# AUTOMAATTINEN PUHUJAN TUNNISTUS

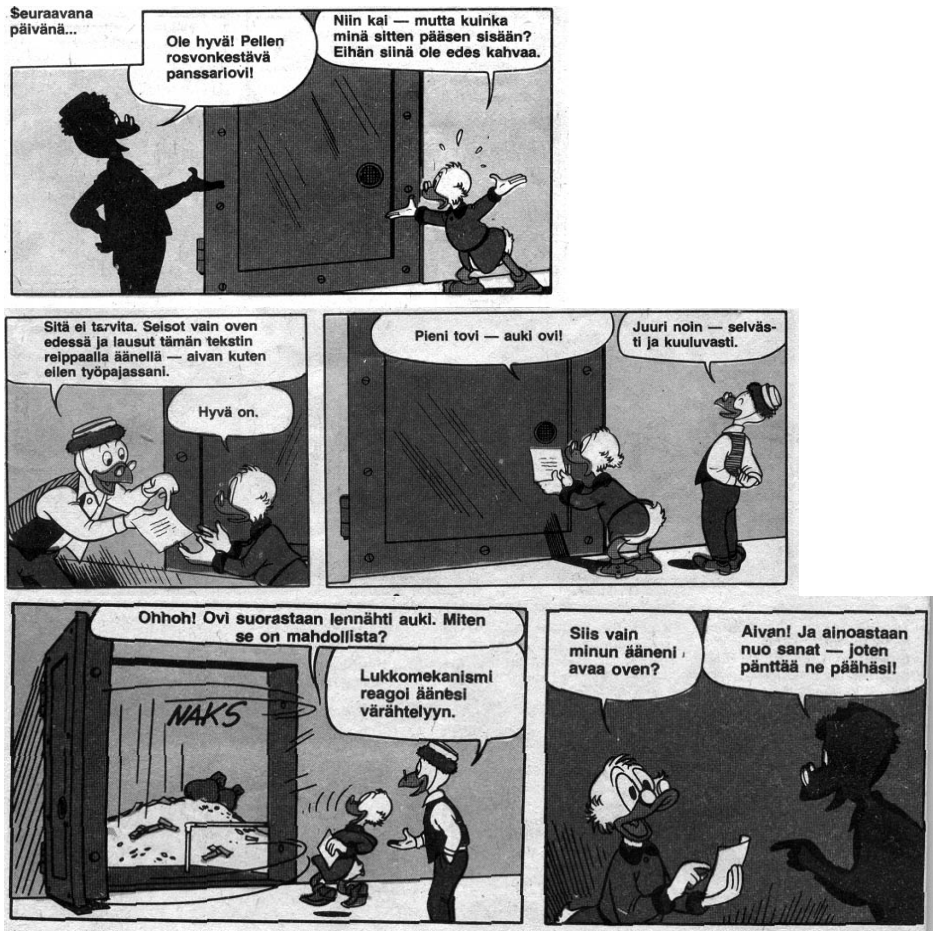
Tomi Kinnunen

Joensuun yliopisto
Tietojenkäsittelytieteen laitos
Pro gradu –tutkielma
15.12.1999

# TIIVISTELMÄ

Tutkielmassa perehdytään kirjallisuuden pohjalta automaattiseen puhujan tunnistukseen, erityisesti tekstistä riippumattomien menetelmien kannalta sekä identifiointi- että verifiointitehtävissä. Piirreirroituksessa erityishuomiota kiinnitetään kepstrin määrittämiseen, koska tämä ja sen johdannaiset ovat tällä hetkellä ylivoimaisesti eniten käytetty piirrejoukko. Luokittelun pääpaino on vektorikvantisoinnissa (VQ), joka on hyväksi havaittu ja yksinkertainen luokittelumenetelmä. Tutkielmaan sisältyy myös kokeellinen osa, jossa testataan televisiosta kerätyllä kohinaisella ja rajoitetulla puhedatalla kepstri-VQ –yhdistelmän toimivuutta puhujan tunnistuksessa.

***Avainsanat:*** Puhujan identifiointi, puhujan verifiointi, tekstistä riippumattomuus, kepstri, vektorikvantisointi.




**Kuvasarja 0.1.** Tekstistä riippuva puhujan verifiointijärjestelmä.

# MERKINNÄT JA LYHENTEET

| | |
|---|---|
| $x$, $\boldsymbol{x}$ | Skalaari, vektori |
| $\lfloor x \rfloor$ | Suurin kokonaisluku, joka on $\leq x$ |
| $\boldsymbol{A}$, $\boldsymbol{A}^{\mathrm{T}}$ | Matriisi, matriisin transpoosi |
| $\lvert X \rvert$ | Äärellisen joukon $X$ kardinaliteetti (alkioiden määrä) |
| $\lvert z \rvert$ | kompleksi- tai reaaliluvun $z$ itseisarvo |
| $\mathbf{R}$ | Reaalilukujen joukko |
| $\mathbf{R}^p$ | $p$-ulotteinen reaalinen vektoriavaruus |
| $s_a(t)$, $s(n)$ | Analoginen signaali, digitaalinen signaali |
| $S(\omega)$ | Signaalin $s$ Fourier-muunnos |
| $x * y$ | Signaalien $x$ ja $y$ konvoluutio |
| | |
| DCT | Diskreetti kosinimuunnos |
| DFT | Diskreetti Fourier-muunnos |
| DTW | Dynaaminen aikasovitus |
| EER | Yhtäsuuri virhemäärä (FA = FR) |
| FA | Väärien hyväksymisvirhe |
| FFT | Nopea Fourier-muunnos |
| FR | Oikeiden hylkäämisvirhe |
| GVQ | Ryhmävektorikvantisointi |
| HMM | Kätketty Markovin malli |
| HPF | Ylipäästösuodatus |
| LP | Lineaariprediktio |
| LVQ | Oppiva vektorikvantisaatio |
| MSE | Keskineliövirhe |
| VQ | Vektorikvantisointi |

# SISÄLLYSLUETTELO

# 1 JOHDANTO

Tarve tunnistaa ihminen on kasvanut automaation kehityksen myötä. Esimerkiksi pikapankin käyttäjältä vaaditaan kortti ja tämän lisäksi tunnusluku, jotka yhdessä identifioivat käyttäjän. Muita sovelluksia löydetään esimerkiksi erilaisista turvajärjestelmistä: vain salasanalla tai tunnuskoodilla pääsee tietylle alueelle. Tällaisiin perinteisiin tunnistusmenetelmiin liittyy kuitenkin yksi heikkous: tunnistuksessa käytettävät ominaisuudet eivät riipu henkilöstä itsestään millään tavalla; toisin sanoen toinen henkilö, jolla on pankkikortti hallussaan ja joka tietää tunnusluvun tai salasanan, tunnistetaan samaksi henkilöksi. Jotta tunnistaminen olisi luotettavampaa, tunnistuksen pitäisi riippua koodin tai salasanan lisäksi henkilön yksilöllisistä ominaisuuksista. Rikostutkinnassa käytettäviä menetelmiä ovat esimerkiksi sormenjälkiin ja DNA-näytteisiin perustuva tunnistaminen. Tässä tutkielmassa luomme katsauksen *puhujan tunnistuksen* ongelmakenttään ja menetelmiin.

## *1.1 Mitä on puhujan tunnistus?*

Puhujan tunnistuksessa on tavoitteena tunnistaa puhujan äänen perusteella, kuka puhuja on kyseessä [16, 31]. Suoritamme jokapäiväisessä elämässä puhujan tunnistamista. Varhaisimmat tutkimukset puheen sisältämästä identiteetti-informaatiosta juontavat juurensa psykologisiin tutkimuksiin 1920-luvulle. Tietokoneiden kehityksen myötä toisen maailmansodan jälkeen on kuitenkin keskitytty pääasiassa tämän prosessin *automatisointiin*. Tässä tutkielmassa perehdymme nimenomaan automaattiseen puhujan tunnistustamiseen. Tulemme samalla kuitenkin huomaamaan myös psykologisten sekä eräiden muidenkin tieteenalojen tutkimusten yhteyden puhujantunnistusongelmaan.

Puhujan tunnistus voidaan jakaa kahteen luokkaan tehtävän luonteen mukaan [11, 14, 16, 31]. *Identifiointitehtävässä* tavoitteena on tunnistaa tuntematon puhuja useamman tunnetun puhujan joukosta. *Verifiointitehtävässä* taas on tarkoituksena antaa päätös siitä, onko tutkittava puhuja se, joka hänen väitetään olevan. Väitös voidaan antaa puhujantunnistusjärjestelmälle monellakin tavalla, esim. tunnuslukuna tai salasanana.

