



OULUN YLIOPISTO  
UNIVERSITY of OULU

TIETOTEKNIIKAN OSASTO

**Markus Särkiniemi**  
**Ilari Iso-Junno**  
**Tapio Kinnunen**

**FASTICA-ALGORITMIN EROTTELUKYKY  
ÄÄNIMAISEMA-ANALYYSISSÄ**

Kandidaatintyö  
Tietotekniikan koulutusohjelma  
toukokuu 2015

Särkiniemi M., Iso-Junno I., Kinnunen T. (2015) FastICA-algoritmin erottelukyky äänimaisema-analyysissä. Oulun yliopisto, Tietotekniikan osasto. Kandidaatintyö, 25 s.

## TIIVISTELMÄ

Ihmisen kuuloaisti kykenee erottelemaan ympäristöstään yksittäisiä ääniä erinomaisella tarkkuudella. Sitä on pyritty jäljittelemään monilla erilaisilla koneellisilla menetelmillä, mutta yksikään niistä ei ole saavuttanut ihmisen kuuloaistin erottelukykyä. Näistä menetelmistä hyödyllisimpiä ovat sokeat signaalin erottelumenetelmät (BSS), joissa äänilähteistä ei tiedetä mitään ennakkoon. Tässä työssä äänilähteiden erotteluun käytetään riippumattomien komponenttien analyysiä (ICA), joka on tehokas BSS-menetelmä. Tällä hetkellä ICA:lla kyetään hyvin erottelemaan äänilähteitä, mutta sen rajoituksia ovat muun muassa mikrofoniin lukumäärä, äänilähteiden tilastollinen riippumattomuus, äänilähteiden liikkumattomuus sekä permutaatio ja skaalaus-ongelma.

Tässä työssä kehitetyn ohjelmiston toteutus pohjautuu valmiiseen FastICA-algoritmiin, joka on tällä hetkellä yksi nopeimmista olemassa olevista ICA-algoritmeista. FastICA:n erottelukykyä arvioidaan koherenssilla 2 - 19 äänilähteen erottelussa. Tulokset osoittavat, että FastICA:n erottelukyky heikkenee äänilähteiden lukumäärän kasvaessa ja samankaltaiset äänilähteet eivät erotu hyvin. Lisäksi matalataajuiset signaalit osoittautuivat huonosti erottuviksi. Erinomaiseen erottelutulokseen päästään alle kahdeksalla äänellä, jolloin yli 95 % äänistä erottuu hyvin. Jatkotutkimuksessa tulisi keskittyä lisämenetelmiin, joilla voisi paikata ICA:n rajoituksia.

Avainsanat: äänimaisema-analyysi, sokea signaalin erottelu, cocktailkutsuilmiö, ICA, FastICA

**Särkiniemi M., Iso-Junno I., Kinnunen T. (2015) The separation performance of the FastICA algorithm in auditory scene analysis.** University of Oulu, Department of Computer Science and Engineering. Bachelor's thesis, 25 p.

## **ABSTRACT**

**The human sense of hearing is able to separate individual sounds in the environment with extraordinary accuracy. Several attempts have been made to imitate it artificially but so far none of these methods have achieved the performance of the human sense of hearing. The most useful of these methods is blind signal separation (BSS) since it does not need any preliminary information on the sound sources. This work used independent component analysis (ICA) for separating sound sources. ICA is an efficient blind signal separation technique. At the moment, ICA is able to separate sound sources well but it has limitations such as the amount of microphones used, statistical independence between sound sources, immobility of sound sources and the permutation and scaling problem.**

**The software developed for this Bachelor's thesis is based on the FastICA algorithm, which is currently one of the fastest ICA algorithms in existence. The accuracy of FastICA is measured by calculating the coherence in the separation of 2 to 19 sound sources. The results indicate that the separation performance of FastICA decreases as the number of sources increases and similar sound sources are not separated all that well. Additionally, low frequency signals are separated poorly. The separation performance is excellent with less than eight sound sources and over 95 % of sounds will separate well. In conclusion, it can be said that further research is needed in order to develop methods that deal with the limitations of ICA.**

**Key words: Auditory Scene Analysis, Blind Signal Separation, Cocktail Party Problem, ICA, FastICA**

# SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

SISÄLLYSLUETTELO

ALKULAUSE

LYHENTEIDEN JA MERKKIEN SELITYKSET

1.	JOHDANTO.....	7
2.	TAUSTA .....	8
2.1.	Mitä ääni on? .....	8
2.2.	Ihmisen kuuloaisti .....	8
2.2.1.	Haas-ilmio .....	9
2.2.2.	Cocktailkutsuilmio .....	9
2.3.	Äänimaisema-analyysi .....	10
2.4.	Laskennallinen äänimaisema-analyysi .....	10
2.5.	Sokea signaalin erottelu.....	11
2.5.1.	Riippumattomien komponenttien analyysi.....	11
2.5.2.	Pääkomponenttianalyysi.....	12
2.6.	FastICA-algoritmi .....	12
2.6.1.	FastICA:n matemaattinen malli .....	12
2.7.	Diskreetti ristikorrelaatio.....	13
2.8.	Koherenssi .....	14
3.	TOTEUTUKSEN KUVAUS .....	15
3.1.	Mikrofonihavaintojen luominen ohjelmallisesti .....	15
4.	MITTAUKSET JA TULOKSET .....	17
4.1.	Testaussuunnitelma .....	17
4.2.	Tulokset .....	17
5.	PROJEKTIN KUVAUS .....	21
5.1.	Ajankäyttö .....	21
6.	JATKOKEHITYS .....	22
7.	YHTEENVETO .....	23
8.	LÄHTEET .....	24

## ALKULAUSE

Tämä kandidaatintyö tehtiin kolmen hengen ryhmässä osana Tietotekniikan osaston kurssia 521275A Sulautettujen ohjelmistojen projekti. Projektissa käytetty laitteisto saatiin lainaan yliopistolta, ja ohjelmistoa työstettiin ja testattiin pääasiassa yliopiston tiloissa. Ryhmän jäseniä olivat Markus Särkiniemi, Ilari Iso-Junno ja Tapio Kinnunen. Erityiset kiitokset korvaamattomasta avusta annetakoon Teemu Tokolalle, joka toimi kurssin ja siten myös tämän työn ohjaajana. Haluamme lisäksi osoittaa kiitokset kurssin muille osallistujille viihtyisästä työskentely- ja opiskeluilmapiiristä.

Oulu, toukokuun 15. päivänä 2015

Markus Särkiniemi  
Ilari Iso-Junno  
Tapio Kinnunen

## LYHENTEIDEN JA MERKKIEN SELITYKSET

ASA	äänimaisema-analyysi
BSS	sokea signaalin erottelu
CASA	laskennallinen äänimaisema-analyysi
IBM	ideaalinen binäärimaski
ICA	riippumattomien komponenttien analyysi
PC	pöytätietokone
PCA	pääkomponenttianalyysi
PCM	pulssikoodimodulaatio
SVD	singulaariarvohajotelma
T-F	aika - taajuus
wav	äänitiedostomuoto
<b>A</b>	sekoitusmatriisi
<b>S</b>	alkuperäisten äänien matriisi
<b>W</b>	estimoitu sekoitusmatriisin käänteismatriisi
<b>X</b>	havaintomatriisi
<i>K</i>	mikrofonien lukumäärä
<i>M</i>	näytteiden lukumäärä
<i>N</i>	äänien lukumäärä
<i>f, g</i>	diskreetti reaaliarvoinen funktio
<i>o</i>	alkuperäinen ääni
$\hat{o}$	alkuperäisen äänen vastaääni
$\ddot{a}$	eroteltu ääni
$n(t)$	kohina ajan hetkellä $t$
$s(t)$	äänilähteiden vektori ajan hetkellä $t$
$x(t)$	havaintosignaali ajan hetkellä $t$
<i>dB</i>	desibeli
<i>h</i>	tunti
<i>Hz</i>	hertsi
*	konvoluutio

## 1. JOHDANTO

Ihmiset altistuvat elämässään jatkuvasti erilaisille äänille, kuten linnun laululle, katuesiintyjän soitolle tai lapsen naurulle. Eri äänilähteistä lähtevät äänet, eli paineaallot, yhdistyvät keskenään kulkiessaan ilman halki. Ne saapuvat ihmisen korvaan yhtenä summa-aaltona, mutta ihmisen kuuloaisti kykenee erottelemaan yksittäiset äänilähteet toisistaan ja tunnistamaan ne. Tätä ihmisen kuuloaistin ominaisuutta kutsutaan äänimaisema-analyysiksi (Auditory Scene Analysis, ASA). [1]

Äänimaisema-analyysiä on jäljitelty koneellisesti. Sen mahdollisia sovelluskohteita ovat muun muassa kuulolaitteet sekä erinäiset valvonta- ja vakoilujärjestelmät [2]. Koneellinen äänimaisema-analyysi on vaikea toteuttaa käytännössä, joten siihen liittyvän tutkimuksen tähänastiset saavutukset eivät ole saavuttaneet ihmisen kuuloaistin suorituskykyä [2, 3]. Tämä tekee aiheen lisätutkimisesta erityisen tarpeellista sekä varsin haasteellista.

