėgos filtras

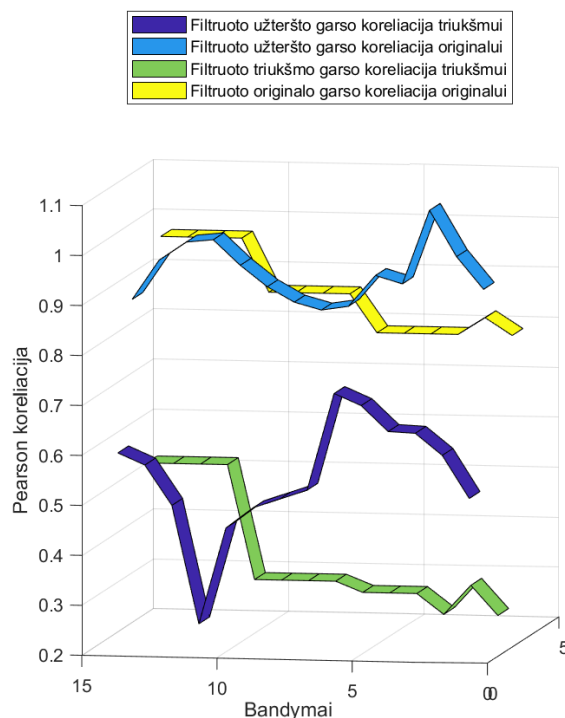
Ši kombinacija pateikė gerus rezultatus visose akustinio triukšmo filtravimo bandymuose (žr. 68 pav., 69 pav.). Pasirinkta naudoti šią kombinaciją, nes jos rezultatai

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – 94.2%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 55.1%;
- STI rodiklis – 0.192;
- STOI rodiklis – 0.260.



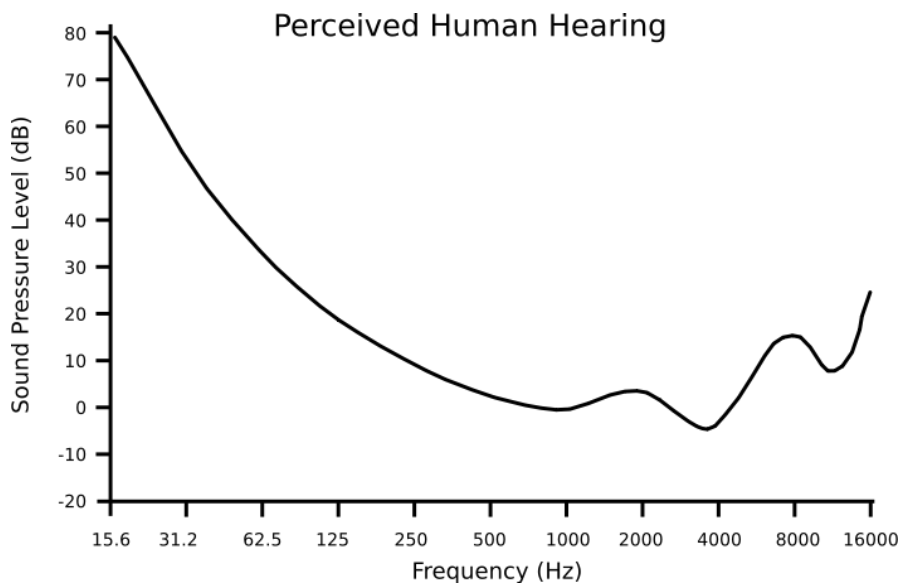
68 pav. Antrosios kombinacijos bandymo pavyzdys



69 pav. Antrosios kombinacijos bandymų grafikas

3.3.2. Algoritmo tobulinimas remiantis psichoakustika

Psichoakustika tai mokslas apie žmogaus subjektyvųjį garso suvokimą. Psichoakustika diktuoja, jog žmogus yra mažiau jautrus žemiems dažniams (20-250Hz) bei aukštiems dažniams (5000Hz ir daugiau). Tačiau įranga yra vienodai jautri visiems spektrams, todėl, norint patobulinti algoritmo sprendimus žmogaus atžvilgiu (sumažins koreliaciją kompiuterio atžvilgiu) būtina sumažinti algoritmo jautrumą žemiems bei aukštiems dažniams (arba padidinti jų filtravimą) [32] (žr. 70 pav.).



70 pav. Psichoakustikos diktuojamas žmogaus jautrumas dažnių spektrui, esant tokiems šių garsų lygiams mums garsai atrodytų vienodo garso. [32]

Dažnių jautrumo mažinimui priskiriamos šios reikšmės (MATLAB kodas):

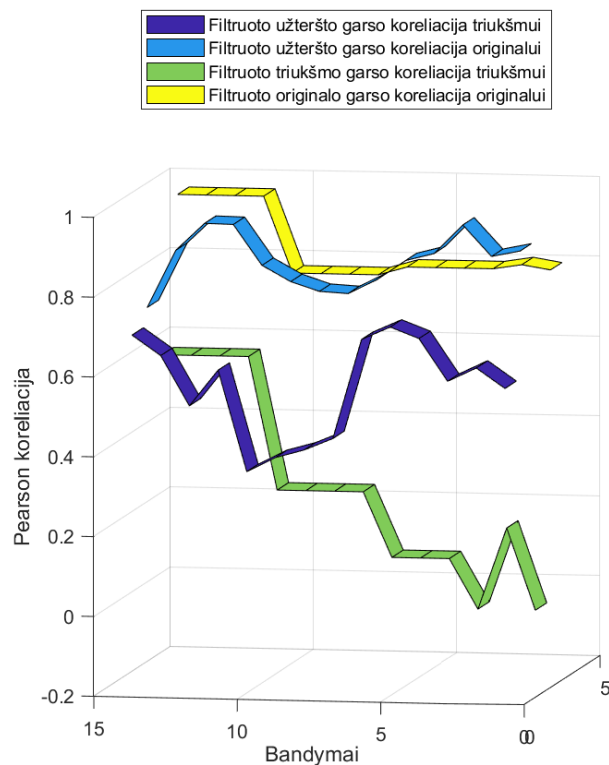
`energiesPsychoacustics = [5 4 3 2 1 1 1 1 2 3];`

Papildomai formulėms priskiriama `energiesPsychoacustics` koeficiento pavyzdys:

$$EFI = EFI * \text{energiesPsychoacustics} \quad (3.103)$$

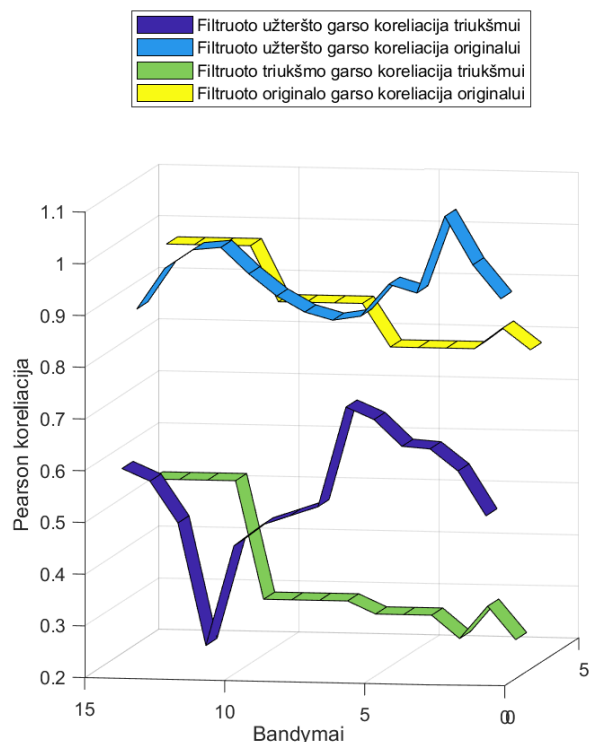
Psichoakustikos formulės įterptos 3.2.1 skyriuje.

Tiriant antrąją algoritmo kombinaciją be psichoakustinio modelio, gaunamas filtruoto užteršto įrašo norimų garsų koreliacijos skirtumas -14%, triukšmo koreliacijos skirtumas -43%. Algoritmas vis dar veiksnus, išfiltruoja vis 40% triukšmo prarandant 14% norimų garsų (žr. 71 pav.).



71 pav. Algoritmo bandymai be psichoakustinio modelio

Tiriant antrąją algoritmo kombinaciją pritaikius psichoakustinį modelį, gaunamas filtruoto užteršto įrašo norimų garsų koreliacijos skirtumas -6%, triukšmo koreliacijos skirtumas -45%. Algoritmas yra veiksnus, išfiltruoja vis 40% triukšmo prarandant 6% norimų garsų (žr. 72 pav.).



72 pav. Algoritmo bandymai pritaikius psichoakustinę modelį

Matoma, jog pritaikius Psichoakustikos modelį prarandama 6% mažiau norimų garsų ir išfiltruojama 2% daugiau triukšmo. Šie rezultatai eksponentiškai auga tiriant algoritmo konfigūracijas bei kombinacijas su kitais algoritmais, todėl galima teigti jog psichoakustinio modelio pritaikymas yra veiksmingas.