Kuten kaikki hahmontunnistusongelmat, myös puhujan tunnistus on kaksivaiheinen. *Opetusvaiheessa* muodostetaan kullekin puhujalle oma matemaattinen mallinsa, joka kuvaa juuri tämän puhujan

yksilöllisiä piirteitä. Mallintamiseen tarvitaan suuri joukko puhedataa, jotta oikean tunnistuspäätöksen todennäköisyys olisi tarpeeksi suuri ja väärän päätöksen todennäköisyys tarpeeksi pieni.

Opetusvaihe on yhteinen sekä identifiointi- että verifiointitehtäville. Menetelmät poikkeavatkin vasta *tunnistusvaiheen* osalta. Identifioinnissa tutkittavan puhujan mallia verrataan kaikkien muiden puhujien malleihin ja valitaan sitten jonkin sopivan kriteerin mukaan "lähimpänä oleva" malli. Verifioimisessa tutkittavan puhujan mallia verrataan vain väitetyn puhujan malliin ja tehdään päätös siitä, ovatko mallit "tarpeeksi lähellä" toisiaan. Jos ovat, annetaan positiivinen tunnistustulos, muutoin negatiivinen. Yleisesti identifioiminen on laskennallisesti raskaampi kuin verifioiminen, erityisesti jos populaation koko on suuri.

### *1.2 Miksi puhujan tunnistus on mahdollista?*

On hyvin epätodennäköistä, että kahdella eri henkilöllä olisi tismalleen samalla tavalla käyttäytyvä äänentuottoelimistö [22], koska äänen muodostamiseen osallistuu niin paljon eri elimiä. Omatakseen identtisen äänen kahden henkilön hyvin monien fysiologisten muuttujien pitäisi sattua kohdakkain sekä lisäksi heidän (aikaisin opittu) puhetyylinsä tulisi vielä olla samanlainen. Puhujan identiteetti siis korreloi sekä fysiologisten että psykologisten ominaispiirteiden kanssa [30]. Tärkeä kysymys puhujan tunnistusprobleemassa on, kuinka näitä piirteitä pitäisi mitata. Raa'asta puhesignaalista olisi löydettävä sellaiset piirteet, jotka mallintavat puhujan yksilöllisiä ominaisuuksia.

### *1.3 Puhujan tunnistuksen luokittelu*

Puhujan tunnistusmenetelmät jaetaan *tekstistä riippuviin* ja *tekstistä riippumattomiin* menetelmiin [14, 16, 30]. Tekstistä riippumattomissa menetelmissä puhuja tunnistetaan puhutun tekstin sisällöstä riippumatta. Tekstistä riippuvissa menetelmissä tunnistettava henkilö lukee ennalta määrätyn tekstin, joka ei välttämättä ole sama kuin opetusvaiheessa käytetty.

Eräs mielenkiintoinen tekstistä riippuvien verifiointimenetelmien alaluokka on *tekstikehotteinen puhujan verifiointi* [30]. Tällöin tunnistettavan puhujan on luettava joka kerta eri teksti, jonka kone

satunnaisesti määrää. Tämän menetelmän tarkoituksena on vähentää järjestelmän “huijattavuutta”. Henkilö *A* voisi yrittää murtautua turvajärjestelmään nauhoittamalla esimerkiksi puhelinkeskustelussa henkilön *B* puhetta ja toistamalla tunnistusvaiheessa tämän nauhoitteen. Tekstikehotteisen puhujan verifiointijärjestelmän huijaamista varten tarvittaisiin kehittynyt puhesyntetisoija, joka pystyisi tuottamaan tismalleen halutun henkilön äänenvärillä koneen pyytämät sanat.

Tekstistä riippuviin menetelmiin liittyy hyvin läheisesti toinen tutkimusala, *puheentunnistus*, jossa ideaalisessa tapauksessa saadaan puhujasta riippumatta selville tekstin lingvistinen sisältö. Puhujan- ja puheentunnistuksessa käytetään pitkälti samoja metodiikkoja, minkä tulemme huomaamaan myöhemmissä luvuissa.

Puhujatietokannat ovat joko *avoimia* tai *suljetuttuja* joukkoja [11, 16]. Ensin mainittu tarkoittaa, että tunnistettava puhuja voi puuttua tietokannasta kokonaan, jälkimmäisessä taas kaikki puhujat ovat tiedossa. Identifiointitehtävässä tämä on oleellinen seikka, sillä jos tunnistettavan puhujan malli puuttuu kokonaan tietokannasta, on annettava luokittelutulos “ei päätöstä”.

### *1.4 Sovelluksia*

***Turvallisuussovellukset.*** Tavalliset mekaaniset avaimet ja lukot ovat vähitellen korvautumassa elektronisilla vastineillaan. Tunnistettavalla voi olla esimerkiksi tunnuskortti ja -luku, joilla hän pääsee tietylle alueelle. Tietokoneistuvassa maailmassa käyttäjätunnukset ja salasanat ovat arkipäivää. Jotta luvaton pääsy alueelle tai toisen käyttäjän tiedostoihin voitaisiin estää, kannattaa tunnistaminen tehdä käyttäen useampaa *modaliteettia* eli vuorovaikutustapaa. Salasanan tai tunnusluvun lisäksi voidaan käyttää vaikkapa verkkokalvon tai kämmenen skannausta ja/tai puheeseen perustuvaa tunnistamista. Käytettäessä tunnistuksessa monia modaliteetteja väärän tunnistuksen todennäköisyys putoaa merkittävästi. Tyypillisiä todellisessa käytössä olevia sovelluksia ovat mm. puhelinpankkipalvelut [esim. 31].

***Puheentunnistusjärjestelmät.*** Puheen- ja puhujan tunnistuksen välillä on monia yhteyksiä. Tutkimukset ovat osoittaneet, että puheentunnistusjärjestelmän tarkkuus kasvaa huomattavasti, kun järjestelmä “viritetään” kohdepuhujan ääntä varten. Eräs puheentunnistuksen perusongelmista on universaalin, puhujasta riippumattoman mallin puuttuminen. Jos puhetta tunnistava järjestelmä voisi

ensin tunnistaa puhujan henkilöllisyyden automaattisesti, se voisi adaptoida itsensä tunnistamaan juuri tämän henkilön puhetta. Puhekäyttöliittymät ovat nykyaikana intensiivisen tutkimuksen kohteena, joten puhujan tunnistuksen kysymykset saattavat nousta oleellisen tärkeiksi, mikäli puheohjattavat järjestelmät yleistyvät tulevaisuudessa.

***Rikostutkimus.***. Rikostutkimuksessa on tavoitteena selvittää rikoksen tekijä. Nykytekniikka on mahdollistanut monia *biometrisia* mittaustapoja [13, 30], joilla rikoksen tekijä voidaan selvittää. Ns. *fyysisiä* biometriikkoja ovat esimerkiksi sormenjäljet, DNA-testit, verkkokalvon tai kämmenen skannaus, kasvojen tunnistus jne. Näihin ihminen ei voi itse vaikuttaa. Sen sijaan ns. *suoritusbiometriikkoihin* ihminen voi ainakin osittain käytöksellään vaikuttaa. Tällaisia mittoja ovat mm. käsiala ja puhe. Puhujan tunnistamista voidaan käyttää oikeudessa todisteena, mikäli rikoksen tekijän ääni on talletettu rikoksen tapahtumahetkellä. Hankaluutena tällaisissa sovelluksissa on, ettei epäilty halua tulla tunnistetuksi ja voi esimerkiksi tahallaan muuttaa ääntään poliisitutkimuksen yhteydessä tehtävässä nauhoituksessa. Automaattisen puhujan tunnistuksen tavoite on objektiivisesti selvittää puhujan identiteetti. Kuitenkaan todisteena ei voida käyttää pelkästään henkilön ääneen perustuvaa automaattista tunnistamista, vaan mukana on oltava ihmisen tekemää subjektiivista arviointia yksittäisten todisteiden painoarvosta. Nämä kysymykset ovat kriittisiä, koska pahimmassa tapauksessa voidaan tuomita syytön henkilö ja vapauttaa oikea syyllinen epäilyksistä.