Työn tarkoituksena on tutkia yhtä koneellista äänimaisema-analyysimenetelmää, FastICA:ta, ja selvittää erityisesti kuinka äänilähteiden lukumäärä vaikuttaa sen erottelukykyyn.

## 2. TAUSTA

Ääni on väliaineessa etenevää painevaihtelua. Ihminen havaitsee kuuloaistillaan ilmassa tapahtuvat painevaihtelut ääniksi. Kuuloaistimus alkaa ihmisen tärykalvosta, josta se kuuloluiden, simpukan ja värekarvojen kautta etenee sähköisenä signaalina aivoihin. Tuo sähköinen signaali koostuu usein äänten sekamelskasta, josta ihminen kykenee erottamaan eri äänilähteitä ja keskittymään juuri halumaansa lähteeseen. Tällaista äänten erottelukykyä kutsutaan cocktailkutsuilmiöksi. Ihmisen laajempaa äänien havainnoinnin mallia kutsutaan puolestaan äänimaisema-analyysiksi (ASA), jota pyritään mallintamaan tietoteknisillä laitteilla. Ihmismallia matkivia menetelmiä kutsutaan termillä laskennallinen äänimaisema-analyysi (CASA).

Eri äänilähteiden erotteluun on pyritty keksimään myös keinotekoisia menetelmiä. Sokeassa signaalin erottelussa (BSS) lähtöoletus on, ettei käsiteltävästä äänilähteestä saada ylimääräistä tietoa mikrofoniin antamien analogisten signaalien lisäksi. BSS-pohjaisista ratkaisuista riippumattomien komponenttien analyysimenetelmät (ICA) keskittyvät äänisignaalien tilastollisiin ominaisuuksiin ja suoriutuvat erottelusta kohtuullisen tehokkaasti äänilähteiden ollessa tilastollisesti riippumattomia. Pääkomponenttianalyysillä (PCA) vähennetään datan korrelaatiota ja päästään datan ei-toivotuista komponenteista eroon. PCA-tekniikkaa käytetään usein datan esikäsittelyyn ICA-menetelmälle.

### 2.1. Mitä ääni on?

Ääni syntyy kappaleen värähtelystä. Äänilähteen värähtely aiheuttaa väliaineen, kuten ilman, molekyyliessä vuorottain tihentymiä ja harventumia. Näitä kutsutaan äänialloiksi. Ääni etenee, kun väliaineen molekyylit värähtelevät edestakaisin törmäillen toisiinsa ja synnyttäen näin eräänlaisen ketjureaktion. [4]

Ääni on siis lyhyesti ilmaistuna painevärähtelyä, joka etenee aaltoliikkeenä väliaineessa. Äänenkorkeutta mitataan taajuudella eli painevärähdysten lukumäärällä aikayksikössä. Mittayksikkönä taajuudelle käytetään hertsiä (Hz), jonka yksikkö on 1/s. Yksi hertsi tarkoittaa yhtä värähdystä sekunnissa. Ihminen aistii pienen taajuuden matalana äänenä ja suuren taajuuden korkeana äänenä. Äänenvoimakkuus puolestaan määräytyy staattisen ilmanpaineen ja äänenpaineen erosta. Ihmisen havainnossa ääni on voimakkaampi tämän eron ollessa suurempi. [5]

### 2.2. Ihmisen kuuloaisti

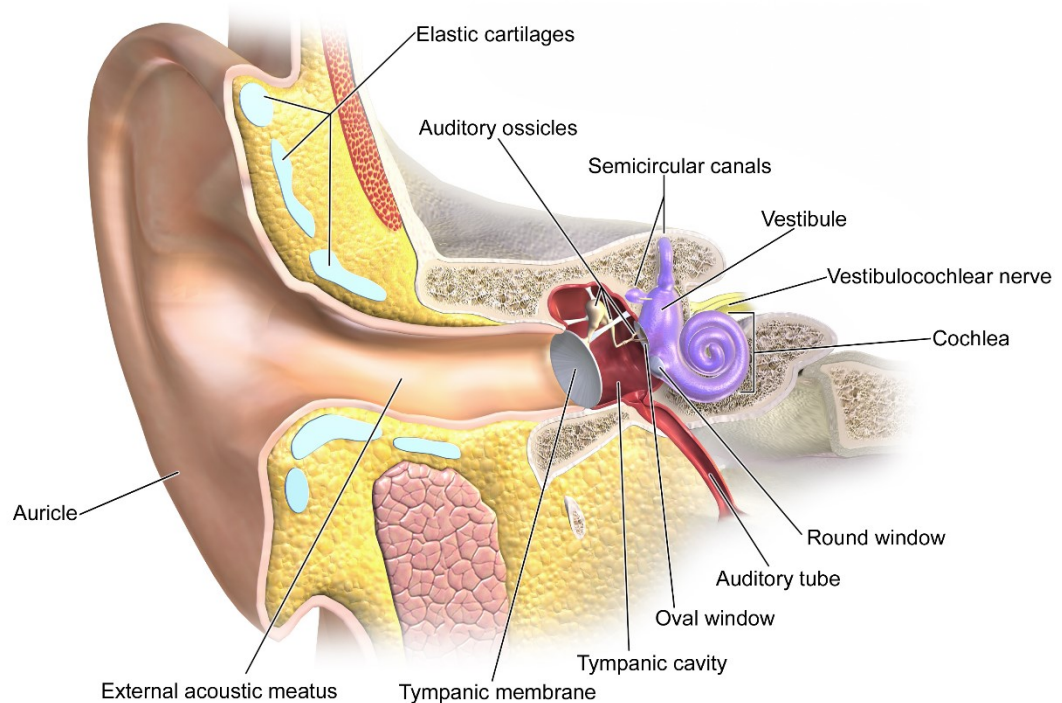
Ihmisen kuulema ääni on korvan aistimaa ilmanpaineen vaihtelua. Kuuloaistin havaittavissa oleva taajuusalue on noin 20 - 20 000 Hz. Ihmisen kuulo aistii äänenvoimakkuuden kutakuinkin logaritmisesti, mistä syystä äänenvoimakkuutta kuvataan desibeleillä (dB), joka on logaritminen suure. Esimerkiksi äänenvoimakkuus aistitaan kaksinkertaisena, kun se on kasvanut 10 dB. [5]

Paineaallot ohjautuvat korvanlehdessä korvakäytävään, jonka päässä on tärykalvo. Paineaallot saavat tärykalvon värähtelemään, minkä seurauksena myös kuuloluut alkavat värähdellä. Kuuloluut vahvistavat värähtelyä ja siirtävät sen simpukkaan, jonka sisällä paineaallot etenevät nesteessä. Simpukassa sijaitsevat aistinsolut, joiden värekarvat alkavat värähdellä paineaaltojen seurauksena. Tämä synnyttää soluissa



sähköisiä signaaleja, jotka kulkeutuvat aivojen kuulokeskukseen kuulohermoä pitkin. [6]

Kuvassa 1 esitetään korvan rakenne tarkemmin.