3.3.3. Algoritmo nustatymų tyrimas

Pasiūlytas (žr. 3.3.1.2 skyrių) algoritmas naudoja šiuos nustatymus (MATLAB kodo iškarpa):

```

pushFactor = 1.1;
param.diffProfilePerFreqAvgCnt = 5;
param.fftShortAvgCnt = 25;
param.fftLongAvgCnt = 200;
internalOne = 2;
internalTwo = 2;
internalDividedOne = 6;
internalDividedTwo = 1.8;
param.fftLongAvgCntShrt = internalDividedOne / ((pushFactor-
1)/internalOne+1);
param.thresholdDownWeight = internalDividedTwo / ((pushFactor-
1)/internalTwo+1);
param.thresholdAvgCnt = 200 * pushFactor;
param.threshold = 1;

```

```

param.freqsMinCnt = 500;
param.ADiMaxCnt = 500;
param.ADiMax = 10;
param.ADiAnswMin = 10;
param.ADiAnswMinCnt = 100;
param.peakGainsAvgCnt = 1;
param.datDiv = 1.0;
param.minOfMon = 0.2;
param.maxOfMon = 0.4;
param.monPlus = 0.1;
param.monIncrease = 2.2;
param.monPullDownNeighboursToDiv = 7;
param.peakGainsDiv = 1.5;
param.ADiWeightMin = 0.5;
param.ADiWeightMinus = 0.1;
param.ADiWeightOne = 1;
param.ADiWeightPlus = 0.1;

```

Atitinkamai, nustatymai išvesti iš eilės:

{1.1,5,25,200,6,1.8,220,1,500,500,10,10,100,1,1,0.2,0.4,0.1,2.2,7,1.5,0.5,0.1,1,0.1,2,2}

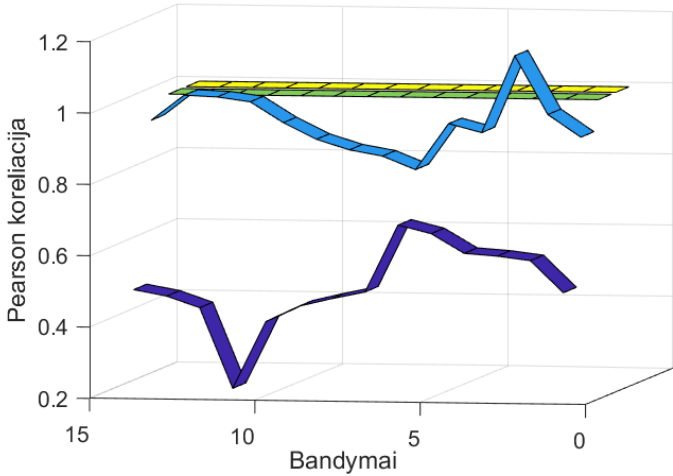
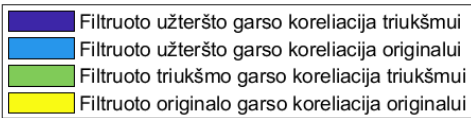
Algoritmas turi daug nustatymų, keičiant nustatymus gaunami skirtingi rezultatai. Todėl atlikti bandymai su skirtingais nustatymų variantais. Nustatymai pateikti eilėje atitinkamai 3.3.3 skyriuje pateiktiems nustatymams.

Verta pabrėžti, jog duomenų palyginimas koreliacijos būdu nėra pilnas algoritmo veiksmo apibrėžimas. Algoritmas išfiltruoja daug (~15%) originalaus pravažiuojančių mašinų garso, kadangi tai netrukdo suprasti kas vyksta, tačiau sumažina girdimo garsų dažnių spektro dydį (skirtingų dažnių kiekį) paliekant tik unikalčiausius duomenis, taip žmogui sumažinant nuovargį, palengvinant informacijos atpažinimą (žr. 2.1.2 skyrių). Filtruotiems garso įrašams, papildomai, apskaičiuojami STI, STOI garso įrašo kokybės indeksai (žr. 4 lentelę).

4 lentelė. Algoritmo nustatymų bandymai (ištrauka iš priedo nr.2)

Algoritmo nustatymai	Santykinė koreliacija norimiems garsams	Santykinė koreliacija užterštiems garsams	STI	STOI
1.1 5 25 20 6 1.8 220 1 500 500 10 10 100 1 1 0.2 0.4 0.1 2.2 7 1.5 0.5 0.1 1 0.1 2 2	0.917497777	0.449927902	0.210485085	0.263439318
1.1 5 25 200 6 1.8 220 1 500 500 10 10 100 1	0.88574475	0.417503038	0.229542993	0.26491005

1 0.2 0.5 0.1 2.2 7 1.5 0.5 0.1 1 0.1 2 2				
1.1 5 25 200 6 1.8 220 1 500 500 10 10 100 1 1 0.2 0.4 0.3 2.2 7 1.5 0.5 0.1 1 0.1 2 2	0.941800967	0.505368638	0.200297087	0.262925094
1.1 5 25 200 6 1.8 220 1 500 500 10 10 100 1 1 0.4 0.4 0.1 2.2 7 1.5 0.5 0.1 1 0.1 2 2	0.941553825	0.505403774	0.200297087	0.262925094
1.1 5 25 200 6 1.8 220 1 500 500 10 10 100 1 1 0.5 0.4 0.1 2.2 7 1.5 0.5 0.1 1 0.1 2 2	0.941553825	0.505403774	0.200297087	0.262925094
1.1 5 25 200 6 1.8 220 1 500 500 10 10 100 1 1 0.2 0.4 0.4 2.2 7 1.5 0.5 0.1 1 0.1 2 2	0.941553825	0.505403774	0.200297087	0.262925094
1.1 5 25 200 6 1.8 220 1 500 500 10 10 100 1 1 0.2 0.4 0.5 2.2 7 1.5 0.5 0.1 1 0.1 2 2	0.941553825	0.505403774	0.200297087	0.262925094



73 pav. Geriausios nustatymų kombinacijos rezultatų bandymo grafikas

Tiriant algoritmų nustatymus surasta optimali konfigūracija, kuri yra veiksmi visose akustinio triukšmo situacijose. Tiriant taip pat pastebėta, jog ši konfigūracija nėra optimali pavienėms situacijoms. Galimas algoritmo patobulinimas – naudoti skirtingas konfigūracijas skirtingose situacijose (miesto, darbo ir pan. triukšmams).

Tiriant akustinio triukšmo filtravimo algoritmą, pastebėta jog geriausiai akustinį triukšmą gerai apibrėžia dvi paprastos taisyklės:

- Tirtas akustinis triukšmas išlaiko savo vidutinę galią tame tikrame laiko tarpe, apibrėžtame dažnių rėžyje.
- Daugelis mums sutinkamų akustinių triukšmo atvejų nesudaro mažesnio SNR nei 0, tad galima teigti jog norimos garso informacijos branduolys dažniausia perneša daugiausia jėgos.

Kombinuojant šias taisykles galima aprašyti paprastesnę ir greitesnę algoritmą atpažįstantį akustinį triukšmą. Pasiūlytas algoritmas neapsiriboja 0 SNR, tiriami įrašai su mažiausiu -10.7SNR.

Surastas optimalus algoritmo kadravimo tankis - 0.043 sekundės.

Surasti optimalūs visų situacijų nustatymai: {1.1 5 25 150 6 1.8 220 1 500 500 10 10 10 1 1 0.2 0.4 0.2 2.2 7 1.5 0.5 0.1 1 0.1 2 2} (žr. 73 pav.).

Surasti optimalūs nustatymai, pritaikyti tik žmogaus kalbėjimui filtruoti: {1.1 5 5 200 6 1.8 220 1 500 500 10 10 100 1 1 0.2 0.4 0.1 2.2 7 1.5 0.5 0.1 1 0.1 2 2}. Filtruojant gauti rezultatai - +6% tikslesnis reikšmingų garsų atkūrimas, +3.9% daugiau išfiltruoto triukšmo, lyginant su optimaliais visų situacijų nustatymais.

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus (su optimaliais visų situacijų nustatymais):

- Koreliacija originaliam signalui (santykinė pradinei) – 94.8%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 50.7%;
- STI rodiklis – 0.200;
- STOI rodiklis – 0.2629.

Naudojant optimalius nustatymus visoms situacijoms, pasiūlytas algoritmas geba išfiltruoti vidutiniškai 49.29% triukšmo prarandant tik 5.14% reikšmingos garsų informacijos pagal pasvertą (užteršto garso koreliacijai) tiesioginę garso koreliaciją. Be to STI ir STOI rodikliai padidėjo atitinkamai 43.79%, 1.31%.

3.3.4. Reikalavimų atitikimo apžvalga

Apžvelgiami pasiūlyto akustinio triukšmo filtravimo algoritmo parametrai, kiekvieno reikalavimo atžvilgiu.

1. Algoritmas turi gebėti apdoroti duomenis realiu laiku (neturi atsižvelgti į turimus ateities duomenis testavimo įrašuose).

Algoritmas neanalizuoja ateities duomenų ir veikia apdorodamas kadrus, dėl to galima teigti jog jis pritaikomas realaus laiko duomenų apdorojimui. Testavimui naudojant i7-4720HQ procesorių, MATLAB aplinkoje, algoritmas geba apdoroti 221 sekundžių įrašo per 10 sekundžių (22 kartais greičiau nei realus laikas). Nenaudojant dažnių duomenų filtravimo, algoritmas akustinio triukšmo sprendimus 221 sekundžių įrašo išveda per 0.9 sekundės (245 kartais greičiau nei realus laikas). Algoritmo akustinio triukšmo sprendimų išeitis apjungta su MMSE LSA [38] algoritmu 221 sekundžių įrašo apdoroja per 7 sekundes (31 kartų greičiau nei realus laikas).