## *1.5 Tutkielman sisältö*

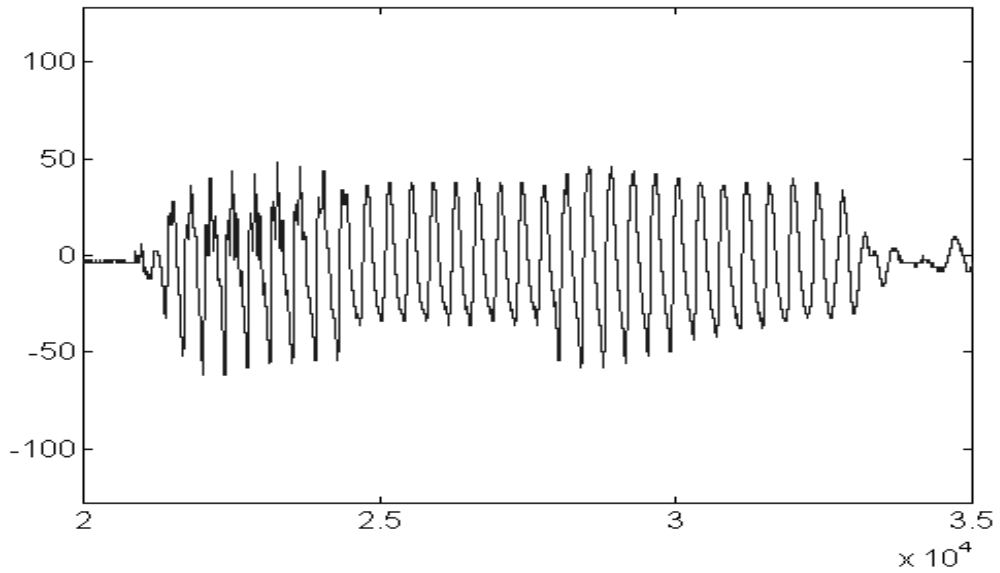
Tämän tutkielman pääpaino on tekstistä riippumattomassa puhujan tunnistuksessa. Sisältö on jaoteltu seuraavasti. Luvussa 2 kertaamme DSP:n perusmenetelmiä, kuten Fourier-analyysia. Luvussa 3 perehdymme ihmisen puheentuottomekanismiin ja tämän matemaattiseen mallintamiseen. Luvussa 4 käsittelemme piirreirroitusta eli prosessia, jossa puhesignaalista määritetään puhujaa karakterisoivat parametrit. Luvussa 5 tarkastelemme luokittelua eli prosessia, jolla piirteet jaotellaan eri puhujaluokkiin ja kuinka tuntematon puhuja tunnistetaan. Tutkielman tekoon sisältyi myös kokeellinen osio, jonka kuvaus ja saadut tulokset on annettu kuudennessa luvussa. Viimeisessä luvussa vertailemme saatuja tuloksia kirjallisuudessa saavutettuihin ja pohdimme, miten tunnistustarkkuutta voitaisiin edelleen parantaa.

# 2 DIGITAALINEN PUHEEN KÄSITTELY JA ANALYYSI

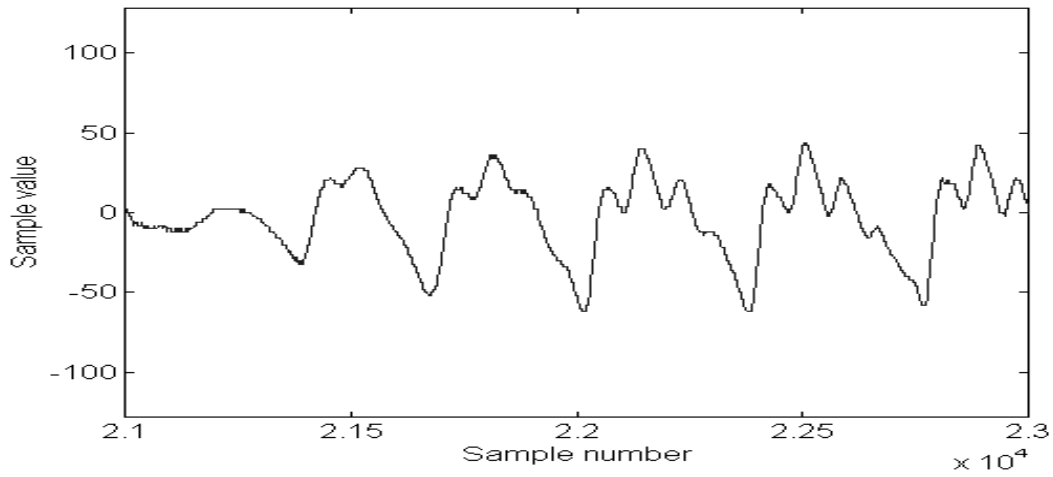
Tässä luvussa käymme lyhyesti läpi DSP:n perusteista signaalin digitoinnin sekä Fourier-analyysin perusteet. Tulemme myöhemmissä luvuissa soveltamaan intensiivisesti näitä menetelmiä.

## *2.1 Puhesignaalin digitointi*

Ihmisen puhe ilmenee ilmanpaineen vaihteluina, jotka muunnetaan mikrofonin avulla jännitevaihteluiksi. Signaali voidaan esittää piirtämällä mitatut paineen (jännitteen) arvot aikatasossa. Tällaista signaalin esitysmuotoa kutsutaan *aaltomuodoksi*. Kuvissa 2.1 (a) ja (b) on esimerkki tyypillisestä puheen aaltomuodosta.




**Kuva 2.1. (a)** Tyypillinen puhesignaalin aaltomuoto (mieshenkilö sanoo ”anna”). Vaaka-akselilla ovat näytearvojen numerot (≅ aika) ja pystyakselilla näytteiden arvot (≅ ilmanpaine/jännite). Kuvan signaali on näytteistetty taajuudella 44.1 kHz (cd-laatu) ja kvantisoitu 8-bittiseksi etumerkillisiksi kokonaisluvuiksi. Toisin sanoen käytössä oleva amplitudiasteikko kattaa $2^8$ = 256 tasoa, 2-komplementtiesityksessä siis arvot -128..127.




**Kuva 2.1. (b)** (a)-kohdan aaltomuotoa suurennettuna. Silmämääräisesti nähdään, että signaali pysyy likimain stationaarisena ainakin lyhyellä aikavälillä.

Käytännön kannalta mielenkiintoiset reaalimaailman signaalit ovat useimmiten *analogisia* eli luonteeltaan jatkuvia ja dynaamisia. Käsiteltäessä ja analysoitaessa tietokoneella analogista signaalia se täytyy kuitenkin muuntaa *digitaaliseksi* signaaliksi, joka voidaan esittää äärellisellä määrällä äärellisiä lukuarvoja. Tätä prosessia kutsutaan *A/D-muunnokseksi* (ADC) [32]. A/D-muunnoksen voidaan teoriassa ajatella muodostuvan kahdesta eri vaiheesta: ensin signaali *näytteistetään*, jonka jälkeen se *kvantisoidaan* haluttuun esitystarkkuuteen.

*Näytteistäminen* tarkoittaa yksinkertaisesti, että signaalista otetaan näytearvo tasaisin väliajoin. *Näytteenottoväli* $T$ tarkoittaa näytteiden välistä aikaa ja sen käänteisarvo $F_s = 1/T$ on *näytteenottotaajuus*. Näytteenottotaajuuden arvo tarkoittaa yhdessä sekunnissa otettavien näytteiden lukumäärää. Eräs lähtökysymys digitaalisen signaalinkäsittelytehtävän alussa on päättää, onko signaalin hyvä laatu vai vähäinen tilantarve oleellisempi eli mikä on tilanteeseen sopivin näytteenottotaajuus. Puheanalyysissa ja puheenkoodauksessa tyydytään usein hyvinkin pieniin näytteenottotaajuuksiin, esimerkiksi 8 kHz:iin, joka on vaikkapa tavallisen cd-äänitteen 44.1 kHz:iin verrattuna hyvin pieni. Digitaalista puhetta pystyy vielä hyvin ymmärtämään 8 kHz:n näytteistystaajuudella.