Kuva 1. Korvan rakenne.<sup>1</sup>

### 2.2.1. Haas-ilmiö

Koska ihmisellä on kaksi korvaa, ääni saapuu niihin hieman eri aikaan, ja toinen korva kuulee äänen voimakkaampana. Näiden tietojen avulla aivot kykenevät päättelemään äänen tulosuunnan. Tätä ominaisuutta kutsutaan Haas-ilmiöksi. [6]

### 2.2.2. Cocktailkutsuilmiö

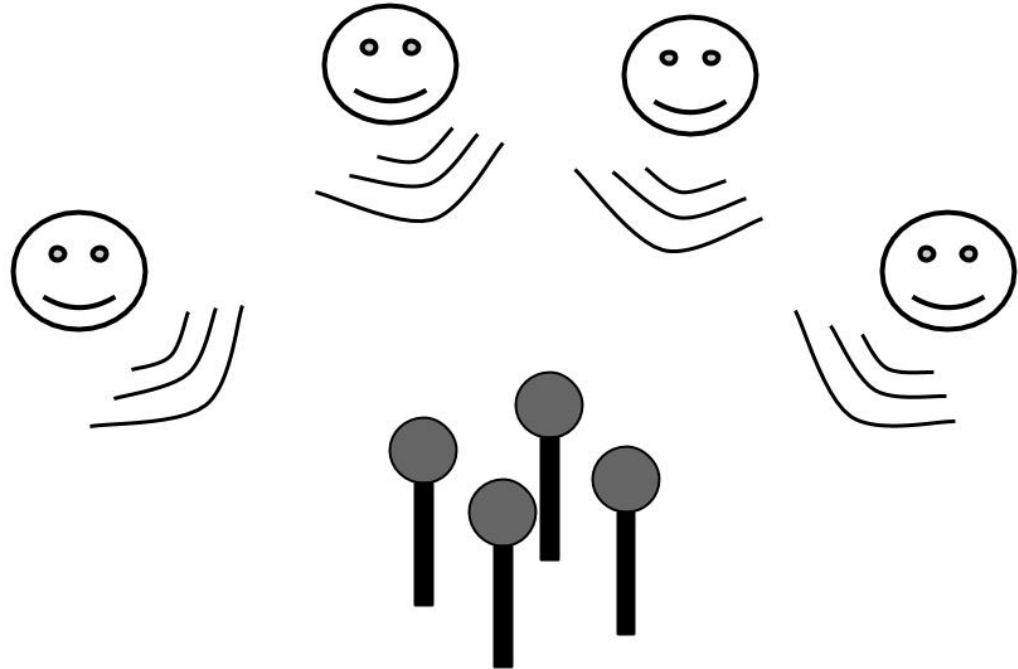
Cocktailkutsuilmiöllä (Cocktail Party Problem) tarkoitetaan ihmisen kykyä erottaa tietty äänilähde, esimerkiksi toisen ihmisen puhe, muiden häiritsevien äänilähteiden joukosta. Cocktailkutsuilmiön esitteli ensimmäisenä Colin Cherry 1950-luvulla. [7] Tässä työssä käsitellään cocktailkutsuilmiötä tietoteknisestä näkökulmasta.

Jos  $K$  mikrofonia tallentaa  $N$  erilaista ääntä yhtä aikaa, äänite on matriisi, joka koostuu  $K$  määrästä vektoreita. Niistä jokainen on painotettu lineaarinen  $N$  äänen summa. Otetaan  $M$  määrä näytteitä, jolloin voidaan ilmaista lähdematriisi  $\mathbf{S} = N \times M$  ja havaintomatriisi  $\mathbf{X} = K \times M$ . Nyt  $\mathbf{X}^T$  voidaan laskea sekoitusmatriisiin  $\mathbf{A} = K \times N$  ja  $\mathbf{S}^T$  avulla, joten:

<sup>1</sup> Blausen.com staff. "Blausen gallery 2014". Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762.

$$\mathbf{X}^T = \mathbf{A}\mathbf{S}^T \quad (1)$$

[8] Tätä havainnollistetaan kuvassa 2.



Kuva 2. Havaintomatriisi X luodaan nauhoittamalla päällekkäisiä puheääniä.

### 2.3. Äänimaisema-analyysi

Bregmanin mukaan kuunnellessaan ihminen suorittaa äänimaisema-analyysin (ASA). Hänen mukaansa tämä on kaksivaiheinen prosessi, jonka ensimmäisessä vaiheessa kuultu äänien sekoitus jaotellaan elementteihin. Elementti kuvailee merkittävää akustista tapahtumaa, ja yksi elementti on pieni osa koko kuulemistapahtumaa. Toisessa vaiheessa ne elementit, jotka arvellaan saapuneen samasta akustisesta lähteestä, ryhmitellään omiksi äänivirroiksi. Akustisena äänilähteenä toimivat esimerkiksi ihmispuhujan äänielimet, ja kuulemistapahtumassa puhujan äänestä muodostuu aivoihin äänivirta. Elementtien ryhmittely äänivirroiksi voi olla tieto- tai mallipohjainen. [1]

### 2.4. Laskennallinen äänimaisema-analyysi

Laskennallinen äänimaisema-analyysi (CASA) on tutkimusalue, joka pyrkii toteuttamaan ihmiskuulojärjestelmää keinotekoisesti. Osa CASA-järjestelmistä mallintaa hyvinkin tarkasti tiedettyjä ihmisen kuulojärjestelmän prosesseja, kun taas osassa käytetään lisänä myös paremmin toimivia menetelmiä. Kaikissa CASA-järjestelmissä on pohjalla ihmiskuulijan havainnointikyky ja kuulojärjestelmä. Se erottaa CASA-menetelmät muista äänenerottelumenetelmistä. [9]

Kuten ASA-teoriassa, myös CASA:ssa suoritetaan puheen erottelu tyypillisesti kahdessa osassa: paloittelemalla ja kokoamalla. Saatuaan sisääntulosignaalin, CASA-järjestelmä jakaa signaalin ensin eri taajuuskanaville käyttäen gammaäänisuodatinpankkia. Tämän jälkeen jaotellut signaalit paloitellaan vielä ajallisesti osiin. Tuloksena saadaan esitys, jota kutsutaan simpukkagrammiksi (cochleagram). Jaotteluvaiheessa sisääntulosignaali hajotetaan aika-taajuus (T-F)-segmentteihin. Ryhmittelyvaihe yhdistää ne segmentit, jotka ovat todennäköisesti saapuneet samasta äänilähteestä, äänivirraksi. [10]

CASA-menetelmien käyttöön pyritään luomaan ideaalinen binäärimaski (IBM). IBM on kaksiulotteinen binäärimatriisi, jonka jokainen alkio vastaa yhtä T-F-segmenttiä. IBM luodaan jokaiselle äänilähteelle erikseen. IBM:n luonnissa merkitään ykkösellä äänilähteelle kuuluviksi arvioidut T-F-segmentit ja nolllalla äänilähteelle kuulumattomat T-F-segmentit. Kun maskia käytetään, nolllalla merkityt T-F-segmentit poistetaan täysin äänien sekoituksesta ja ykkösellä merkityt jätetään. Ideaalitulanteessa jäljelle jäävät T-F-segmentit sisältävät vain yhden äänilähteen aiheuttamaa ääntä.

## 2.5. Sokea signaalin erottelu

Sokealla signaalin erottelulla (Blind Signal Separation, BSS) tarkoitetaan menetelmää, jolla voidaan erottaa eri äänilähteet toisistaan yhdestä tai useammasta havaitusta sekoittuneesta signaalista [11]. Cocktailkutsuilmiö on tyypillinen esimerkki sokean signaalin erottelusta [8].

Termillä sokea viitataan menetelmän tietämättömyyteen: ei tiedetä kuinka signaalit ovat sekoittuneet keskenään [11, 12], ja ainoa tieto tulee sekoittuneesta signaalista itsestään [12, 13]. Sekoittunutta signaalia  $x(t)$  voidaan useissa käytännön tilanteissa mallintaa kaavalla

$$x(t) = \mathbf{A}s(t) + n(t) \quad (2)$$

jossa matriisi  $\mathbf{A}$  on siirtofunktio äänilähteiden ja havainnoidun signaalin välillä,  $s(t)$  on äänilähteiden vektori ja  $n(t)$  kuvaa kohinaa ajan hetkellä  $t$ . BSS:llä pyritään estimoimaan matriisi  $\mathbf{A}$  ja palauttamaan äänilähteet  $s(t)$  sekoittuneesta signaalista  $x(t)$ . [13]