2. Algoritmas turi gebėti atskirti akustinį triukšmą nuo reikšmingų garsų;

Įrodyta, jog algoritmo išeitis teigiamai koreliuoja su reikšmingais garsais.

3. Algoritmo konfigūracija turi būti lanksti, t.y. norint patobulinti atpažinimą arba pritaikyti specifinei panaudai nekeičiant charakteristikų analizių ar kodo;

Algoritmas turi virš 15 konfigūracijos elementų.

4. Algoritmas turi gebėti išfiltruoti bent 40% akustinio triukšmo kai tai įmanoma (kai reikalinga informacija neperdengia akustinio triukšmo);

Algoritmas geba išfiltruoti vidutiniškai 49.29% triukšmo koreliuojant su užterštais garsais.

5. Algoritmas turi išlaikyti bent 80% reikiamos informacijos, žmogus turi greitai ir tiksliai suprasti kas tai per garsai.

Filtruojant užterštus garsus siūlomas algoritmas vidutiniškai praranda 5.14% norimos informacijos koreliuojant su užterštais garsais.

6. Algoritmas turi sumažinti plataus dažnių spektro informacijos dažnių diapazoną kai tai įmanoma (pvz.: miesto triukšmas, šlapiu keliu pravažiuojanti mašina skleidžia plataus diapazono garsus).

Algoritmas sumažina plataus dažnių spektrų diapazoną, išskirdamas tik unikaliausią informaciją, pagal dizainą. Tai sunku pamatuoti bandymų ar koreliacijų keliu, tačiau galima logiškai išvesti iš formulių. Be to pabrėžtina, jog būtent ši savybė, nors ir pageidautina, tačiau bandymų metu mažina teigiamą koreliaciją norimiems garsams ir neigiamai veikia STI bei STOI rodiklius.

7. Akivaizdu, jog algoritmas turės savo būseną aprašomą perduodamų parametrų, kurią įtakos netolima praeities analizė. Ši būseną atspindi filtruojamą aplinką ir netūrėtų apimti daugiau nei 30 praeities sekundžių, tam jog algoritmas būtų lankstus filtruojant dinaminį akustinį triukšmą;

Algoritmas kadruoja garsą 0.043 sekundės dalimis, tai reiškia jog algoritmo nustatymuose netūrėtų būti didesnių užlaikymų nei 697 kadru. Nustatymuose didžiausias užlaikymas yra 500 kadru, tačiau bandymu metu įrodyta, jog šiuos užlaikymus galima sumažinti iki 200 kadru (~8.6 sekundės) su minimaliu algoritmo veiksnimo sumažėjimu.

8. Algoritmas turi sumažinti plataus dažnių spektro triukšmą. Tokio triukšmo pamatavimas gali būti labai paprastas – 1024 kvadratinių dažnių reikšmių (GFT) vidurkis.

Algoritmas sumažina plataus dažnių spektro energiją nuo 6.79 iki 5.04. Bendra vidutinė garso energija sumažėja 34%.

3.3.5. Garso filtravimo algoritmų kombinavimas

Siekiant pasiūlyti veiksmingiausią akustinio triukšmo filtravimo algoritmą - atlikti tyrimai, kombinuojant pasiūlytą akustinio triukšmo filtravimo algoritmo akustinio triukšmo sprendimų išeitį su egzistuojančiais sprendimais. Algoritmas kombinuojamas su S. F. Boll [36] spektrų atimties, bei MMSE LSA [38] algoritmais.

3.3.5.1. Analoginis triukšmo rodiklis

Ištirtas akustinio triukšmo filtravimo algoritmas geba atpažinti akustinį triukšmą skirtingose dažniuose, tačiau pačio triukšmo filtravimo algoritmas išbandytas ganėtinai primityvus (reguliuojami 10 dažnių). Galima apjungti algoritmo dažnių sprendimus į bendrą analoginį triukšmo rodiklį ir taip išbandyti algoritmo veiksnumą tobulinant kitus triukšmo filtravimo algoritmus.

Išvesties apjungimui ir triukšmo apibendrinimui būtina apsirašyti naują formulę ADi (kadangi visi dažniai apjungiami į vieną rodiklį). Formulės įvestis bus akustinio triukšmo filtravimo algoritmo išvestis $filtruoti(EqF)$:

$$mostImportantFreqs = [1\ 4\ 8\ 8\ 8\ 5\ 3\ 1\ 1]; \quad (3.104)$$

$$param.freqsMin = \min(filtruoti, (param.freqsMin * (param.freqsMinCnt - param.datDiv) + filtruoti) / param.freqsMinCnt) \quad (3.105)$$

$$curFilter = filtruoti * mostImportantFreqs \quad (3.106)$$

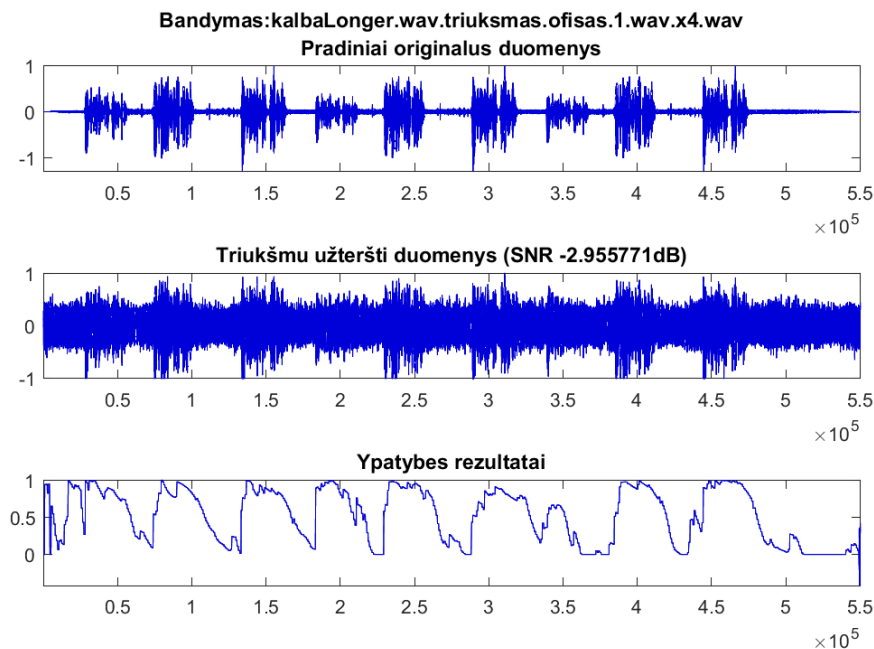
$$EDiVal = (|curFilter| - |param.freqsMin| * mostImportantFreqs)^2 \quad (3.107)$$

$$EDiVal = |\text{mean}(EDiVal)| \quad (3.108)$$

$$param.ADiMax = \max(EDiVal, (param.ADiMax * (param.ADiMaxCnt - param.ADiWeightOne) + EDiVal) / param.ADiMaxCnt) \quad (3.109)$$

$$ADi = \min \left(param.minOfMon, \max \left(param.maxOfMon, \left(EDiVal / param.ADiMax - param.ADiWeightMinus \right) * \left(param.ADiWeightOne + param.ADiAnswMin + param.ADiWeightPlus \right) \right) \right) \quad (3.110)$$

Gaunamas rezultatas apytiksliai nuo 0 iki ~1, kas reiškia reikšmingų garsų kiekį nuo 0 iki 1, t.y. 0 – reiškia tik triukšmą (žr. 74 pav.).



74 pav. analoginio triukšmo duomenų rodiklio išeities duomenų pavyzdys

3.3.5.2. S. F. Boll spektrų atimties filtro kombinavimas

Ištirtą akustinio triukšmo algoritmą galima panaudoti įvairiais būdais, pavyzdžiui kaip klasifikatorių vietoje S. F. Boll [36] spektrų atimties balsto aptykimo algoritmo. Kadangi ištirtas algoritmas geba apibrėžti triukšmą analogine kreive, o ne taip/ne atsakymą, galima patobulinti filtravimą įterpiančiam dinaminis skaičiavimus vietoje BAA algoritmo sąlygų.