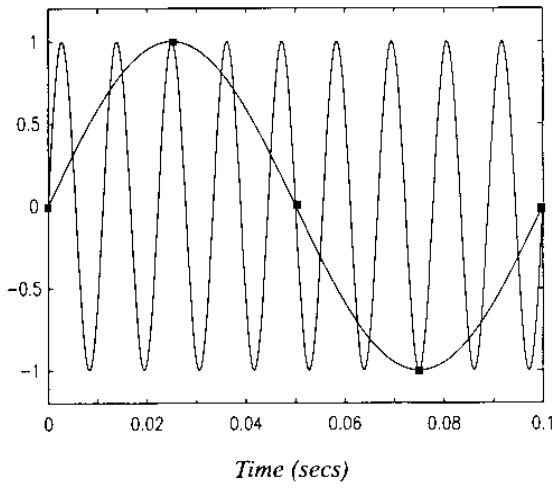
Merkitään analogista puhesignaalia $s_a(t)$ :llä, missä $t \in \mathbf{R}$. Funktio $s_a(t)$ ilmoittaa siis ilmanpaineen (jännitteen) arvon ajan hetkellä $t$. Tunnus $s$ viittaa puheeseen ja alaindeksi $a$ analogiseen. Analogisesta signaalista näytteistämällä saatu digitaalinen signaali $s(n)$ on

$$s(n) = s_a(nT) = \{ \ldots, s(-1), s(0), s(1), \ldots \} \quad (2.1.1)$$

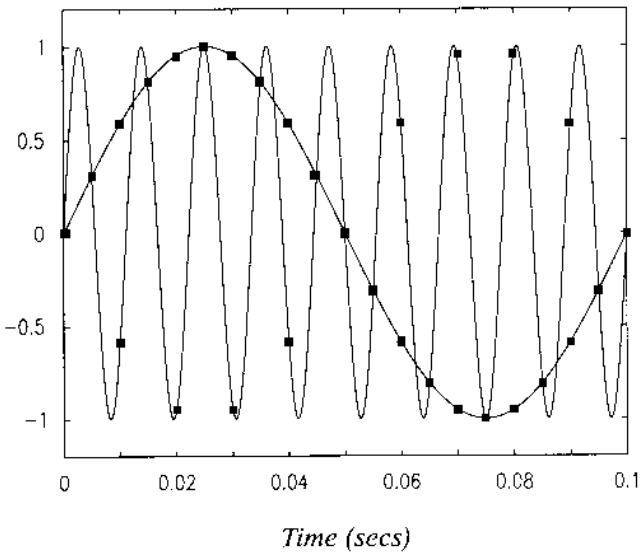
Näytearvojen indeksoiminen kokonaisluvuilla on yleinen käytäntö signaalinkäsittelyssä. Jatkossa tarkoitamme kaikilla signaaleilla digitaalisia signaaleita.

*Näytteenottoteoreeman* [32] mukaan alkuperäinen analoginen signaali voidaan täysin konstruoida uudelleen näytearvoista, mikäli signaali on jaksollinen ja $F_s > 2F_{\max}$ eli näytteenottotaajuus on suurempi kuin analogisen signaalin korkeataajuisimman komponentin taajuus kaksinkertaisena. Taajuutta $2F_{max}$ kutsutaan signaalin *Nyquist-taajuudeksi* [32].

Näytteistettäessä Nyquist-taajuutta suuremmalla taajuudella vältytään haitalliselta *aliasing*-ilmiöltä, joka tarkoittaa, että kaksi eritaajuista signaalia saavat samat näytearvot. Tällöin informaatiota kadotetaan, koska kaksi eritaajuista signaalia tulkitaan samaksi signaaliksi. Aliasing on haitallinen ilmiö, sillä tällöin korkeat taajuudet esiintyvät aaltomuodossa matalina taajuuksina; tehtäessä taajuusanalyysiä matalien taajuuksien energiajakauma näyttää tällöin olevan suurempi kuin mikä se todellisuudessa on! Mikäli näytteistystaajuutta ei ole mahdollista kasvattaa, on käytettävä muita keinoja tämän haitallisen ilmiön eliminoimiseksi. Tähän palaamme tarkemmin piirreirroituksen yhteydessä. Aliasing-ilmiötä on havainnollistettu kuvissa 2.2 (a) ja (b).




**Kuva 2.2. (a)** [12]. Aliasing-ilmiö. Kaksi eritaajuista signaalia (10 Hz ja 90 Hz) näytteistetään 40 Hz taajuudella. Näytearvot (mustat pisteet) ovat samat molemmille signaaleille, koska näytteenottotaajuus on liian pieni. Tämä tarkoittaa käytännössä, että 90 Hz taajuinen signaali näkyy digitaalisessa signaalissa taajuutena 10 Hz, mikä vääristää taajuuksien energiajakaumaa.




**Kuva 2.2. (b)** [12]. Aliasing-ilmiön eliminointi. Samat signaalit näytteistetään 200 Hz:n taajuudella, jolloin aliasingia ei esiinny (sillä $F_s$ = 200 Hz > $2F_{\max}$ = 2·90 Hz = 180 Hz).

Jälkimmäinen vaihe A/D-muunnoksessa on *kvantisointi*. Tämä tarkoittaa näytearvojen koodaamista sopivaan lukuesitykseen, jota on helppo käsitellä tietokoneella. Eräs tyypillinen koodaus on näytearvojen esittäminen etumerkillisinä kokonaislukuina. Esimerkiksi 16-bittisillä kokonaisluvuilla saatava kvantisointiasteikko kattaa $2^{16}$ = 65536 arvoa eli arvot –32768…32767. Tämä tarkkuus on käytännössä useimpiin sovelluksiin riittävä. Puhetta käsiteltäessä riittää useasti pienempikin resoluutio.

## *2.2 Fourier-analyysi*

Tehtäessä signaalin muokkausta ja analyysiä joudutaan signaali usein muuntamaan aaltomuotoesityksestä taajuusesitykseksi. *Fourierin muunnoksella* saadaan selville signaalin kunkin taajuuden amplitudi signaalissa ja käänteismuunnoksella taajuusalueeseen muunnettu signaali voidaan muuntaa takaisin aika-alueeseen. Kyse on siis pohjimmiltaan signaalin eri esitysmuodoista. Seuraavassa tarkastelemme diskreettiä Fourier-analyysia, ja erityisesti diskreettiä Fourier-muunnosta (DFT), sillä tämä on signaalianalyysin tärkeimpiä apuvälineitä ja tulemme soveltamaan sitä piirreirroituksen yhteydessä.

### 2.2.1 Fourierin muunnos

Kirjallisuudessa Fourierin muunnokselle esitetään hiukan toisistaan poikkeavia määritelmiä ja merkintätapoja, mutta kaikissa on oleellisesti kysymys samasta asiasta. Tässä esityksessä käytämme ajan ja amplitudin suhteen diskreetin signaalin $s(n)$ Fourierin muunnokselle seuraavaa määritelmää [8, 32]:

$$S(\omega) = \sum_{n=-\infty}^{\infty} s(n) e^{-i\omega n}, \qquad 0 \le \omega \le 2\pi \tag{2.2.1}$$

missä $\omega$ on taajuus ja $i$ imaginaariyksikkö. Signaalikirjallisuuden yleisen tavan mukaan käytämme pieniä kirjaimia kuvaamaan aika-alueen signaaleja ja isoja taajuusalueen signaaleja. Alkuperäiselle signaalille $s(n)$ tehty Fourierin muunnos antaa tuloksenaan kompleksiarvoisen $2\pi$-jaksollisen funktion.

Muunnos antaa kahdenlaista tietoa: sekä amplitudi- että vaiheinformaatiota. Useimmissa sovelluksissa, myöskään puhujan tunnistuksessa, emme ole kiinnostuneita eri taajuuksien vaihe-eroista. Sen sijaan erittäin olennainen on tieto tietyn taajuuden amplitudista funktiossa $s(n)$. Amplitudi saadaan ottamalla muunnoksesta itseisarvo $|S(\omega)|$ eli *moduli*, joka määritellään kompleksiluvulle $z = a + bi$ seuraavasti:

$$|z| = \sqrt{a^2 + b^2} \tag{2.2.2}$$

Fourierin muunnoksen itseisarvoa sanotaan *spektriksi* tai *amplitudispektriksi*.

### 2.2.2 Lyhyen aikavälin analyysi

On selvää, että relaation 2.2.1 toteuttaminen oikealla tietokoneella ei ole mahdollista, koska laskettavana on ääretön summa. Käytännön sovelluksissa signaali $s(n)$ koostuu aina äärellisestä määrästä arvoja. Puheanalyysisovelluksissa puhesignaali jaetaan pienempiin aikakaistoihin (*kehyksiin*), joita käsitellään itsenäisinä signaaleinaan. Kehyksen pituus tulisi valita siten, että sen sisällä oleva signaali olisi likimain *stationaarinen* eli taajuuksien suhteen vakio. Deller & al. [8]

antavat nyrkkisääntönä puheen tapauksessa kehyksen pituudeksi suuruusluokan 20 millisekuntia. Tämän pituisen kehyksen sisällä puhesignaalin voidaan olettaa pysyvän likimain stationaarisena.

Oikean pituisen kehyksen valinta on ongelma useimmissa signaalianalyysitehtävissä. Jos kehys on lyhyt (tarkka aikaresoluutio), Fourier-analyysi antaa epätarkan taajuusresoluution, koska signaalin perusjakso ei välttämättä mahdu kehykseen. Jos kehys taas on pitkä, taajuusresoluutio luonnollisesti paranee, mutta aikaresoluution pienenemisen kustannuksella.