### 2.5.1. Riippumattomien komponenttien analyysi

Riippumattomien komponenttien analyysi (Independent Component Analysis, ICA) on yleisnimitys menetelmille, joilla saadaan selville toisistaan riippumattomat äänilähteet yhdestä tai useammasta sekoittuneesta signaalista. Siinä oletetaan äänilähteiden olevan lineaarisesti riippumattomia, ei-Gaussisia ja akselit eivät ole välttämättä ortogonaalisia. ICA on iteratiivinen erottelumenetelmä. ICA:ssa iteratiiviset menetelmät perustuvat tilastollisiin ominaisuuksiin, esimerkiksi huipukkuutteen (tiheysfunktion muodon mittari), entropiaan tai yhteiseen informaatioon sekoittuneesta signaalista. Alla olevassa listassa on erilaisiin ominaisuuksiin perustuvia ICA:n toteutuksia [8]:

- suurimman uskottavuuden estimointi

- tiedonsiirron maksimointi
- negatiivisen entropian maksimointi
- ei-lineaarinen pääkomponenttianalyysi

Ideana on löytää matriisin  $W$  estimaattori, joka on alkuperäisen sekoitusmatriisin  $A$  käänteismatriisi [8, 14 - 16]. ICA on todettu tehokkaaksi menetelmäksi sokean signaalin erottelussa [15, 17], mutta siinä on myös muutamia ongelmia:

- äänilähteiden pitää olla tilastollisesti riippumattomia [8]
- permutaatio ja skaalaus -ongelma (The permutation and scaling problem) [8]
- lähteiden oletetaan olevan paikallaan (Stationary mixing model) [8]

Kaikesta huolimatta ICA on parempi kuin PCA, koska korreloimattomuus ei yksinään riitä tuntemattomien äänilähteiden erottamiseen [15]. Kun äänilähteitä on enemmän kuin mikrofoneja, ICA:lla saadaan huonoja erottelutuloksia [18].

### 2.5.2. Pääkomponenttianalyysi

Pääkomponenttianalyysi (Principal Component Analysis, PCA) on tunnettu menetelmä datan käsittelyssä [19]. Sen ideana on määrittellä uudet muuttujat, jotka ovat lineaarisia yhdistelmiä kaikista alkuperäisistä muuttujista. Tavoitteena on vähentää datan korrelaatiota tekemällä ortogonaalinen projektiio. Yleensä dimensiota vähennetään, koska tällöin päästään eroon ei-toivotuista komponenteista [8]. Yleinen PCA-toteutus on singulaariarvohajotelma (singular value decomposition, SVD) [8, 20].

## 2.6. FastICA-algoritmi

Tämä työ pohjautuu tehokkaaseen FastICA-algoritmiin, jonka on kehittänyt Helsingin yliopiston tietojenkäsittelytieteen professori Aapo Hyvärinen. Algoritmi on saatavilla valmiina ohjelmistopaketteina eri ohjelmointikielillä [21]. FastICA-algoritmi käyttää kiinteän pisteen iterointimenetelmää, joka on 10 - 100 kertaa nopeampi kuin ICA:ssa perinteisesti käytetty gradienttilaskeutumismenetelmä [22].

### 2.6.1. FastICA:n matemaattinen malli

ICA:n täsmälliseen määrittelyyn tarvitaan piilevien muuttujien mallia. Oletetaan, että tarkkailemme  $n$ :ää lineaarista sekoitusta  $x_1, \dots, x_n$   $n$ :stä riippumattomasta komponentista:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \dots + a_{jn}s_n, \text{ kaikilla } j \quad (3)$$

Aikaindeksi  $t$  puuttuu kokonaan. ICA-mallissa oletetaan, että jokainen sekoitelma  $x_j$ , kuten myös jokainen riippumaton komponentti  $s_k$ , on satunnainen muuttuja kunnollisen aikasignaalin sijasta. Tarkkaillut arvot  $x_j(t)$  ovat näytteitä tästä satunnaisesta muuttujasta. Esimerkiksi cocktailkutsuilmiön mikrofonisignaalit ovat

kyseisiä näytteitä. Ilman turhaa yleistystä voidaan olettaa, että sekä sekoittuneilla muuttujilla että riippumattomilla komponenteilla on nollakeskiarvo. Jos näin ei ole, tarkkaillut muuttujat  $x_i$  voidaan aina keskittää vähentämällä näytteen keskiarvo, jolloin näytteestä saadaan nollakeskiarvoisia.

On kätevää käyttää ilmaisuun matriisimallia. Yleisesti kursiivilla merkityt pienet kirjaimet tarkoittavat vektoria ja kursiivilla olevat isot kirjaimet matriisia. Kaikki vektorit tulee ymmärtää pystyvektoreina. Merkitään  $x$ :llä satunnaista vektoria, jonka elementtinä ovat sekoitukset  $x_1, \dots, x_n$  ja vastaavasti  $s$ :llä satunnaista vektoria elementeillä  $s_1, \dots, s_n$ . Merkitään sekoitusmatriisia  $A$ :lla, jonka elementit ovat  $a_{ij}$ . Näillä merkinnöillä voidaan sekoitusmalli esittää muodossa:

$$x = As \quad (4)$$

Jotta voidaan viitata sekoitusmatriisiin  $A$  pystyvektoreihin, malli voidaan kirjoittaa myös muodossa:

$$x = \sum_{i=1}^n a_i s_i \quad (5)$$

ICA-malli on produktiivinen malli, joka kuvailee kuinka havaittu data syntyy komponenttien  $s_i$  sekoittuessa. Riippumattomat komponentit ovat niin sanottuja piileviä muuttujia, eikä niitä voida suoraan tarkkailla. Sekoitusmatriisi  $A$  on myös tuntematon. ICA-mallissa tarkkaillaan vektoria  $x$ , josta matriisi  $A$  ja vektori  $s$  estimoidaan.

ICA-malli pohjautuu tilastollisuuteen. Perusoletus on, että komponentit  $s_i$  ovat tilastollisesti riippumattomia. Toinen tarvittava oletus on, että riippumattomien komponenttien tulee olla ei-Gaussisia. Sekoitusmatriisista  $A$  voidaan laskea sen käänteismatriisi, jota merkitään  $W$ :llä ja riippumaton komponentti saadaan kaavasta:

$$s = Wx \quad (6)$$

ICA on yksi laajimmin käytetyistä BSS-metodeista. Malliin voidaan sisällyttää myös kohinatermi  $n$ . [23]

## 2.7. Diskreetti ristikorrelaatio

Signaalinkäsittelyssä ristikorrelaatiota voidaan käyttää mittarina kahden signaalin väliselle samankaltaisuudelle. Samankaltaisten signaalien ristikorrelaation huippukohta on huomattavasti korkeampi kuin erilaisten signaalien. Siksi ristikorrelaation avulla voidaan esimerkiksi löytää erotellulle äänisignaalille pari alkuperäisten äänisignaalien joukosta. Ristikorrelaatiota kutsutaan myös nimellä liukuva pistetulo. Diskreeteille reaaliarvoisille funktioille  $f$  ja  $g$  ristikorrelaatio määritellään [24]:

$$(f \star g)_n = \sum_{m=-\infty}^{\infty} f_m g_{n+m} \quad (7)$$

## 2.8. Koherenssi

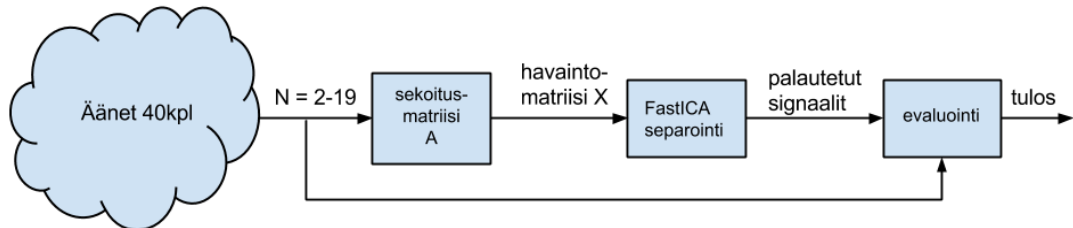
Koherenssin avulla voidaan ilmaista kahden signaalin lineaarista riippuvuutta. Koherenssi perustuu samantaajuisten komponenttien vertailuun, jota varten aikatason signaali muunnetaan taajuustasoon käyttämällä Fourier'n muunnosta. Kun vertailtavat taajuuskomponentit vastaavat täysin toisiaan, koherenssi on arvoltaan yksi. Jos niillä ei puolestaan ole mitään suhdetta, koherenssi on nolla. Koherenssi on suuri, kun vertailtavalla taajuudella signaalin vaihe pysyy kiinteässä suhteessa toisen signaalin vaiheeseen. Koherenssi on puolestaan pieni, kun vertailtavalla taajuudella signaalin vaihe muuttuu suhteessa toisen signaalin vaiheeseen. Koherenssi on määritelty [25]:

$$C_{ij}(f) = \frac{|P_{ij}(f)|^2}{P_{ii}(f)P_{jj}(f)} \quad (8)$$

Kaavassa  $P_{ij}(f)$  on ristitehotiheyspektri  $i$ :n ja  $j$ :n välillä,  $P_{ii}(f)$  ja  $P_{jj}(f)$  ovat  $i$ :n ja  $j$ :n tehotiheyspektrit.  $C_{ij}(f)$  on koherenssin arvo taajuudella  $f$ .