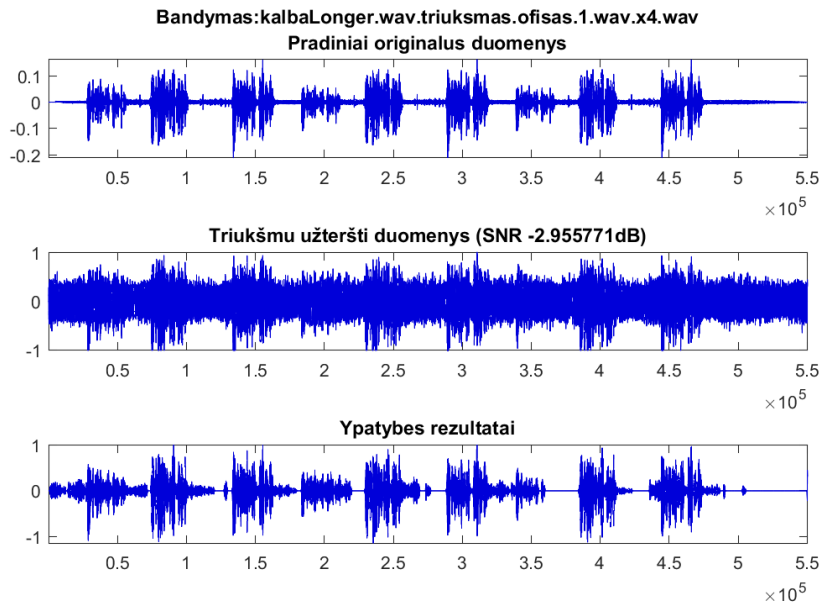
Triukšmo modelis gali būti apmokomas tik esant akustiniam triukšmui ($ADi < 0.3$), be to galima patobulinti kitus skaičiavimus.

Originalus S. F. Boll [36] spektrų atimties filtras atnaujina triukšmą tik neaptikus žmogaus balso, triukšmas atnaujinamas naudojant 10 kadrų vidurkį, be to visa amplitudė numušama iki 3% garso. Kadangi turime analoginį triukšmo įvertinimą ADi šias formules galima patobulinti pakeičiant naujomis, kur N yra triukšmo įvertinimas, Y yra garso įvestis, X yra garso išeitis, o V yra kadrų vidurkio kiekis (originaliai – 10):

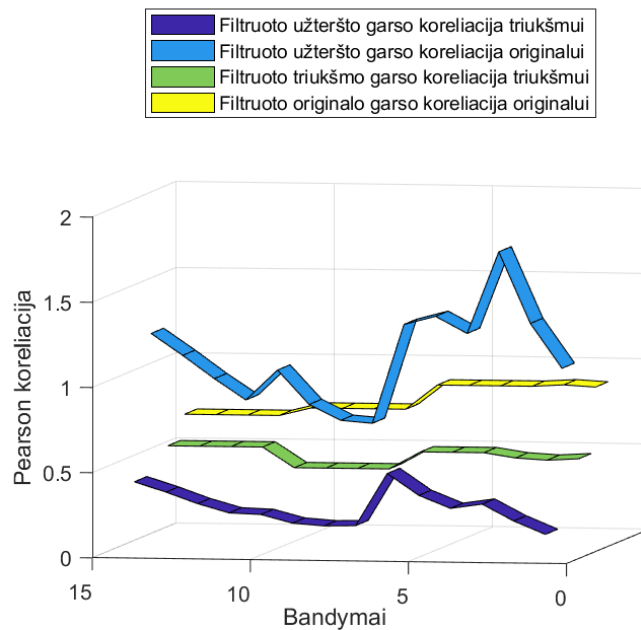
$$V = \begin{cases} V * 10 * (ADi + 0.1), & \text{kai } ADi < 0.3 \\ 150, & \text{kai } ADi \geq 0.3 \end{cases} \quad (3.111)$$

$$N = (N * X + Y)/(X + 1) \quad (3.112)$$

$$V = V * (ADi + 0.2) \quad (3.113)$$



75 pav. Patobulinto S. F. Boll [36] spektrų atimties algoritmo bandymo pavyzdys



76 pav. Patobulinto S. F. Boll [36] spektrų atimties algoritmo bandymų grafikas

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (santykinė pradinei) – 114.5%;
- Koreliacija triukšmo signalui (santykinė pradinei) – 27.1%;
- STI rodiklis – 0.363 (vidutinis), pradinis vidutinis rodiklis – 0.139;

- STOI rodiklis – 0.4249 (vidutinis), pradinis vidutinis rodiklis – 0.259;
- Garso galia – 1.886 (vidutinė), pradinė vidutinė garso galia – 6.793.

Patobulinus S. F. Boll [36] spektrų atimties algoritmą matomas ryškus koreliacijų skirtumas į teigiamą pusę. Algoritmo sukuriamas muzikinis triukšmas stipriai sumažėjo, norimų garsų išskyrimas pasidarė tikslesnis net -10 SNR įrašuose. Patobulintas S. F. Boll [36] spektrų atimties algoritmo norimų garsų atkūrimas padidėjo nuo -25% iki +12% (viso +37%), triukšmo išfiltravimas pakito nuo +82% iki +70% (viso -17%) palyginus su originalu tiesioginės koreliacijos būdu (žr. 75 pav., 76 pav.).

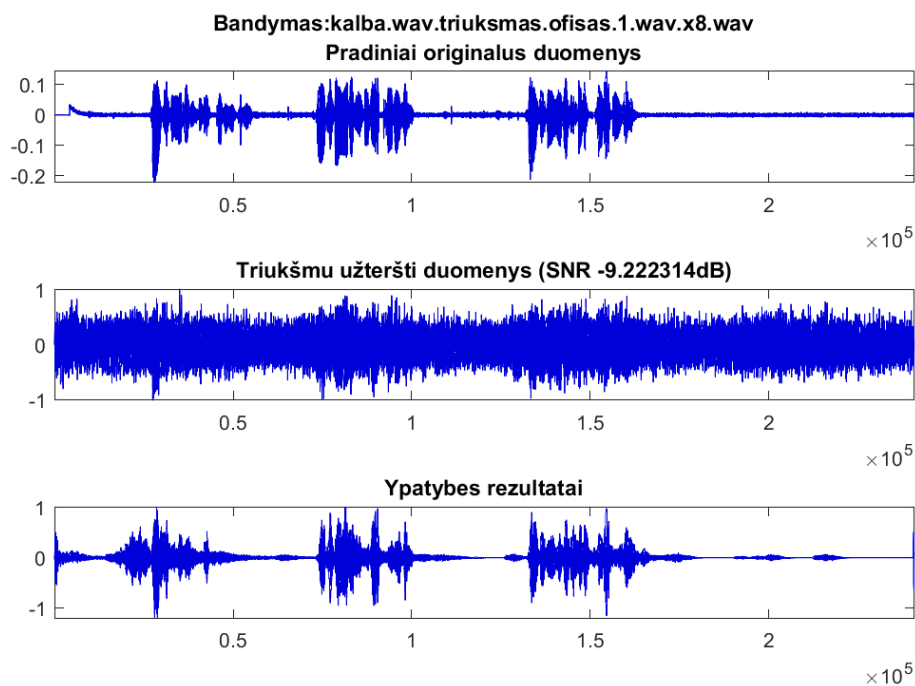
3.3.5.3. Minimalios vidutinės kvadratinės paklaidos logaritminės-spektrinės amplitudės įvertinimo filtro kombinavimas

Ištirtą akustinio triukšmo algoritmą galima panaudoti vietoje MMSE LSA [38] – BAA algoritmo. Kadangi ištirtas algoritmas geba apibrėžti triukšmą analogine kreive, o ne taip/ne atsakymą, galima patobulinti filtravimą įterpiant dinaminis skaičiavimus vietoje BAA algoritmo sąlygų.

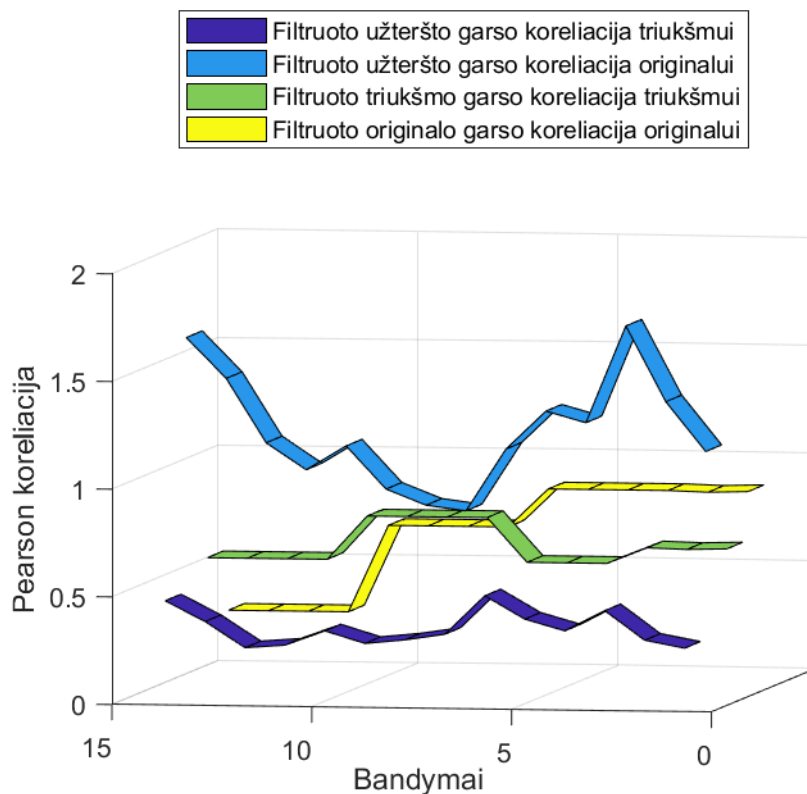
Triukšmo modelis apmokomas tik dominuojant akustinio triukšmo garsams ($ADI < 0.1$), be to galima reguliuoti amplitudės skaičiavimus analogine kreive, atnaujinamos šios algoritmo formulės:

$$l_{x,ML}(k) = \max\{p(Y_k(m)|l_x(k)), l_d(k)\} * (ADI + 0.2) \quad (3.114)$$

$$l_d = l_d * (ADI - 1), \text{ kai } ADI < 0.1 \quad (3.115)$$



77 pav. Patobulinto MMSE LSA [38] algoritmo bandymo pavyzdys



78 pav. Patobulinto MMSE LSA [38] algoritmo bandymų grafikas

Algoritmo įvertinimai filtruojant triukšmu užterštus garsus:

- Koreliacija originaliam signalui (vidutinė, santykinė pradinei) – 121.7%;
- Koreliacija triukšmo signalui (vidutinė, santykinė pradinei) – 30.2%;
- STI rodiklis – 0.333 (vidutinis), pradinis vidutinis rodiklis – 0.139;
- STOI rodiklis – 0.408 (vidutinis), pradinis vidutinis rodiklis – 0.259;
- Garso galia – 3.712 (vidutinė), pradinė vidutinė garso galia – 6.793.