### 2.2.3 Diskreetti Fourierin muunnos (DFT)

Olkoon yhden kehyksen pituus $L$. Indeksoidaan kehyksen näytteitä arvoilla 0, … , $L$ - 1. Laskettaessa kehyksen Fourier-muunnosta voidaan ilman rajoituksia olettaa, että kehyksen ulkopuolella signaalin arvot ovat nollia [32]. Tällöin Fourierin muunnos koostuu vain äärellisestä summasta:

$$S(\omega) = \sum_{n=0}^{L-1} s(n) e^{-i\omega n}, \qquad 0 \le \omega \le 2\pi \tag{2.2.3}$$

Vaikka tämä onkin laskennalliselta kannalta huomattavasti mukavampi kuin ääretön summa, on muunnos edelleen taajuusmuuttujan suhteen jatkuva funktio. Seuraava askel kohti käytännöllistä toteutusta on ”näytteistää” taajuusalue sopivalla tasavälisellä jaolla, jolloin pääsemme DFT:n käsitteeseen.

Olkoon $N \ge L$ haluttu taajuusalueen jakopisteiden määrä. Taajuusalueen [0, 2π] näytteistäminen tapahtuu yksinkertaisesti korvaamalla kaavassa 2.2.3 taajuus $\omega$ arvolla $\omega_k = 2\pi k/N$, missä $k$ = 0, … , $N$ - 1. Koska sovimme signaalin nollaksi välin [0, $L$ - 1] ulkopuolella ja $N \ge L$, voidaan summaamisen yläraja vaihtaa $L$ – 1 :stä $N$ – 1 :ksi. Näillä muutoksilla edellinen kaava saadaan muotoon

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-i 2\pi k n / N}, \qquad k = 0, \ldots, N-1 \tag{2.2.4}$$

Relaatiota 2.2.4 sanotaan signaalin $s(n)$ *diskreetiksi Fourierin muunnokseksi* (DFT). DFT:lle on kehitetty laskennallisesti nopea toteutus, ns. *FFT*-algoritmi [32]. FFT:n kompleksisuus on $O(N \log N)$, kun taas kaavan 2.2.4 suoraviivainen soveltaminen vaatii ajan $O(N^2)$. FFT:tä käytettäessä $N$:n tulee olla kakkosen potenssi (esim. $N = 1024$). FFT:n kehittäminen on johtanut Fourierin muunnoksen laajaan käyttöön myös signaalinkäsittelyn ulkopuolella. Tässä yhteydessä meidän ei ole syytä paneutua syvällisemmin FFT:n algoritmiseen toteutukseen, koska useimmiten se on on saatavilla valmiina aliohjelmana.

# 3 PUHEEN TUOTTAMINEN JA HAVAITSEMINEN

Tässä luvussa perehdymme puhetieteessä käytettyihin peruskonsepteihin, jotka luovat pohjaa myöhemmissä luvuissa käsiteltäville asioille sekä yleisemminkin puheenkäsittelyn tutkimuskentälle. Kohdassa 3.1 kertaamme lyhyesti äänen fysikaaliset ominaisuudet. Kohdissa 3.2 - 3.4 käsittelemme ihmisen puheentuottamisprosessia sekä sen matemaattista mallintamista hyödyntäen edellisessä luvussa käsittelemäämme Fourier-analyysia. Viimeisessä kohdassa käsittelemme joitakin psykoakustiikan asioita.

## *3.1 Ääni ja sen ominaisuudet*

Ääni syntyy kun äänen lähde saa väliaineen hiukkaset värähtelemään aaltoliikkeeseen [2]. Hiukkaset siirtävät energiaa eteenpäin muodostaen väliaineeseen tihentymiä ja harventumia, jotka ihminen havaitsee ilmanpaineen vaihteluina. Puhe voidaan siis ajatella monimutkaisina ilmanpaineiden vaihteluina.

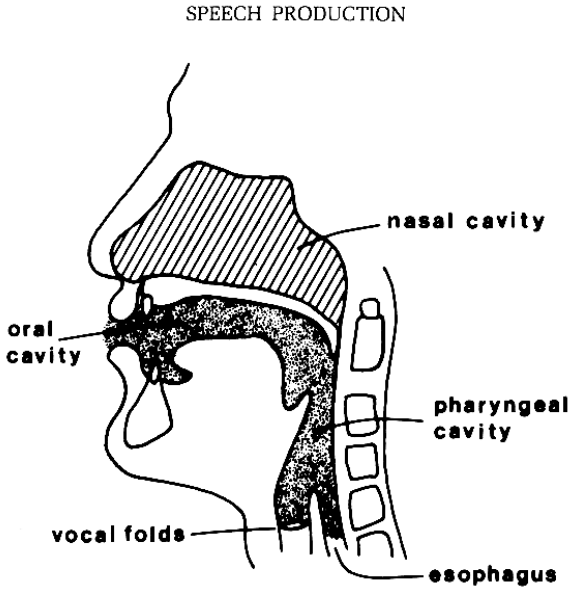
Yksinkertaisin mahdollinen ääni (ns. *puhdas ääni* tai *siniääni*) syntyy kun kappale saatetaan harmoniseen värähdysliikkeeseen ja se värähtelee vain yhdellä ainoalla taajuudella. Siniäänen saa aikaan esimerkiksi ääniraudalla. Käytännössä kaikki luonnossa esiintyvät äänet, myös puhe, muodostuvat kuitenkin useista taajuuskomponenteista. Tästä syystä edellisessä luvussa käsittelemämme Fourier-analyysi tulee erittäin hyödylliseksi apuvälineeksi puheanalyysissa. Koko Fourier-analyysin lähtökohta on, että useammasta taajuuskomponentista muodostuva aalto, ns. *kompleksiaalto* voidaan konstruoida summaamalla yksinkertaisia siniaaltoja, joilla on kullakin oma taajuutensa, vaiheensa ja amplitudinsa [2, 34]. Kompleksinen aalto voi olla joko jaksollinen tai jaksoton.

Äänen matalataajuisinta komponenttia sanotaan äänen *perustaajuudeksi* ja merkitään $F_0$:lla. Jos ääni on jaksollinen, muut taajuudet ovat perustaajuuden monikertoja $kF_0$, $k$=1,2,… ja niitä sanotaan äänen *yläsäveliksi* tai *ylä-ääneksiksi* [34]. Tällöin sanotaan, että taajuudet ovat toisiinsa nähden *harmonisessa suhteessa*. Harmoninen taajuusjakauma on aina jaksollisen äänen tunnusmerkki [34].

Ei-jaksollisen äänen tapauksessa taajuudet eivät ole jakautuneet yhtä säännöllisesti, vaan niitä voi olla mielivaltaisen paljon ja ne voivat jakaantua miten tahansa toisiinsa nähden. Tästä syystä jaksottomasta äänestä on hankalaa erottaa sen korkeutta, koska taajuuskomponenttien suhteet eivät muodosta mitään selvää trendiä. Esimerkkinä puheessa esiintyvästä jaksottomasta äänestä on [s]-äänne. Hakasulkumerkinnällä tarkoitamme tietyn kielellisen yksikön “akustista” ilmentymää.

### *3.2 Puheen tuottaminen*

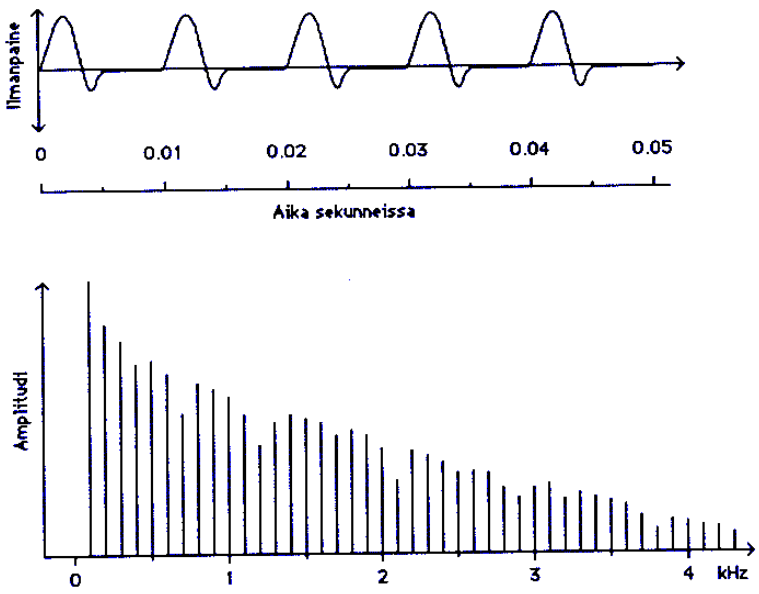
Ihminen tuottaa puhetta näennäisen helposti, mutta itse asiassa tämä prosessi on kaikkea muuta kuin yksinkertainen. Äänen tuottaminen lähtee aivoista, ihmisen aikomuksesta puhua, ja päätyy lopulta kuultavaksi ääneksi, akustiseksi signaaliksi [2, 30]. Välissä tapahtuu kuitenkin paljon muutakin.