### 3. TOTEUTUKSEN KUVAUS

Toteutettu algoritmi perustuu FastICA:han, ja toteutuksen rakenne näkyy kuvassa 3. Työn testiaineisto sisältää 40 eri ääntä. Niistä haluttu määrä,  $N$  kappaletta, sekoitetaan sekoitusmatriisiin  $A$  avulla. Näin saatu havaintomatriisi  $X$  syötetään FastICA-algoritmille, jonka tehtävä on erotella signaalit takaisin alkuperäisiä vastaaviksi. Evaluoinnissa hyödynnetään ristikorrelaatiota eroteltujen äänten ja alkuperäisten äänten parittamiseen. Pareja vertaillaan koherenssin avulla, ja tulokset tallennetaan. Testiohjelmistossa ei ole käyttöliittymää, koska se ei ole tarpeellinen.



Kuva 3. Toteutuksen rakenne.

Alla olevassa taulukossa 1 kuvataan testiäänten ominaisuudet. Äänet luetaan ja tallennetaan PCM-muodossa (pulsseekoodimodulaatio, Pulse Code Modulation), ja ne muutetaan 32-bittiseen liukulukumuotoon FastICA:n käsittelyä varten.

Taulukko 1. Testiäänten ominaisuudet

tiedostomuoto	wav
tallennusmuoto	16-bittinen PCM
käsittelymuoto	32-bittinen liukuluku
näytteenottotaajuus	44100 Hz
Kanavia	mono
Pituus	5 s
Määrä	40 kpl

#### 3.1. Mikrofonihavaintojen luominen ohjelmallisesti

Äänilähteiden sekoittaminen, eli mikrofonihavaintojen luominen, toteutetaan algoritmissa luomalla sopivan kokoinen sekoitusmatriisi  $A$ , jonka jokaisen alkion arvoksi arvotaan tasajakautuneita satunnaislukuja 0,5 ja 2,5 väliltä. Näiden raja-arvojen puitteissa ääni voi mikrofonihavainnossa olla maksimissaan viisi kertaa voimakkaampi kuin toinen. Mikrofonien havainnot luodaan ICA-mallin mukaisesti. Halutut sekoittuvat äänet koostetaan matriisiin  $S$ , jossa jokainen sarake vastaa yhtä äänilähdettä. Havaintomatriisissa  $X$  yksi sarake vastaa puolestaan yhtä mikrofonihavaintoa.  $X$  saadaan matriisiin  $S$  ja  $A$  pistetulosta. Sekoitusmatriisiin  $A$  yksi sarake määrää sekoitettavien äänilähteiden keskinäisen voimakkuuden yhteen mikrofonihavaintoon.  $A$ :n rivi määrää yhden äänilähteen voimakkuudet eri mikrofonihavaintoihin.

$$X = S \cdot A \quad (9)$$

Satunnaisuuden takia ohjelmallisesti luotu joukko mikrofonihavaintoja vastaa todellista äänitettyä mikrofonihavaintojen joukkoa vain hyvin harvoin. Reaalimaailman tilanteessa yksi ääni saapuu mikrofoneihin eri aikaan. Tässä oletetaan, että mikrofonit sijaitsevat hieman erillään toisistaan. Näin kyseinen ääni eri mikrofoneissa on viivästynyt. Ohjelmallinen mikrofonihavaintojen luominen ei toteuta viive-eroja, vaan äänilähteet ilmenevät samanaikaisesti kaikissa mikrofonihavainnoissa. Viive-erot jätettiin työn ulkopuolelle.



## 4. MITTAUKSET JA TULOKSET

Tässä työssä testataan FastICA:n erottelukykyä äänilähteiden erottelussa ja tutkitaan erityisesti äänilähteiden lukumäärän vaikutusta algoritmin erottelukykyyn.

Tuloksista havaitaan, että matalataajuiset äänet erottuvat huonosti. Samoin äänimäärän kasvaessa erottelutulos heikkenee.

### 4.1. Testaussuunnitelma

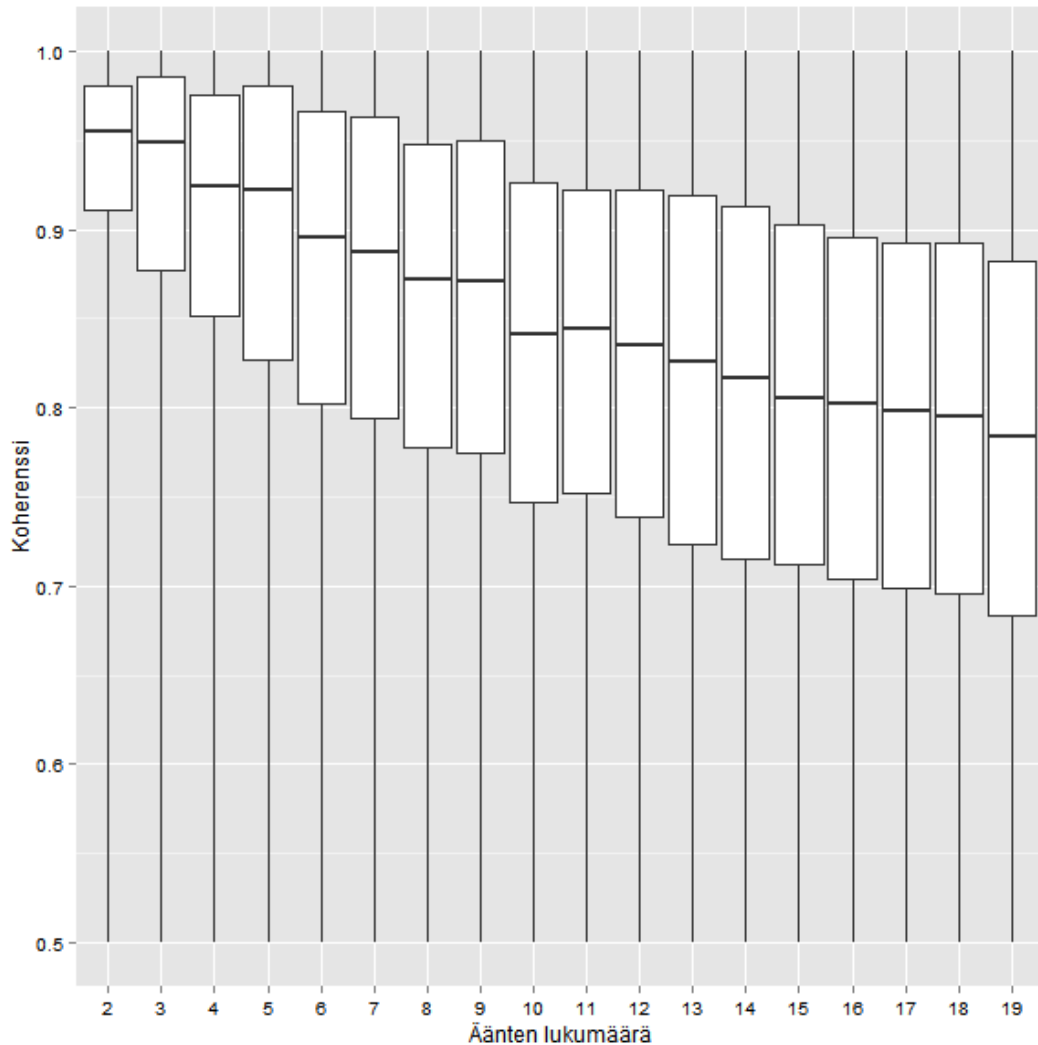
Testataan ICA-mallin mukaisesti ohjelmallisesti sekoitettujen ideaalisten äänten erottelua FastICA:lla. Yksittäinen sekoittamaton ääni on kohinaton, viiden sekunnin mittainen ja yksikanavainen (mono). Testissä äänilähteiden määrä vaihtelee 2 - 19 välillä ja algoritmille syötetään äänilähteiden määrä. Jokaisella äänimäärällä tehdään sata testiä, ja näin testiajoja on yhteensä 1800. Jokaisessa niistä käytetään satunnaisia ääniä 40 äänen joukosta. Äänten joukossa on mahdollisimman monipuolisesti erilaisia ääniä, kuten puhe-, soitin- ja luontoääniä. Jokaista algoritmilla eroteltua ääntä verrataan alkuperäiseen ääneen, ja samankaltaisuuden mittarina käytetään koherenssia. Eri äänimäärillä saatuja tuloksia vertaillaan, ja arvioidaan sitä, kuinka FastICA:n erottelukyky muuttuu äänimäärän kasvaessa.