Patobulinus MMSE LSA [38] algoritmą matomas ryškus koreliacijų skirtumas į teigiamą pusę. MMSE LSA [38] algoritmo norimų garsų atkūrimas pakito nuo -2% iki +21% (padidėjo 23%), triukšmo išfiltravimas pakito nuo +57% iki +70% (viso +13%) palyginus su originalu tiesioginės koreliacijos būdu. Be to STI bei STOI rodikliai pasikeitė atitinkamai +1.9%, +60% (žr. 77 pav., 78 pav.).

3.3.6. Akustinio triukšmo algoritmo tyrimo apžvalga ir pasiūlymai

Pasiūlytas ir ištirtas akustinio triukšmo filtravimo algoritmas. Tyrimų metu naudoti 14 ofiso, gatvės bei gamyklos triukšmais užterštų įrašų. Atlikus eksperimentinius tyrimus, nustatyta, jog algoritmas yra veiksmingas. Algoritmas geba išfiltruoti vidutiniškai 49.29% akustinio triukšmo, prarandant vidutiniškai 5.14% reikšmingų garsų, visų tyrimų metu matuojant pagal Pearson

koreliaciją, santykinai pradinei koreliacijai. Išfiltravus užterštą įrašą, garso kokybės rodikliai STI ir STOI padidėja atitinkamai - 43.79% ir 1.31%. Algoritmas sumažina vidutinę plataus dažnių spektro energiją nuo 6.79 iki 5.04 (garso energija sumažėja 34%). Atlikus algoritmo nustatymų tyrimus, surasti optimalūs akustinio triukšmo filtravimo algoritmo nustatymai, tinkami visoms ofiso, gatvės bei gamyklos testuojamoms situacijoms: {1.1 5 25 150 6 1.8 220 1 500 500 10 10 10 1 1 0.2 0.4 0.2 2.2 7 1.5 0.5 0.1 1 0.1 2 2}. Pastebėta, jog algoritmas gali dirbti efektyviau, naudojant atskirus nustatymus skirtingose aplinkose (ofiso, gamyklos ir pan. triukšmams). Atlikus eksperimentinius tyrimus nustatyta optimali konfigūracija {1.1 5 5 200 6 1.8 220 1 500 500 10 10 100 1 1 0.2 0.4 0.1 2.2 7 1.5 0.5 0.1 1 0.1 2 2}, kuri pritaikyta žmogaus kalbėjimui (ofisui). Optimalios ofiso įrašų tyrimų konfigūracijos rezultatai: 6% didesnė reikšmingų garsų atkūrimo koreliacija, 3.9% daugiau išfiltruoto triukšmo matuojant Pearson koreliacijos būdu, santykinai pradinei koreliacijai.

Pasiūlyti S. F. Boll [36] spektrų atimties bei MMSE LSA [38] triukšmo filtravimo algoritmų patobulinimai, kurie remiasi pasiūlyto akustinio triukšmo filtravimo algoritmo analogine išėjimu, aprašančia akustinio triukšmo kiekį kiekviename garso kadre. Tyrimų metu nustatyta, jog patobulinto S. F. Boll [36] spektrų atimties algoritmo reikšmingų garsų atkūrimas padidėjo nuo -25% iki +12% (viso +37%), triukšmo išfiltravimas sumažėjo nuo +82% iki +70% (viso -17%) matuojant Pearson koreliacijos būdu, santykinai pradinei koreliacijai. Tyrimų metu nustatyta, jog patobulinto MMSE LSA [38] algoritmo reikšmingų garsų atkūrimas pakito nuo -2% iki +21% (viso +23%), triukšmo išfiltravimas pakito nuo +57% iki +70% (viso +13%), matuojant Pearson koreliacijos būdu, santykinai pradinei koreliacijai. S. F. Boll [36] spektrų atimties algoritmo garso kokybės rodikliai STI ir STOI atitinkamai pasikeitė -1%, +67%. MMSE LSA [38] algoritmo garso kokybės rodikliai STI ir STOI atitinkamai pasikeitė +1.9%, +60%;

Tyrimų metu nustatyta, jog tiksliausiai testuojamus įrašus išfiltruoja patobulintas MMSE LSA [38] algoritmas. Reikšmingų garsų atkūrimas vidutiniškai siekia +12% santykinai pradinėms užteršto įrašo koreliacijoms pagal Pearson. Bendras išfiltruojamo triukšmo kiekis vidutiniškai siekia -70%, santykinai pradinėms užteršto įrašo koreliacijoms pagal Pearson. Vidutiniai garso kokybės rodikliai STI ir STOI padidėja atitinkamai 138% bei 57%, pradinėms užteršto įrašo rodikliams. Algoritmas sumažina vidutinę plataus dažnių spektro energiją nuo 6.79 iki 3.79 (garso energija sumažėja 79%).

Remiantis tyrimais, siūloma naudoti skirtingas algoritmų konfigūracijas, filtruojant akustinį triukšmą iš skirtingų aplinkų. Galimas algoritmo patobulinimas – atpažinti aplinką ir pritaikyti skirtingą konfigūraciją, ar ir skirtingą filtravimo algoritmą.

4. IŠVADOS IR REKOMENDACIJOS

1. Ištirti garso analizės bei filtravimo algoritmai susitelkia į siauro spektro problemos sprendimą (dažnai tik žmogaus balso filtravimui), trūksta bendrojo akustinio triukšmo spektro filtravimo sprendimų (miesto, gamyklos, ofiso aplinkos ir kt.) (žr. 1.12 skyrių);

2. Siekiant surasti garso charakteristikas, kurios koreliuoja su reikšmingais arba akustinio triukšmo garsais,- ištirta 16 garso charakteristikų. Norint išsiaiškinti charakteristikų veiksmingumą, tyrimo metu buvo naudojami 14 aplinkos (ofiso, gatvės, gamyklos) įrašų, koreliuojant algoritmų išvestį su triukšmo bei reikšmingais garso duomenimis (pagal Pearson arba Spearman koreliaciją). Tyrimų metu nustatytos šios reikšmingiausios charakteristikos: dominuojantys dažniai, dažnių energija, spektro geometrinis tiesiškumas, dažnių triukšmingumas, mažos jėgos spektro kiekis, vidutinė kvadratinė vertė, spektrų atstumas;

3. Siekiant pasiūlyti veiksmingą garsų filtravimo algoritmą,- pasiūlyti ir ištirti charakteristikų apjungimo algoritmai. Atlikus eksperimentinius tyrimus - nustatyti veiksmingiausi algoritmai: Monotoniškų spektrų filtras (vidutinė Pearson koreliacija originaliam signalui– 100%, vidutinė Pearson koreliacija triukšmo signalui– 70.2%) bei adaptyvaus limito maksimalios jėgos filtras (vidutinė Pearson koreliacija originaliam signalui– 98.7%, vidutinė Pearson koreliacija triukšmo signalui– 63.3%);

4. Pasiūlytas ir ištirtas akustinio triukšmo filtravimo algoritmas. Tyrimų metu naudoti 14 ofiso, gatvės bei gamyklos triukšmais užterštų įrašų. Atlikus eksperimentinius tyrimus, nustatyta, jog algoritmas yra veiksmingas. Atlikus algoritmo nustatymų tyrimus, surasti optimalūs akustinio triukšmo filtravimo algoritmo nustatymai, tinkami visoms ofiso, gatvės bei gamyklos testuojamoms situacijoms. Algoritmas geba išfiltruoti vidutiniškai 49.29% akustinio triukšmo, prarandant vidutiniškai 5.14% reikšmingų garsų, visų tyrimų metu algoritmo išeitį koreliuojant pagal Pearson koreliaciją, santykinai pradinei koreliacijai. Išfiltravus užterštą įrašą, garso kokybės rodikliai STI ir STOI padidėja atitinkamai - 43.79% ir 1.31%. Algoritmas sumažina vidutinę plataus dažnių spektro energiją nuo 6.79 iki 5.04 (garso energija sumažėja 34%). Pastebėta, jog algoritmas gali dirbti efektyviau, naudojant atskirus nustatymus skirtingose aplinkose (ofiso, gamyklos ir pan. triukšmams). Atlikus eksperimentinius tyrimus nustatyta optimali konfigūracija, kuri pritaikyta žmogaus kalbėjimui (ofisui). Optimalios ofiso įrašų tyrimų konfigūracijos rezultatai: 6% didesnė reikšmingų garsų atkūrimo koreliacija, 3.9% daugiau išfiltruoto triukšmo matuojant Pearson koreliacijos būdu, santykinai pradinei koreliacijai.