**Kuva 3.1** [2]. Ihmisen ääniväylän tärkeimmät osat.

Keuhkoista puhallettavan ilmamassan paine-ero vallitsevaan ilmanpaineeseen verrattuna saa kurkunpäässä sijaitsevat äänihuulet jaksolliseen värähdysliikkeeseen. Äänihuulten väliin jäävä rako, ns. *äänirako* avautuu ja sulkeutuu jaksollisesti päästäen kerrallaan pieniä ilmapulsseja *ääniväylään*, joka muuntaa pulssisarjan ymmärrettäväksi puheeksi [2, 34]. Kuvassa 3.2 on esitetty, millaiselta tällaisen *kurkunpää-äänen* (tai *glottaalisten pulssien*) aaltomuoto näyttäisi, jos sitä voitaisiin mitata ilman ääniväylän muuntovaikutusta. Koska ääni on jaksollista, muut taajuudet ovat harmonisessa suhteessa toisiinsa nähden, mikä havaitaan kuvasta.

Kurkunpään muoto ja koko vaikuttavat kurkunpää-äänen taajuuteen ja taajuusvaihteluihin. Kurkunpää-äänen taajuus määrää suoraan puheen perustaajuuden, joka on miehillä keskimäärin 120 Hz, naisilla 220 Hz ja lapsilla 300 Hz [34]. Puheen muokkauksen kannalta tärkein komponentti on ääniväylä, jonka tärkeimmät osat ovat kuvassa 3.1. Ääniväylä alkaa kurkunpäästä ja päättyy huuliin ja sieraimiin.

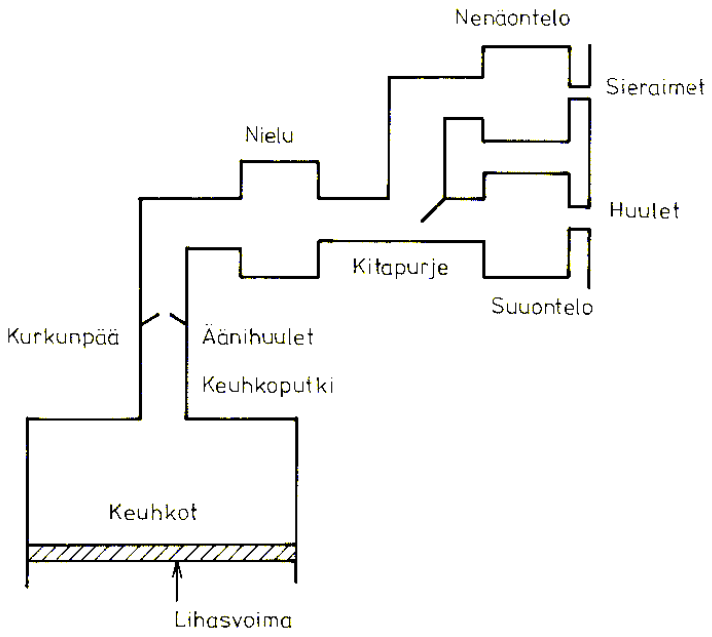



**Kuva 3.2.** [34]. Kurkunpää-äänen aaltomuoto ja spektri.

Karkeasti luokitellen puheessa esiintyy kahdenlaisia äänteitä: *soinnillisia* ja *soinnittomia* [2, 8, 34]. Esimerkkejä soinnillisista äänteistä ovat kaikki vokaalit ja soinnittomista esimerkiksi [s] ja [k]. Kuvaamamme äänentuottoprosessi pätee tarkalleen ottaen vain soinnillisiin äänteisiin. Soinnittomat äänteet muodostetaan pakottamalla jokin ääniväylän osa hyvin ahtaaksi, jolloin ilmamassan kulku tämän kohdan läpi on rajoittunempaa ja ääniväylään muodostuu pyörrevirtauksia, jotka saavat äänen kuulostamaan kohinaiselta [8, 34]. Soinnillinen ääni on jaksollista, soinniton jaksotonta.

### *3.3 Lähde-suodin -malli*

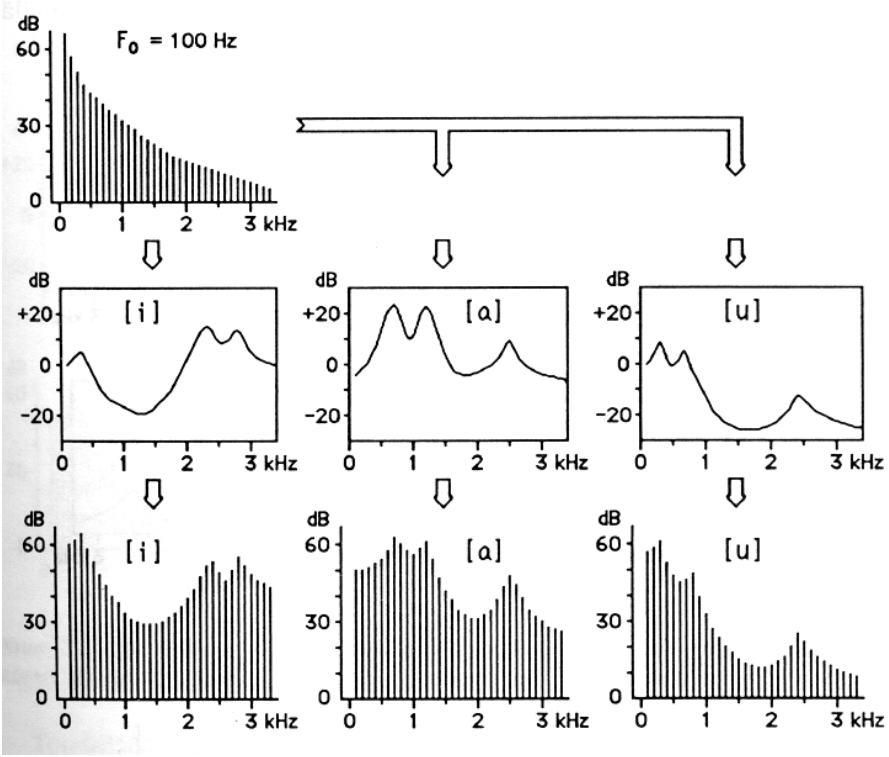
Ääniväylän toimintaa on kuvattu erilaisilla malleilla aina virtapiiriesityksistä akustisiin malleihin, joihin meidän ei ole tarvetta syventyä tässä esityksessä muuten kuin pintapuolisesti. Kuvassa 3.3 on hahmotelman omaisesti ääniväylän ns. *akustinen putkimalli*. Tässä mallissa ääniväylä koostuu erilaisista “kammioista“ ja niitä yhdistävistä “putkista“ joissa ilman tuloa säädellään “venttiileillä“.




**Kuva 3.3.** [3]. Ääniväylän akustinen putkimalli. Tällainen malli riittää monesti kuvaamaan varsin hyvin ääniväylän toimintaa.

Ääniväylän sisällä oleva ilma on jakaantunut erilaisten kammioiden sisälle ja muodostuu joukosta *ilmapatsaita*, jotka resonoivat tietyillä taajuuksilla [34]. Kun äänilähteen aiheuttama ilmavirtaus saapuu ääniväylään, tämän ilmapatsaat alkavat resonoida ominaistaajuuksillaan. *Äänilähteellä* tarkoitamme joko kurkunpää-ääntä (soinnilliset äänteet) tai jaksotonta kohinaa (soinnittomat äänteet). Ääniväylä toimii *suodattimena* äänilähteen tuottamalle äänelle, toisin sanoen se korostaa toisia taajuuksia ja heikentää toisia. Suotimen muuntovaikutus riippuu ääniväylän muodosta ja pituudesta, joita ihminen voi itse säädellä [2, 34]. Spontaanissa puheessa ääniväylän muoto muuttuu koko ajan.

Olemme tulleet kuvanneeksi Fantin [9] esittämän äänentuottomallin, jota kutsutaan *lähde-suodin-malliksi*. Tässä mallissa perusajatus siis on, että äänen lähde ja ääntä muokkaava suodatin ovat toisistaan riippumattomia komponentteja. Mallia on havainnollistettu kuvassa 3.4.




**Kuva 3.4.** [34]. *Lähde-suodin –malli*. Äänilähteenä on perustaajuudeltaan 100 Hz:n suuruinen kurkunpää-ääni (ylhäällä). Keskellä nähdään erilaisia suodattimia. Äänilähteen spektri suodattuu alimmaisten kuvien mukaiseksi. Huomaa erityisesti suodattimien spektreissä olevat lokaalit maksimit. Nämä taajuudet ovat ääniväylän resonanssitaajuuksia ja niitä sanotaan *formanteiksi*.