Testiajossa kuunnellaan eroteltuja ääniä ja niiden alkuperäisiä vastineita, kun koherenssi on matala. Näin määritetään signaalille raja-arvo, jonka alle jäävät erotellut äänet tallennetaan kuunneltaviksi. Näin voidaan arvioida korvakuulolla koherenssin mukaan huonosti erottuneita ääniä.

### 4.2. Tulokset

Tuloksista koottiin kuva 4. Palkkien sisällä olevat poikkiviivat kuvaavat sadan testin koherenssien mediaanien keskiarvoa. Palkkien ylä- ja alareunat kuvaavat vastaavasti ensimmäisen ja kolmannen kvartiilin keskiarvoa.

Kuvasta nähdään, että erottelukyky heikkenee äänimäärän kasvaessa. Merkillepantavaa on, että mediaanien keskiarvo yhtä poikkeusta lukuun ottamatta laskee äänimäärän kasvaessa. Lisäksi erottelutarkkuuden hajonta kasvaa aluksi voimakkaasti äänimäärän kasvaessa noin kymmeneen, ja lopuksi heikommin äänimäärän lähestyessä 19:ää.



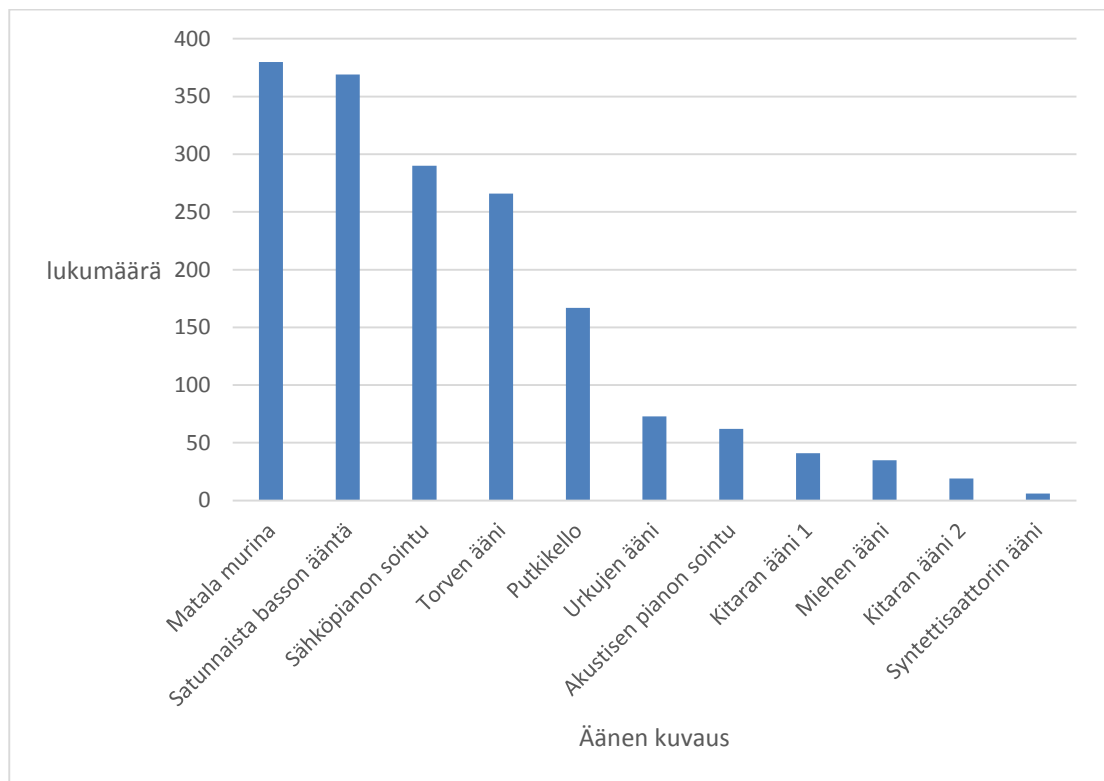
Kuva 4. FastICA:n erottelukyvyn tulokset satunnaisilla äänillä.

Raja-arvon määrittämiseksi kuunneltiin eroteltuja ääniä, joiden koherenssin kolmannen kvartiilin arvo vaihteli. Ristikorrelaatiolla löydettiin heikosti erottuneelle äänelle  $\hat{a}$  sen vahvinta komponenttia vastaava alkuperäinen ääni  $o$ . Alkuperäisestä äänestä  $o$  muodostettiin vastaääni  $\hat{o}$ , ja se yhdistettiin eroteltuun ääneen  $\hat{a}$ . Näin poistettiin erotellusta äänestä vahvin komponentti. Jäljelle jäänyttä ääntä  $\hat{a} - \hat{o}$  kuunneltiin ja arvioitiin. Kun koherenssi oli alle 0,7, jäljelle jäänyt ääni oli voimakasta. Tuon raja-arvon alle jääneet äänet todettiin huonosti erottuneiksi.

Varsinaisessa testissä määritetyn raja-arvon alle jääneet äänet tallennettiin analyysiä varten. Tallennetuissa äänissä kuului selvästi useampia äänilähteitä. Osassa niistä yksi ääni kuului selvästi muita vahvempana, ja se pystyttiin hyvin ihmiskorvin yhdistämään alkuperäiseen. Useimmiten päällekkäisiä ääniä kuului kahdesta neljään. Se on hyvä tulos, sillä eroteltavia alkuperäisiä ääniä oli jopa yli 15. Lähes poikkeuksetta ihmiskorva kykeni erottamaan, mitkä alkuperäiset äänilähteet olivat kyseessä. Huonoimmat erottelutulokset olivat käyttökeltottomia, koska niistä ei voitu erottaa yksittäistä ääntä.

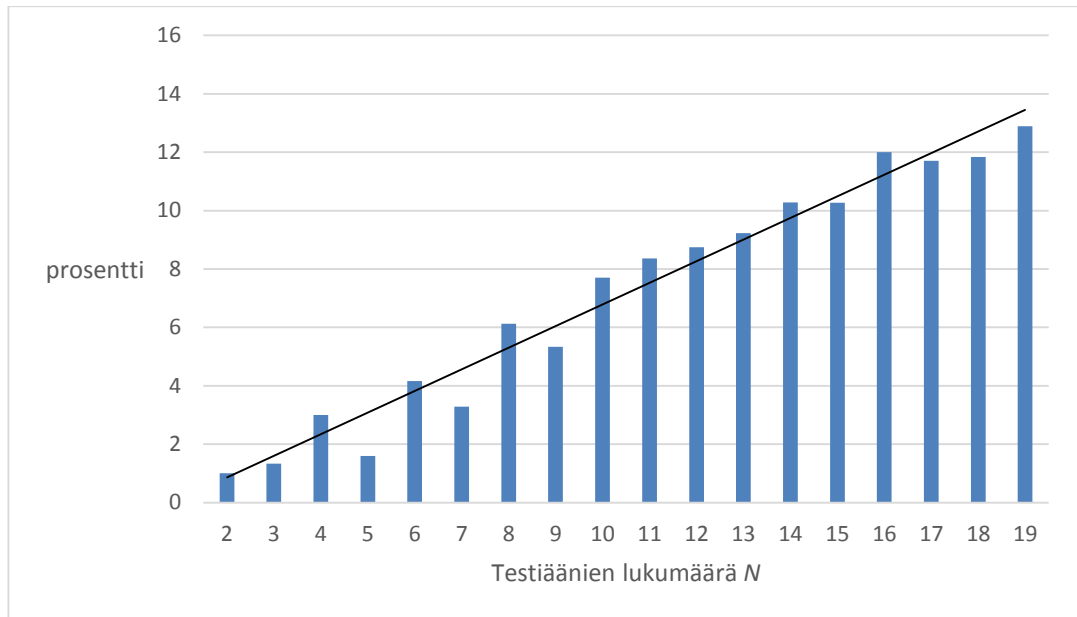
Kuvasta 5 nähdään arviointikriteerin mukaan huonoiten erottuneet äänet. Kuvassa lukumäärä ilmaisee, kuinka monesti kyseinen ääni arvioitiin huonosti erottuneeksi