5. Pasiūlyti S. F. Boll [36] spektrų atimties bei MMSE LSA [38] triukšmo filtravimo algoritmų patobulinimai, kurie remiasi pasiūlyto akustinio triukšmo filtravimo algoritmo analogine išeitimi, aprašančia akustinio triukšmo kiekį kiekviename garso kadre. Tyrimų metu nustatyta, jog patobulinto

S. F. Boll [36] spektrų atimties algoritmo reikšmingų garsų atkūrimas padidėjo nuo -25% iki +12% (viso +37%), triukšmo išfiltravimas sumažėjo nuo +82% iki +70% (viso -17%) matuojant Pearson koreliacijos būdu, santykinai pradinei koreliacijai. Tyrimų metu nustatyta, jog patobulinto MMSE LSA [38] algoritmo reikšmingų garsų atkūrimas pakito nuo -2% iki +21% (viso +23%), triukšmo išfiltravimas pakito nuo +57% iki +70% (viso +13%), matuojant Pearson koreliacijos būdu, santykinai pradinei koreliacijai. S. F. Boll [36] spektrų atimties algoritmo garso kokybės rodikliai STI ir STOI atitinkamai pasikeitė -1%, +67%. MMSE LSA [38] algoritmo garso kokybės rodikliai STI ir STOI atitinkamai pasikeitė +1.9%, +60%;

6. Tyrimų metu nustatyta, jog tiksliausiai testuojamus įrašus išfiltruoja patobulintas MMSE LSA [38] algoritmas. Reikšmingų garsų atkūrimas vidutiniškai siekia +12% santykinai pradinėms užteršto įrašo koreliacijoms pagal Pearson. Bendras išfiltruojamo triukšmo kiekis vidutiniškai siekia -70%, santykinai pradinėms užteršto įrašo koreliacijoms pagal Pearson. Vidutiniai garso kokybės rodikliai STI ir STOI padidėja atitinkamai 138% bei 57%, pradinėms užteršto įrašo rodikliams. Algoritmas sumažina vidutinę plataus dažnių spektro energiją nuo 6.79 iki 3.79 (garso energija sumažėja 79%). Remiantis tyrimais, rekomenduojama naudoti skirtingas algoritmų konfigūracijas, filtruojant akustinį triukšmą iš skirtingų aplinkų. Rekomenduojamas algoritmo patobulinimas – atpažinti aplinką ir pritaikyti skirtingą konfigūraciją, ar ir skirtingą filtravimo algoritmą.

6. ŠALTINIAI

1. *Akustinis Triukšmas* [interaktyvus]. [žiūrėta 2016-10-15]. Prieiga per: https://lt.wikipedia.org/wiki/Triuk%C5%A1mas#Akustinis_triuk.C5.A1mas
2. *Noise impacts on health (Europe comission summary)* [interaktyvus]. [žiūrėta 2016-10-17]. Prieiga per: <http://ec.europa.eu/environment/integration/research/newsalert/pdf/47si.pdf>
3. GORDON E. MOORE, *Cramming more components onto integrated circuits* [interaktyvus]. [žiūrėta 2016-10-19]. Prieiga per: <https://drive.google.com/file/d/0By83v5TWkGjvQkpBcXJKT1IITTA/view>
4. *A Simple But Efficient Real-Time Voice Activity Detection Algorithm* [interaktyvus]. [žiūrėta 2016-10-21]. Prieiga per: <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2009/contents/papers/1569192958.pdf>
5. CHRISTOS STERGIOU, DIMITRIOS SIGANOS, *Neural Networks* [interaktyvus]. [žiūrėta 2016-10-22]. Prieiga per: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
6. *Greitos Furjė Transformacijos* [interaktyvus]. [žiūrėta 2016-10-23]. Prieiga per: <http://www.vartiklis.lt/science/math/in-brief.htm>
7. *Time Complexity* [interaktyvus]. [žiūrėta 2016-10-25]. Prieiga per: http://btechsmartclass.com/DS/UI_T4.html
8. *Fast Fourier Transform* [interaktyvus]. [žiūrėta 2016-10-25]. Prieiga per: https://en.wikipedia.org/wiki/Fast_Fourier_transform
9. *Voice Activity Detection (VAD) Tutorial* [interaktyvus]. [žiūrėta 2016-10-28]. Prieiga per: <http://practicalcryptography.com/miscellaneous/machine-learning/voice-activity-detection-vad-tutorial/>
10. SCOTT C. DOUGLAS, *Introduction to Adaptive Filters* [interaktyvus]. [žiūrėta 2016-11-05]. Prieiga per: <https://pdfs.semanticscholar.org/aa48/98919244e59159bd276109b10dfbaa5dedf1.pdf>
11. S. DUBNOV, *Spectral flatness* [interaktyvus]. [žiūrėta 2016-11-05]. Prieiga per: <https://ieeexplore.ieee.org/document/1316889/>
12. *Short Term Processing of Speech* [interaktyvus]. [žiūrėta 2016-11-05]. Prieiga per: <http://iitg.vlab.co.in/?sub=59&brch=164&sim=857&cnt=1>
13. *Dominant Frequency Extraction* [interaktyvus]. [žiūrėta 2016-11-10]. Prieiga per: <https://arxiv.org/pdf/1306.0103.pdf>
14. *VAD Techniques for Real-Time Speech Transmission on the Internet* [interaktyvus]. [žiūrėta 2016-11-10]. Prieiga per: <http://homepage.tudelft.nl/w5p50/pdffiles/VAD%20Techniques%20for%20Real-Time%20Speech%20Transmission%20on%20the%20Internet.pdf>
15. *Least mean squares filter* [interaktyvus]. [žiūrėta 2016-11-10]. Prieiga per: https://en.wikipedia.org/wiki/Least_mean_squares_filter
16. *Least Mean Square (LMS) Adaptive Filter* [interaktyvus]. [žiūrėta 2016-11-15]. Prieiga per: <http://www.ni.com/example/31220/en/>
17. *Vinerio filtrai* [interaktyvus]. [žiūrėta 2016-11-15]. Prieiga per: <http://www2.el.vgtu.lt/~sarunas/lithuanian/students/spec/wiener.pdf>
18. *Adaptivūs Mažiausios Kvadratinės Paklaidos Filtrai* [interaktyvus]. [žiūrėta 2016-11-15]. Prieiga per: <http://www2.el.vgtu.lt/~sarunas/lithuanian/students/spec/adaptive.pdf>

19. *Wiener filtering* [interaktyvus]. [žiūrėta 2016-11-15]. Prieiga per: http://www.logicnet.dk/reports/EEG/ImprovingEP.htm#_Toc24556383
20. *Wiener filter* [interaktyvus]. [žiūrėta 2016-11-20]. Prieiga per: https://en.wikipedia.org/wiki/Wiener_filter
21. *Ivadas į filtrus* [interaktyvus]. [žiūrėta 2016-11-20]. Prieiga per: http://www.ifko.ktu.lt/~girrol/Skaidres/Literatura/Lietuviska/DSP_Ivadas_Filtrai.pdf
22. *FIR Filter For Audio Practitioners* [interaktyvus]. [žiūrėta 2016-11-20]. Prieiga per: http://www.hxaudiolab.com/uploads/2/5/5/3/25532092/fir_filter_for_audio_practitioners.pdf
23. *Triukšmo slopinimas* [interaktyvus]. [žiūrėta 2017-01-01]. Prieiga per: <http://www2.el.vgtu.lt/~sarunas/lithuanian/students/spec/noise.pdf>
24. *REPET: Repeating Pattern Extraction Technique* [interaktyvus]. [žiūrėta 2017-01-01]. Prieiga per: <http://www.zafarrafii.com/repet.html>
25. *Deep Learning For Monaural Speech Separation* [interaktyvus]. [žiūrėta 2017-01-01]. Prieiga per: https://posenhuang.github.io/papers/DNN_Separation_ICASSP2014.pdf
26. *Deep Learning For Monaural Source Separation* [interaktyvus]. [žiūrėta 2017-01-05]. Prieiga per: <https://sites.google.com/site/deeplearningsourceseparation/>
27. *Deep Learning For Monaural Source Separation* [išėities kodai]. [žiūrėta 2017-01-05]. Prieiga per: <https://github.com/posenhuang/deeplearningsourceseparation>
28. *A Neural Network for Real-Time Signal Processing* [interaktyvus]. [žiūrėta 2017-01-05]. Prieiga per: <https://papers.nips.cc/paper/284-a-neural-network-for-real-time-signal-processing.pdf>
29. *Impulse Noise (audio)* [interaktyvus]. [žiūrėta 2017-01-10]. Prieiga per: [https://en.wikipedia.org/wiki/Impulse_noise_\(audio\)](https://en.wikipedia.org/wiki/Impulse_noise_(audio))
30. *Noise abatement approaches* [interaktyvus]. [žiūrėta 2017-10-15]. Prieiga per: http://ec.europa.eu/environment/integration/research/newsalert/pdf/noise_abatement_approaches_FB17_en.pdf
31. *Warning and emergency signs -Urban noise filtering for better driver performance* [interaktyvus]. [žiūrėta 2017-11-15]. Prieiga per: https://www.researchgate.net/publication/257937412_Warning_and_emergency_signs_-_Urban_noise_filtering_for_better_driver_performance
32. *Psichoakustikos santrauka* [interaktyvus]. [žiūrėta 2017-11-15]. Prieiga per: <https://en.wikipedia.org/wiki/Psychoacoustics>
33. *Digital signal processing algorithms for noise reduction, dynamic range compression, and feedback cancellation in hearing aids* [interaktyvus]. [žiūrėta 2017-05-15]. Prieiga per: ftp://ftp.esat.kuleuven.be/sista/kngo/reports/phd_KimNgo.pdf
34. *Orlaivių integruotos diagnostikos sistemų matavimų duomenų perdavimo saugos metodikos sukūrimas* [interaktyvus]. [žiūrėta 2017-11-15]. Prieiga per: http://dspace.vgtu.lt/bitstream/1/928/3/1975_Rudinskas_Disertacija_WEB.pdf
35. *Kalmano filtras* [interaktyvus]. [žiūrėta 2017-05-12]. Prieiga per: https://lt.wikipedia.org/wiki/Kalmano_filtras
36. STEVEN F. S. F. BOLL, *Suppression of Acoustic Noise in Speech Using Spectral Subtraction* [interaktyvus]. [žiūrėta 2017-10-12]. Prieiga per: <https://pdfs.semanticscholar.org/0e40/0ee7aa7e53777596eae63c4062a657d60f5.pdf>