### *3.4 Äänentuottomallin formulointi*

Puheen ajatellaan siis muodostuneen siten, suodatin (ääniväylä) muokkaa äänilähteen (glottaalinen lähde tai kohina) muodostamaa ääntä. Lisäksi huulet aiheuttavat äänen *säteilyä* vapaaseen kenttään. Usein huulet voidaan kuitenkin tulkita osaksi ääniväylää, jolloin niiden aiheuttamaa akustista säteilyvaikutusta ei tarvitse mallintaa erikseen. Puheentuottomallia on havainnollistettu vuokaaviona kuvassa 3.5. Malli voidaan kirjoittaa Fourier-muunnosten avulla seuraavaan muotoon

$$S(\omega) = E(\omega)H(\omega)R(\omega). \qquad (3.4.1)$$

Käsiteltäessä pelkästään amplitudispektrejä tämä kirjoitetaan itseisarvoja käyttäen muotoon

$$|S(\omega)| = |E(\omega)|\,|H(\omega)|\,|R(\omega)|, \tag{3.4.2}$$

missä:

$S(\omega)$ on puhesignaalin Fourier-muunnos,

$E(\omega)$ on äänilähteen (glottaalinen lähde tai kohina) Fourier-muunnos,

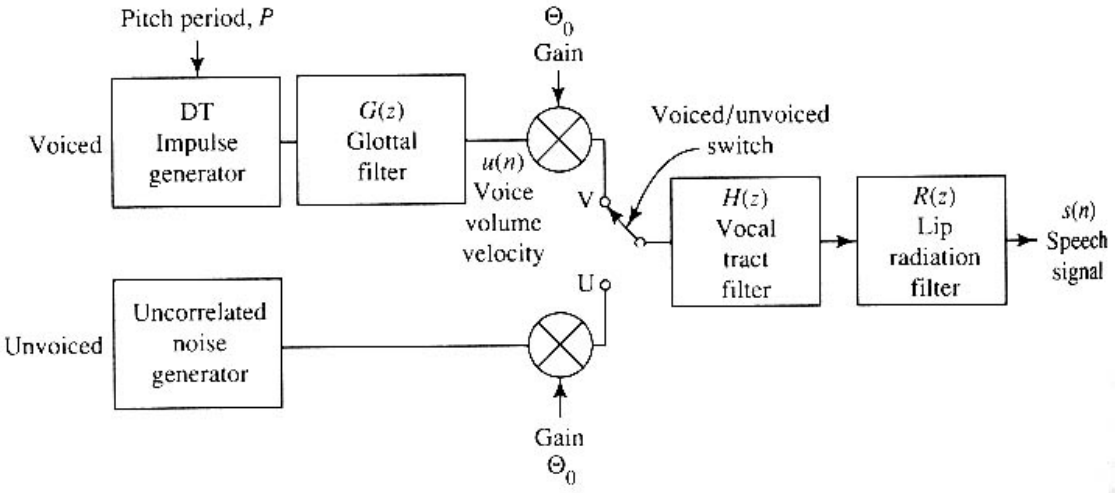
$H(\omega)$ on suodattimen (ääniväylän) Fourier-muunnos ja

$R(\omega)$ huulien aiheuttaman säteilyn Fourier-muunnos.

Relaatio 3.4.1 voidaan esittää aika-alueessa konvoluution avulla seuraavasti:

$$s(n) = e(n) * h(n) * r(n), \tag{3.4.3}$$

missä $*$ on konvoluutio-operaattori. Tämän esityksen puitteissa meidän ei ole tarpeellista paneutua syvällisemmin konvoluution ominaisuuksiin. Konvoluutio on eräänlainen kahden signaalin välinen yleistetty "tulo". Konvoluution ja Fourierin muunnoksen välillä on kuitenkin seuraava hyödyllinen yhteys: kahden aikatasossa esitetyn signaalin konvoluutio on sama kuin signaalien Fourier-muunnosten tulo [32]. Tämä pätee myös kääntäen: kahden signaalin Fourier-muunnosten konvoluutio on näiden signaalien tulo aikatasossa. Käytimme jälkimmäistä ominaisuutta esittäessämme juuri taajuusalueen kertolaskun aika-alueen konvoluutiona.




**Kuva 3.5.** [8]. Äänentuottomallin vuokaavioesitys. Äänen lähteenä on joko sarja glottaalisia pulsseja (soinnilliset äänteet) tai kohina (soinnittomat äänteet). Ääniväylä ja huulet muodostavat peräkkäisen suodatinrakenteen, joka muokkaa lähdesignaalin puhesignaaliksi $s(n)$.

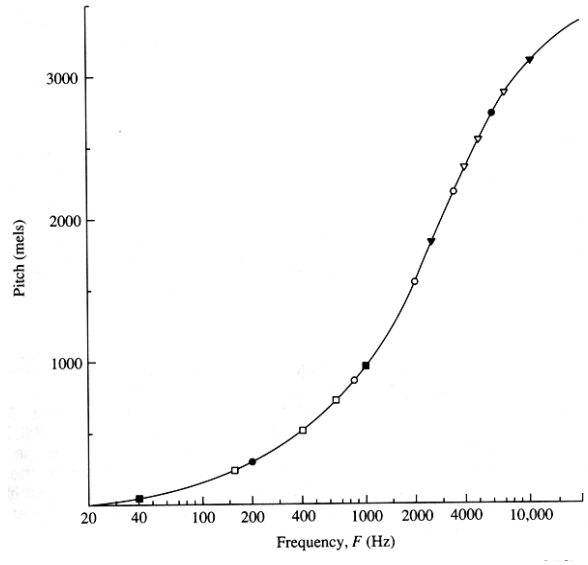
### *3.5 Psykoakustiikkaa*

Puhekommunikaatiossa on aina kaksi osapuolta: puheen lähettäjä eli puhuja ja sen vastaanottaja eli kuulija. *Psykoakustiikka* on tieteenala, joka tutkii ihmisen äänen havaitsemismekanismia [8]. Psykoakustiikassa ollaan kiinnostuttu mm. korvan toiminnasta, äänen suunta-aistimuksesta, puheen ymmärrettävyyteen vaikuttavista akustisista tekijöistä, perustaajuuden havaitsemisesta jne. Tässä kohdassa käsittelemme muutamia psykoakustiikan asioita, joita tulemme hyödyntämään tulevissa luvuissa.

Äänen korkeuden ja taajuuden käsitteet sekoitetaan usein keskenään. Taajuus on fysikaalinen suure, kun taas äänen korkeus on psykologinen mitta [2, 8]. Äänen *korkeus* tarkoittaa ihmisen *havaitsemaa* äänen perustaajuutta [8]. Taajuuden ja äänenkorkeuden välistä relaatiota kuvataan ns. *mel-asteikolla,* joka on esitetty kuvassa 3.6. Mel-asteikko määritettiin aikoinaan puhtaasti psykologisten koejärjestelyjen avulla ja sille on esitetty joukko matemaattisia approksimaatioita. Esimerkiksi Fant ehdottaa seuraavaa muotoa olevaa approksimaatiota [10]:

$$F_{mel} = k \log_{10}(1 + \frac{F_{Hz}}{1000}) \tag{3.5.1}$$

Taajuuden 1000 Hz äänenkorkeudeksi on sovittu 1000 mel-yksikköä. Mel-asteikko on likimain lineaarinen 1000 Hz:iin asti ja siitä ylöspäin logaritminen. Tämän tulkinta on siis, että 1000 Hz:iin asti ihminen havaitsee sävelkorkeudessa lineaarisen muutoksen taajuuteen nähden. Tämän yli menevissä taajuuksissa taajuuden täytyy muuttua yhä enemmän, jotta ihminen havaitsisi eron äänen korkeudessa. Puheen- ja puhujantunnistusmenetelmissä usein spektrianalyysin tuottamille taajuuksille tehdään jatkokäsittelyä mel-asteikkoa hyödyntäen, sillä sen avulla voidaan pienentää suuridimensioisen taajuusasteikon dimensiota siten, ettei tunnistuksen kannalta olennaista informaatiota kadoteta [30]. Lisäksi mel-asteikon käyttäminen usein parantaa tunnistustuloksia! [37].




**Kuva 3.6.** [8]. Mel-asteikko, joka kuvaa (perus)taajuuden ja äänenkorkeuden välistä yhteyttä. Huomaa, että taajuusakseli on logaritminen.

Psykoakustiset mittaukset ovat osoittaneet, että tietyn taajuuden havaitsemiseen vaikuttaa tämän taajuuden tietyssä ympäristössä, ns. *kriittisellä kaistalla* [8] olevat muut taajuudet. Kriittisen kaistan leveys ei ole vakio, vaan se riippuu taajuudesta. Alle 1 kHz:n taajuuksille kriittinen kaistanleveys on likimain vakio, n. 100 Hz, ja tätä suuremmille taajuuksille kaistanleveys kasvaa logaritmisesti [8].