kaikkien 1800 testiajon aikana. Selkeyden vuoksi alle kaksi kertaa huonosti erottuneet äänet jätettiin kuvassa huomioimatta. Matalataajuiset äänet, kuten basson ääni tai leijonan murina, erottuivat huonoiten FastICA-algoritmillä. Instrumenttien äänet erottuivat toiseksi huonoiten. Tallennetuista äänitiedostoista kävi ilmi, että alla olevat äänet olivat lähes joka kerta sekoittuneet keskenään. Äänet valittiin satunnaisesti ja yksi ääni esiintyi keskimäärin 472,5 kertaa testin aikana. Näin ollen esimerkiksi matalan murinan onnistunut erottelu tapahtui noin 20 % todennäköisyydellä.



Kuva 5. Huonosti erottuneet äänet.

Äänten palautus onnistui erinomaisesti, kun eroteltavien äänien lukumäärä oli alle kahdeksan. Tällöin huonosti erottuneita ääniä oli alle 5 %. Tämä nähdään kuvasta 6. Testiäänten lukumäärän  $N$  kasvaessa huonosti erottuneiden äänten prosentuaalinen lukumäärä kasvaa lähes lineaarisesti. Satunnaisuuden takia prosenttiosuus hieman vaihtelee. Lineaarisuutta voitaisiin tutkia 19:ää suuremmilla äänimäärillä ja useammalla kuin 1800 testiajolla. Reaalitilannetta ajatellen 19 äänilähdettä on jo riittävä lukumäärä.



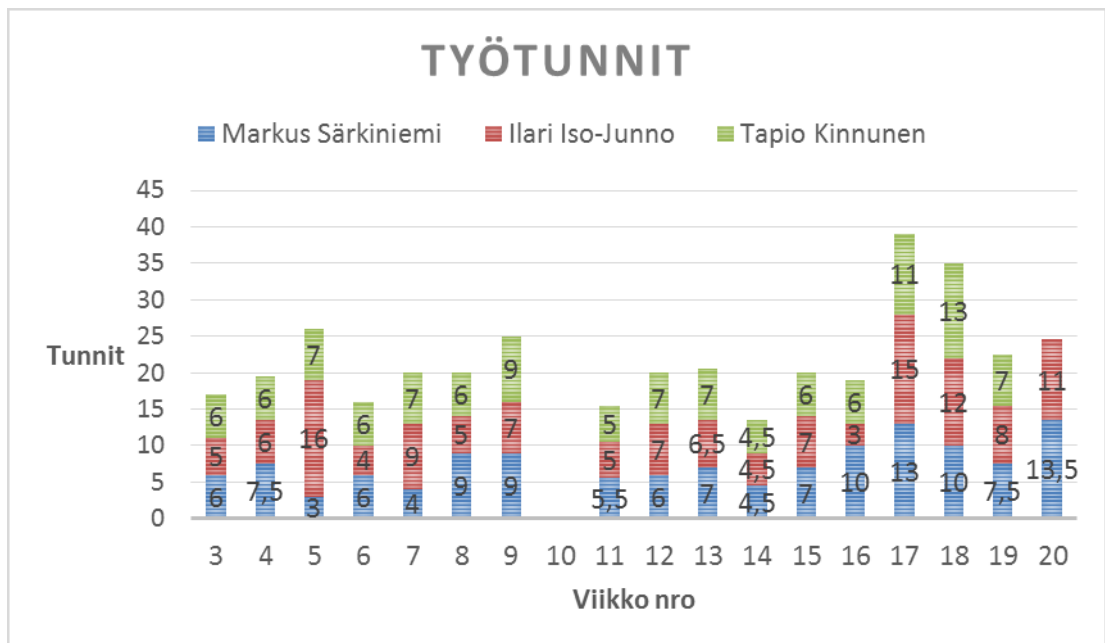
Kuva 6. Huonosti erottuneiden äänien prosenttiosuus testiäänien lukumäärän  $N$  suhteen.

## 5. PROJEKTIN KUVAUS

Projekti aloitettiin perehtymällä tieteellisiin julkaisuihin ja muuhun saatavilla olevaan kirjallisuuteen. Tämän jälkeen tutustuttiin laitteistoon ja valmisteltiin projektin ohjelmistoympäristö. Seuraavaksi mietittiin, kuinka voitaisiin hyödyntää taustatutkimuksessa löytyneitä valmiita ohjelmistopaketteja, ja päädyttiin käyttämään Aapo Hyvärisen FastICA:ta. FastICA-algoritmia kokeiltiin aluksi satunnaisesti valituilla äänitiedostoilla. Onnistuneiden erotusten jälkeen ohjelmistoa kehitettiin algoritmin käyttöä varten. Ohjelmiston kehittyessä sitä testattiin aluksi BeagleBone Black -kehitysalustalla, mutta nopeuden ja käytettävyyden vuoksi päädyttiin lopulta PC-ympäristön käyttöön. Tämän jälkeen laadittiin lopulliset testaussuunnitelmat ja toteutettiin ne. Sopivan mittarin löytäminen alkuperäisen ja erotellun signaalin vertailuun osoittautui haasteelliseksi. Useiden mittarien joukosta valittiin koherenssi. Testien tulokset analysoitiin ja havainnollistettiin kuvien avulla. Lopuksi pohdittiin jatkokehitystarpeita ja koostettiin yhteenveto.

### 5.1. Ajankäyttö

Kuvassa 7 nähdään viikkokohtaisesti ryhmän ajankäyttö tunteina.



Kuva 7. Työtunnit viikoittain.

## 6. JATKOKEHITYS

Tässä työssä ei oteta huomioon kaikua, äänilähteiden sijaintia eikä ICA:n rajoituksia. Tulevaisuudessa näitä asioita voidaan ottaa yksitellen mukaan jatkokehitykseen ja näin lähestyä realistisempaa tilannetta. Kaiku voitaisiin ottaa huomioon tutkimalla viivästettyjä signaaleja ristikorrelaation avulla. Tällöin samankaltaiset komponentit olisi mahdollista yhdistää samaan äänilähteeseen, ja näin tarkkuus paranisi. Tietoa äänilähteiden sijainnista voitaisiin hyödyntää sokean signaalin erottelussa. Sijainnilla saadaan yhdistettyä toisistaan riippuvat äänet, mitä ICA:lla ei voida tehdä. ICA:n rajoitukset voidaan todennäköisesti välttää ainoastaan vaihtamalla johonkin toiseen menetelmään. Tällä hetkellä ICA on paras menetelmä sokean signaalin erottelussa, ja näin ollen jatkokehitystä ajatellen ICA:n rinnalle voitaisiin kehittää menetelmiä paikkaamaan ICA:n rajoituksia:

- äänilähteiden pitää olla tilastollisesti riippumattomia
- permutaatio ja skaalaus -ongelma (the permutation and scaling problem)
- lähteiden oletetaan olevan paikallaan (stationary mixing model)

Ihmisten puheäänien erottelun testaaminen olisi järkevää. Näin saataisiin tietoa FastICA:n toimivuudesta oikeisiin järjestelmiin, esimerkiksi robotteihin. Jos oikeita mikrofoneja käytetään havaintojen tallentamiseen, täytyy ottaa huomioon myös äänien synkronointi, jotta erottelu voidaan tehdä. Synkronointi voidaan toteuttaa tarkastelemalla ristikorrelaation maksimin viivettä.

Kohinan  $n$  poistaminen äänestä tulisi ottaa huomioon ajateltaessa esimerkiksi äänitettyjä päällekkäisiä puheääniä. Kohinan poistoon voitaisiin mahdollisesti käyttää kohinaporttia.