37. 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* [interaktyvus]. [žiūrėta 2018-03-22]. Prieiga per: <https://www.itu.int/rec/T-REC-P.862/en>
38. JASON WUNG, SHIGEKI MIYABE, BIING-HWANG (FRED) JUANG, *Speech Enhancement Using Minimum Mean-Square Error Estimation* [interaktyvus]. [žiūrėta 2018-03-22]. Prieiga per: <https://pdfs.semanticscholar.org/e226/4ce6795c93677706f4299b26d895a300637f.pdf>
39. *Speech transmission index* [interaktyvus]. [žiūrėta 2018-03-22]. Prieiga per: https://en.wikipedia.org/wiki/Speech_transmission_index
40. *TAAL2011 - The Short-time objective intelligibility measure (STOI)* [interaktyvus]. [žiūrėta 2018-03-24]. Prieiga per: <http://amtoolbox.sourceforge.net/amt-0.9.8/doc/models/taal2011.php>
41. *A Short-Time Objective Intelligibility Measure For Time-Frequency Weighted Noisy Speech* [interaktyvus]. [žiūrėta 2018-03-24]. Prieiga per: <http://cas.et.tudelft.nl/pubs/Taal2010.pdf>
42. *The Effects Of Low Frequency Noise On Peoplea Review* [interaktyvus]. [žiūrėta 2017-02-24]. Prieiga per: <https://pdfs.semanticscholar.org/1cae/b9d9a78ae539310579dd9563578d7da2d4c4.pdf>
43. *Noise Induced Hearing Loss* [interaktyvus]. [žiūrėta 2017-03-24]. Prieiga per: <https://www.american-hearing.org/disorders/noise-induced-hearing-loss/>
44. *Hearing in Laboratory Animals: Strain Differences and Nonauditory Effects of Noise* [interaktyvus]. [žiūrėta 2017-03-24]. Prieiga per: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3725606/>
45. *Environmental noise exposure degrades normal listening processes* [interaktyvus]. [žiūrėta 2017-03-24]. Prieiga per: <https://www.nature.com/articles/ncomms1849>
46. JINGDONG CHEN, JACOB BENESTY, SENIOR MEMBER, YITENG (ARDEN) HUANG, AND SIMON DOCLO, *New Insights Into the Noise Reduction Wiener Filter* [interaktyvus]. [žiūrėta 2017-03-24]. Prieiga per: <ftp://ftp.esat.kuleuven.be/sista/doclo/reports/04-239.pdf>

7. PRIEDAI

7.1. 1 Priedas. Kalendorinis darbų atlikimo grafikas.

5 lentelė. Priedas: Magistrinio darbo kalendorinis grafikas.

Eil. Nr.	Etapas	Atlikimo laikotarpis nuo iki		Laukiamas rezultatas ir jo pateikimo forma
1-asis semestras				
1.	Pasirinkti magistrinio projekto temą ir darbo vadovą.	2016-09-09	2016-10-21	Temos pavadinimas suderintas su vadovu ir pateiktas studijų programos kuratoriui
2.	Nustatyti tyrimo objektą ir suformuluoti preliminarius darbo tikslus bei uždavinius.	2016-10-21	2016-11-31	Tyrimo objektas bei preliminarūs darbo tikslai aprašyti tarpinėje tiriamojo darbo ataskaitoje
3.	Parinkti analizės metodus ir priemones.	2016-10-21	2016-11-31	Numatomi analizės metodai ir priemonės aprašytos tarpinėje tiriamojo darbo ataskaitoje
4.	Atlikti literatūros apžvalgą.	2016-10-21	2016-12-23	Literatūros tyrimas aprašytas tarpinėje ataskaitoje sprendimų tyrimo dalyje.
5.	Įvertinti egzistuojančių algoritmų tinkamumą, aktualumą.	2016-11-21	2017-01-15	Sprendimų tyrimas aprašytas tarpinėje ataskaitoje sprendimų tyrimo dalyje.
2-asis semestras				
6.	Išanalizuoti pasirinktus algoritmus.	2017-01-15	2017-02-20	Sprendimų tyrimas aprašytas tarpinėje ataskaitoje sprendimų tyrimo dalyje.
7.	Nustatyti algoritmo reikalavimus.	2017-02-20	2017-04-25	Aprašomi algoritmo funkciniai bei nefunkciniai reikalavimai, kokybės reikalavimai.
8.	Nustatyti algoritmo realizavimo būdus.	2017-02-20	2017-04-25	Aprašomi algoritmo realizavimo būdai, skirti testavimui ir bandymams.
9.	Realizuoti algoritmo modelį.	2017-03-12	2017-05-05	Aprašomas algoritmo modelis.
3-asis semestras				
10.	Apibrėžti tyrimo būdą, reikalavimus.	2017-09-20	2017-09-25	Apibrėžtas algoritmo tyrimo būdas, tyrimo iteracijų reikalavimai.
11.	Ištirti garso charakteristikas.	2017-09-25	2017-12-25	Ištirtos garso charakteristikos, padarytos išvados.

12.	Ištirti garso charakteristikų apjungimą.	2017-10-25	2017-12-25	Ištirtos garso charakteristikų apjungimo galimybės, padarytos išvados.
13.	Suprojektuoti algoritmą	2017-12-25	2018-01-10	Suprojektuotas dalinai veikiantis algoritmas, galima pradėti algoritmo bandymus, tobulinimą.
4-asis semestras				
14.	Ištirti algoritmą	2018-01-10	2018-02-13	Ištiriamas algoritmas, jo nustatymai, veiksnumas ir galimi patobulinimai.
15.	Ištirti apjungimą su kitais algoritmais	2018-02-13	2018-03-13	Ištiriamas algoritmo apjungimas su kitais algoritmais, galimos kombinacijos.
16.	Užbaigti ataskaitą	2018-03-13	2018-04-01	Užbaigiama ataskaita, pateikiamos išvados.
17.	Pasiruošti pristatymui	2018-04-01	2018-04-29	Pasiruošiama darbo pristatymui.

5 lentelė yra ištrauka iš dokumento Magistrinis darbas Arnas Navašinskas kalendorinis grafikas.doc (aplankale „ataskaitos priedai“).

7.2. 2 Priedas. Skaičiavimų lentelės.

Algoritmų testavimų, jų konfigūracijų testavimo, charakteristikų tyrimų ir t.t. rezultatų lentelės pateikiamos ataskaitos failų prieduose (aplankale „ataskaitos priedai“ – „bandymų duomenys“).

Lentelės yra per didelės tekstiniam dokumentui (virš 5000 eilučių), lentelių ištraukos pateikiamos tekste.

7.3. 3 Priedas. MATLAB kodas.