ICA:ta voitaisiin testata myös muilla ICA-algoritmeilla, esimerkkinä JADE ja SHIBBS. Eri algoritmien kesken voitaisiin vertailla suoritusajkoja sekä erottelutuloksia. Esimerkiksi matalien äänten erotteluun voitaisiin näin löytää toimivampi algoritmi.

ICA:n mikrofoniongelmaa voitaisiin ratkaista ylivoimattamalla mikrofoniin lukumäärä. Äänitetyistä signaaleista voitaisiin arvioida äänilähteiden lukumäärää ajan suhteen ja syöttää sopiva määrä mikrofoniin signaaleja äänten erotteluun. Esimerkiksi PCA:n soveltuvuutta äänilähteiden lukumäärään arviointiin olisi mahdollista tutkia, mutta myös jo mahdollisia olemassa olevia toteutuksia voitaisiin etsiä.

Reaalitilannetta ajatellen äänilähteiden lukumäärä on tässä testissä jo riittävä. Reaalijasssa tapahtuvaa erottelua ajatellen tarvittaisiin äänilähteiden lukumäärän arvioinnin lisäksi sopiva aikaikkuna, jonka pituisina mikrofoniin havainnot syötettäisiin erottelualgoritmillemme. Jatkossa voitaisiin etsiä optimaalista ikkunankokoa, pohtia keinoja yhdistää ikkunoidut ja separoidut äänilohkot äänivirroiksi sekä tutkia reaaliaikaisuuden laskentatehon tarvetta.

## 7. YHTEENVETO

Tällä hetkellä äänimaisema-analyysin koneellisissa ratkaisuisa ei saavuteta ihmisen erottelukyvyn tasoa, koska tuntemattomien muuttujien lukumäärä on suuri. Siitä huolimatta ICA-menetelmillä saadaan varteenotettavia tuloksia aikaiseksi, jos käytetään ohjelmallisesti luotuja ideaalisia signaaleja.

Työn päämäärä oli FastICA:n erottelukyvyn testaus kohinattomilla ja erilaisilla äänisignaaleilla äänien lukumäärän kasvaessa. Työssä FastICA-algoritmin erottelukykyä testattiin 2 - 19 ideaalisella äänilähteellä ja todettiin, että erottelukyky heikkenee äänimäärän kasvaessa. Erinomaiseen erotteluun päästiin alle kahdeksalla äänellä, jolloin yli 95 % äänistä erottui hyvin. Tulokset olivat kohtuullisia myös suurilla äänimäärillä, mutta niissä oli yleensä useita tulostiedostoja, joissa useampi ääni oli sekoittunut keskenään. Matalat bassoäänet ja leijonan murina olivat yleensä separoituissa äänissä keskenään, eli FastICA ei kyennyt niitä erottelemaan.

FastICA valittiin tutkittavaksi, koska se on yksi nopeimpia ICA-algoritmeja ja hyvin tunnettu. Siihen on saatavilla kirjastoja monelle ohjelmointikielelle, ja siten se on helppo ottaa käyttöön eri laitteistoille.

Tuloksista hyötyvät FastICA:n käyttäjät. Tulosten valossa muut BSS:n tutkijat voivat pohtia FastICA:n soveltuvuutta omiin käyttötarpeisiinsa. Työ luo kohtuullisen kuvan FastICA:n erottelukyvystä, ja siten se voi vaikuttaa merkittävästi päätöksiin käyttää FastICA:ta.

## 8. LÄHTEET

- [1] Bregman A.S. (1984) Auditory scene analysis. Proceedings of the 7th International Conference on Pattern Recognition. : 168-175.
- [2] Hidri A., Meddeb S. & Amiri H. (2012) About Multichannel Speech Signal Extraction and Separation Techniques. Journal of Signal and Information Processing 3(2): 238.
- [3] Wang D. & Brown G.J. (2006) Computational auditory scene analysis: Principles, algorithms, and applications. , Wiley-IEEE Press.
- [4] Äänipää - Tampereen ammattikorkeakoulu (luettu 29.1.2015) Äänen synty. URL: [http://www.aanipaa.tamk.fi/synty\\_1.htm](http://www.aanipaa.tamk.fi/synty_1.htm).
- [5] Tampereen yliopisto (luettu 29.1.2015) Ääni - Perusteet. URL: [www.uta.fi/mute/aai01.htm](http://www.uta.fi/mute/aai01.htm).
- [6] Nygren J. (luettu 29.1.2015) Korvan aistit | Ihminen | Oppiminen | yle.fi. URL: <http://oppiminen.yle.fi/ihminen/aistit/korvan-aistit>.
- [7] Haykin S. & Chen Z. (2005) The Cocktail Party Problem. Neural Comput 17(9): 1875-1902.
- [8] Clifford G.D. (2007) Course materials for HST.582J / 6.555J / 16.456J, Biomedical Signal and Image Processing, MIT OpenCourseWare (<http://ocw.mit.edu>), Massachusetts Institute of Technology. Downloaded on [26 1 2015].
- [9] Brown G.J. & Wang D. (2005) Separation of speech by computational auditory scene analysis. In: Anonymous , Springer: 371-402.
- [10] Hu K. (2012) Speech segregation in background noise and competing speech.
- [11] Matsuoka K. (2002) Minimal distortion principle for blind source separation. SICE 2002. Proceedings of the 41st SICE Annual Conference. , IEEE 4: 2138-2143.
- [12] Pedersen M.S., Larsen J., Kjems U. & Parra L.C. (2007) A survey of convolutive blind source separation methods. Multichannel Speech Processing Handbook : 1065-1084.
- [13] Belouchrani A., Abed-Meraim K., Cardoso J. & Moulines E. (1997) A blind source separation technique using second-order statistics. Signal Processing, IEEE Transactions on 45(2): 434-444.
- [14] Jutten C. & Comon P. (2010) Chapter 1 - Introduction. In: Comon P & Jutten C (eds) Handbook of Blind Source Separation. Oxford, Academic Press: 1-22.



- [15] Hyvärinen A., Karhunen J. & Oja E. (2001, 2002) Introduction. In: Anonymous Independent Component Analysis. , John Wiley & Sons, Inc.: 1-12.
- [16] Nikunen J. (2015) Object-based Modeling of Audio for Coding and Source Separation. Tampereen teknillinen yliopisto.Julkaisu-Tampere University of Technology.Publication; 1276 .
- [17] He X. & Zhu T. (2014) ICA of Noisy Music Audio Mixtures Based on Iterative Shrinkage Denoising and FastICA Using Rational Nonlinearities. In: Anonymous - Circuits, Systems, and Signal Processing. , - Springer US: 1917-1956.
- [18] Lee T., Lewicki M.S., Girolami M. & Sejnowski T.J. (1999) Blind source separation of more sources than mixtures using overcomplete representations. Signal Processing Letters, IEEE 6(4): 87-90.
- [19] Cervigón R., Heneghan C., Moreno J., Castells F. & Millet J. (2007) Evaluation of Propofol Effects in Atrial Fibrillation Using Principal and Independent Component Analysis. In: Davies M, James C, Abdallah S & Plumbley M (eds) - Independent Component Analysis and Signal Separation. , - Springer Berlin Heidelberg: 569-576.
- [20] Müller W., Nocke T. & Schumann H. (2006) Enhancing the Visualization Process with Principal Component Analysis to Support the Exploration of Trends. Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation - Volume 60. Darlinghurst, Australia, Australia, Australian Computer Society, Inc: 121-130.
- [21] Hyvärinen A. (luettu 23.2.2015) FastICA. URL: <http://research.ics.aalto.fi/ica/fastica/>.
- [22] Helsinki University of Technology: Laboratory of Computer and Information Science - Adaptive Informatics Research Centre (luettu 23.2.2015) Fixed-point algorithm for ICA. URL: <http://research.ics.aalto.fi/ica/fastica/fp.shtml>.
- [23] Hyvärinen A. (luettu 01.04.2015) Definition of ICA. URL: [http://cis.legacy.ics.tkk.fi/aapo/papers/IJCNN99\\_tutorialweb/node3.html](http://cis.legacy.ics.tkk.fi/aapo/papers/IJCNN99_tutorialweb/node3.html).
- [24] Lyon D.A. (luettu 29.04.2015) The Discrete Fourier Transform, Part 6: Cross-Correlation. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.167.8344&rep=rep1&type=pdf>.
- [25] Penny W.D. (luettu 29.4.2015) Signal Processing Course. URL: <http://www.fil.ion.ucl.ac.uk/~wpenny/course/course.pdf>.