Pateikiama MATLAB įrankio kodo ištrauka. Visas kodas pateiktas failų prieduose prie darbo ataskaitos (aplankale „ataskaitos priedai“ – „kodas“). Paleisties failo pavadinimas yra „Main.m“

Ištrauka iš failo „EDD.m“. Pateikiamas antrojo bei paskutiniojo pasiūlyto filtravimo algoritmo pagrindinis kodas:

```
function [originalData, AD] = EDD( originalData, sampleRate, frameSize, ...
parametersStructure, frequencies, frequencyUnimportanceWeight, targetDbToLower, ...
qualityFactors, minimumEnergyAllowed, lowEnergiesHasMorePower )
```

```
AD = zeros( (length(originalData)), 2); %variable for analog noise activity
detector output
```

```
numFrame = floor( (length(originalData))/frameSize+1);
numEqBands = size(frequencies);
```



```

numEqBands = numEqBands(2);
frequencyGap = sampleRate/frameSize/2; %famed fft
midFrameFreq = round(frequencies/frequencyGap); %in our framed data

fftPerFreqShort = zeros(2,numEqBands);
fftPerFreqLong = zeros(2,numEqBands);
diffProfilePerFreq = zeros(2,numEqBands);
peakGains = zeros(2,numEqBands);

MPEleft = multibandParametricEQ('NumEQBands',numEqBands,...
    'Frequencies',frequencies,...
    'QualityFactors',qualityFactors,...
    'PeakGains', peakGains(1,:),...
    'SampleRate',sampleRate);

MPEright = multibandParametricEQ('NumEQBands',numEqBands,...
    'Frequencies',frequencies,...
    'QualityFactors',qualityFactors,...
    'PeakGains', peakGains(2,:),...
    'SampleRate',sampleRate);

frame_start = 1;
frame_end = frameSize;
for frame=1:numFrame

[originalData(frame_start:frame_end,:),AD(frame_start:frame_end,:),parametersStructure] = analyzeFrame(originalData(frame_start:frame_end,:),
AD(frame_start:frame_end,:), parametersStructure, frame, frameSize);
    frame_start = frame_start+frameSize;
    frame_end = min(frame_end+frameSize,length(originalData));
end

function [xDataFrame,ADi,param] = analyzeFrame (xDataFrame, ADi, param,
frame, frameSize)

    datSize = size(xDataFrame);

    if (datSize ~= frameSize)
        return;
    end

    [xDataFrame,ADi, param] = testFeature(xDataFrame, ADi, param, frame,
frameSize);

end

function res = growingFunc(currentValue, newValue, growInIterations,
growHigherFasterByPercent, growLowerFasterByPercent)
    if (newValue > currentValue)
        res = (currentValue*(growInIterations-1-
round(growInIterations*growHigherFasterByPercent))+newValue*round(growInIteration
s*growHigherFasterByPercent))/growInIterations;
    else
        res = (currentValue*(growInIterations-1-
round(growInIterations*growLowerFasterByPercent))+newValue*round(growInIterations
*growLowerFasterByPercent))/growInIterations;
    end

return;

```

```

end

function [xDataFrame,ADi, param] = testFeature(xDataFrame, ADi, param, frame,
frameSize)

    fftCurrent = abs(fft(xDataFrame));

    if (frame == 1)
        param.fftShort = fftCurrent;
        param.fftLong = fftCurrent;
    end

    fftCurrentMed = medfilt1(fftCurrent,5);

    param.fftShort =
max(fftCurrentMed, (param.fftShort*(param.fftShortAvgCnt-1) +
fftCurrentMed)/param.fftShortAvgCnt);

    param.fftLong = ((param.fftLong*(param.fftLongAvgCnt-1) +
fftCurrentMed)/param.fftLongAvgCnt); %Original
% param.fftLong = growingFunc(param.fftLong,fftCurrentMed,
param.fftLongAvgCnt, 0, 0); %Original

    for ei=1:numEqBands
        if (ei == 1)
            indexStart = 1;
            spaceToEnd = round(midFrameFreq(ei+1)) - round(midFrameFreq(ei));
            indexEnd = round(midFrameFreq(ei) + spaceToEnd/2);
        else
            if (ei == numEqBands)
                spaceToStart = round(midFrameFreq(ei)) -
round(midFrameFreq(ei-1));
                indexStart = round(midFrameFreq(ei) - spaceToStart/2)+1;
                indexEnd = frameSize;
            else
                spaceToStart = round(midFrameFreq(ei)) -
round(midFrameFreq(ei-1));
                indexStart = round(midFrameFreq(ei) - spaceToStart/2)+1;

                spaceToEnd = round(midFrameFreq(ei+1)) -
round(midFrameFreq(ei));
                indexEnd = round(midFrameFreq(ei) + spaceToEnd/2);
            end
        end
        fftPerFreqShort(:,ei) =[
gmean(param.fftShort(indexStart:indexEnd,1),2)
gmean(param.fftShort(indexStart:indexEnd,2),2)];
        fftPerFreqLong(:,ei) =[ gmean(param.fftLong(indexStart:indexEnd,1),2)
gmean(param.fftLong(indexStart:indexEnd,2),2)];

        diffProfilePerFreq(:,ei) = abs(fftPerFreqLong(:,ei)-
fftPerFreqShort(:,ei));
    end

    dat = max(fftPerFreqLong) - (fftPerFreqShort);
    dat = dat ./ param.datDiv;
    dat(dat < 1) = 1;
    diffProfilePerFreq = diffProfilePerFreq .* (dat)./(
((lowEnergiesHasMorePower)) .* frequencyUnimportanceWeight(1);

```

```

    threshold = (median(diffProfilePerFreq'))'./param.thresholdDownWeight;
    param.threshold = max ( (param.threshold*(param.fftLongAvgCntShrt-1) +
threshold) ./param.fftLongAvgCntShrt , ...
    (param.threshold*param.thresholdAvgCnt +
threshold) ./ (param.thresholdAvgCnt+1));

    diffProfilePerFreq = diffProfilePerFreq - param.threshold;
    diffProfilePerFreq(diffProfilePerFreq>0) = 0;

    diffProfilePerFreq =
pullDownZeroNeighbours (diffProfilePerFreq,max(max(abs(diffProfilePerFreq)))/10);

    diffProfilePerFreq = abs(diffProfilePerFreq).*abs(diffProfilePerFreq);
    diffProfilePerFreq = normalizeData(diffProfilePerFreq, 0,
targetDbToLower, 1000);

    monotonicFreqs = abs(fftPerFreqLong ./ fftPerFreqShort); % ./
((lowEnergiesHasMorePower)) .* frequencyUnimportanceWeight(1);
    monotonicFreqs(isnan(monotonicFreqs)) = 10;
    monotonicFreqs = abs(monotonicFreqs-1);

    minOfMon = min(param.maxOfMon, max(param.minOfMon, min(monotonicFreqs) +
param.monPlus));
    monotonicFreqs(monotonicFreqs > minOfMon) = 0;

    monotonicFreqs =
pullDownZeroNeighbours(monotonicFreqs,min(minOfMon)/param.monPullDownNeighboursTo
Div);
    monotonicFreqs = monotonicFreqs * targetDbToLower*param.monIncrease;

    param.monotonicFreqs = min(monotonicFreqs, (param.monotonicFreqs*(15-1) +
monotonicFreqs)/15);
    param.diffProfilePerFreq = min
(diffProfilePerFreq, (param.diffProfilePerFreq*(param.diffProfilePerFreqAvgCnt-1)
+ diffProfilePerFreq)/param.diffProfilePerFreqAvgCnt);

    currPeakGains = param.monotonicFreqs + param.diffProfilePerFreq;
    currPeakGains = currPeakGains./param.peakGainsDiv.*-1;

    param.peakGains = (param.peakGains*(param.peakGainsAvgCnt-1) +
currPeakGains)/param.peakGainsAvgCnt;

    param.freqsMin = min (param.peakGains, (param.freqsMin*(param.freqsMinCnt-
1) + param.peakGains)/param.freqsMinCnt); %max filter
    param.freqsMin(param.freqsMin > 0) = 0;
    curFilter = param.peakGains .* param.mostImportantFreqsForADi;
    curFilter(curFilter > 0) = 0;
    EDiVal = abs(curFilter) - abs(param.freqsMin) .*
param.mostImportantFreqsForADi;
    EDiVal = EDiVal .* EDiVal;
    EDiVal = abs(mean(mean(EDiVal)));

    param.ADiMax = max(EDiVal, (param.ADiMax*(param.ADiMaxCnt-1) +
EDiVal)/param.ADiMaxCnt);

    if (EDiVal == 0)
        xDataFrame(:) = 0;
        param.ADiAnswMin = 0;
    else

```

```

        final = (EDiVal/param.ADiMax - param.ADiAnswMin*param.ADiWeightMin -
param.ADiWeightMinus) * (param.ADiWeightOne + param.ADiAnswMin+
param.ADiWeightPlus);
        param.ADiAnswMin = min
(EDiVal/param.ADiMax, (param.ADiAnswMin*param.ADiAnswMinCnt+EDiVal/param.ADiMax)/(
param.ADiAnswMinCnt+1));
        ADi(:) = max(0,final);

end

if (frame == 5)
    param.ADiMax = 0;
    param.ADiAnswMin = 5;
end

%INITIALIZATION
if (frame == 1)
    param.diffProfilePerFreq = diffProfilePerFreq;
end

param = fixParams(param);

MPEleft.PeakGains = round(param.peakGains(1,:)*1000)/1000;
MPEleft.QualityFactors = qualityFactors + abs(param.peakGains(1,:))/15;

MPERight.PeakGains = round(param.peakGains(2,:)*1000)/1000;
MPERight.QualityFactors = qualityFactors + abs(param.peakGains(2,:))/15;

MPEleft.PeakGains(MPEleft.PeakGains>0) = 0;
MPERight.PeakGains(MPERight.PeakGains>0) = 0;

MPEleft = fixMpe(MPEleft);
MPERight = fixMpe(MPERight);

[xDataFrame] = [ MPEleft(xDataFrame(:,1)) MPERight(xDataFrame(:,2)) ];
end
end

```