# 4 PIIRREIRROITUS

*Piirreirroitus* on kaikille hahmontunnistustehtäville yhteinen vaihe, jossa signaalista määritetään sitä karakterisoivat parametrit, joita käytetään hahmojen luokitteluun [11]. Puhujan tunnistuksessa piirteiden tulisi olla sellaiset, joiden hajonta samalla puhujalla olisi mahdollisimman pieni mutta joiden poikkeama eri puhujan piirteistä olisi mahdollisimman suuri [11, 30]. Tässä luvussa käymme läpi yksityiskohtaisesti tyypillisen puhujan tunnistamisessa käytetyn piirreirroitusprosessin.

Periaatteessa digitaalisen puhesignaalin näytearvot kuvaavat tietyn puhujan yksilöllisiä ominaisuuksia, jolla hänet voidaan erottaa muista. Käytännössä raakaa signaalia ei kuitenkaan käytetä tunnistamisessa, koska laskennallinen kompleksisuus kasvaa silloin hyvin suureksi. Raaka signaali sisältää paljon toistuvaa ja epäolennaista informaatiota [30], joka voidaan poistaa siitä tunnistusprosessin kärsimättä. Yleensä käy vieläpä niin, että piirreirroitus ei pelkästään *helpota*, vaan *parantaa* tunnistusprosessia, koska tunnistusta häiritsevät epäolennaisuudet poistetaan signaalista.

## *4.1 Piirteiden valinta*

Kaksi eri puhujaa kuulostavat erilaisilta, koska akustisen signaalin taajuuskomponenttien voimakkuus- ja korkeussuhteet ovat jakautuneet eri tavalla [34]. Mutta miksi näin on? Edellisen luvun tietojen perusteella tiedämme ainakin, että eri ihmisillä on jo *fysiologisesti* erilaiset äänentuottomekanismit. Äänen väriin vaikuttavia fysiologisia tekijöitä ovat mm. kurkunpään koko sekä ääniväylän onteloiden tilavuudet ja muodot. Fysiologisiin piirteisiin ihminen ei voi itse vaikuttaa. Fysiologisten tekijöiden lisäksi puhujan tunnistettavuuteen vaikuttavat kuitenkin myös muut tekijät, kuten painotus, intonaatio, rytmiikka ja puhenopeus. Näihin ns. *prosodisiin* tai *suprasegmentaalisiin* piirteisiin [1, 8, 19] ihminen voi itse ainakin osittain vaikuttaa.

Piirreirroituksen tavoitteena olisi siis selvittää puhesignaalista edellä mainittuja piirteitä. Voidaan heti arvata, ettei tämä ole helppo tehtävä. Tulemme soveltamaan kahdessa edellisessä luvussa kuvaamiamme matemaattisia menetelmiä ja äänentuottomalleja intensiivisesti tässä luvussa. Pääpaino on ns. *kepstrin* määrittämisessä, sillä tätä ja sen eri johdannaisia on sovellettu menestyksekkäästi sekä

puheen- että puhujan tunnistuksessa. Yleisesti kepstrin on myönnetty olevan – ellei paras – niin ainakin ”yksi parhaimmista” piirreparametrijoukoista sekä puheen- että puhujan tunnistuksessa.

### *4.2 Puhujan tunnistuksen piirreirroitus algoritmimuodossa*

Jotta kokonaiskuva piirreirroituksesta pysyisi koossa, kuvaamme ensin prosessin karkealla tasolla ennen siirtymistä yksityiskohtiin. Muutamien esisuodatusten jälkeen signaali jaetaan kiinteän mittaisiin kehyksiin, jotka ovat pituudeltaan 20 ms:n luokkaa. Yleensä vierekkäiset kehykset peittävät osittain toisiaan, tavallisesti 50 % verran. Kehykset käsitellään seuraavalla algoritmilla:

ALGORITMI 4.2.1

**FOR EACH** kehys **DO**
(1) Kerro kehys sopivalla ikkunafunktiolla spektrin pehmentämiseksi;
(2) Laske ikkunoidun kehyksen spektri DFT:tä käyttäen;
(3) Laske DFT:n itseisarvon logaritmi;
(4) Suodata log-spektri mel-asteikon mukaisesti sijoitetuilla kolmiosuotimilla;
(5) Laske suodattimien ulostuloista diskreetti kosinimuunnos;
**END FOR**;

Tätä muotoa olevaa algoritmia ovat käyttäneet puhujan tunnistukseen mm. Olsen [30] sekä Brunelli ja Falavigna [4]. Puheentunnistukseen algoritmia ovat soveltaneet esimerkiksi Gagnoulet & al. [17] sekä Young [37]. Seuraavissa alakohdissa täsmännämme ja perustelemme algoritmin jokaista askelta.

### *4.3 Signaalin esikäsittely*

Signaalille joudutaan tekemään lähes aina ennen Fourier-analyysia jonkinasteista esikäsittelyä. Mikäli näytteenottotaajuus on liian pieni, spektrin energiajakauma vääristyy aliasing-ilmiön takia, koska tällöin korkeimmat taajuudet esiintyvät digitaalisessa signaalissa matalempina taajuuksina. Mikäli näytteenottotaajuutta ei ole mahdollista nostaa, aliasing-ilmiö voidaan eliminoida ajamalla signaali ennen A/D-muunnosta analogisen alipäästösuodattimen läpi [32]. Tällainen *anti-aliasing*-suodattimena toimiva suodin vaimentaa jyrkästi tai poistaa kokonaan tietyn kynnystaajuuden ylittäviä taajuuksia.

Puhujan tunnistamisessa aliasingin poistaminen alipäästösuodattimella johtaa kuitenkin erääseen paradoksiin. Puheentunnistuksessa alipäästösuodatus ei tuota ongelmaa, sillä äänteiden tunnistamisen kannalta lingvististä informaatiota on enemmän matalilla taajuuksilla [30]. Sen sijaan puhujan tunnistusta tutkittaessa on havaittu, että taajuusalue 4-10 kHz sisältää yhtä paljon informaatiota puhujan identiteetistä kuin kaista 0-4 kHz:kin [30]. Tähän ilmiöön on saattanut itse kukin törmätä vaikkapa puhelinkeskustelussa. Voi olla hankalaa tunnistaa aluksi, kuka linjan toisessa päässä puhuu, ellei tämä sitä itse kerro. Puhelinverkko on kaistarajoitettu välille n. 0.3 - 4.3 kHz [30, 32].

Koska puheen energiajakauma on vahvasti painottunut matalille taajuuksille ja korkeat taajuudet kuitenkin sisältävät olennaista tietoa puhujan henkilöllisyydestä, suoritetaankin näytteistetylle signaalille usein ylipäästösuodatus. Ylipäästösuodin on alipäästösuotimen “vastakohta”: se vaimentaa tietyn kynnystaajuuden *alittavia* taajuuksia halutussa suhteessa. Youngin [37] mukaan ylipäästösuotimen tarkoitus on kompensoida huulten säteilyvaikutuksen aiheuttamaa korkeiden taajuuksien vaimenemista. Monesti käytetään suodinta, jonka siirtofunktio[1] on muotoa

$$H(z) = 1 - az^{-1}, \tag{4.3.1}$$

missä $a$ on lähellä ykköstä oleva vakio. Tätä muotoa olevaa suodatinta ovat käyttäneet esimerkiksi [6, 15, 24]. Muotoa 4.3.1 oleva suodin voidaan toteuttaa seuraavalla suoraviivaisella kaavalla [25]:

$$\tilde{s}(n) = s(n) - as(n-1), \tag{4.3.2}$$

missä merkintä $\tilde{s}(n)$ tarkoittaa suodatettua signaalia.

[1] Tässä yhteydessä meidän ei ole syytä paneutua siihen, mitä siirtofunktiolla tarkoitetaan. Lisätietoa suotimista ja niiden suunnittelusta saa yleisistä signaalinkäsittelyn kirjoista, kts. esim. [33].

### *4.4 Kehyksiin jako ja ikkunointi*

Jo useaan kertaan on tullut esille, että puheanalyysissa puhesignaali jaetaan *kehyksiin* (frame), joita käsitellään itsenäisinä signaaleinaan. Oikean pituisen kehyksen valinta on oleellinen kysymys. Jos kehys on liian lyhyt, DFT antaa huonon taajuuserottelun, koska etsittyä taajuutta vastaava aallonpituus ei välttämättä mahdu kokonaan kehykseen. Jos kehyksen pituutta kasvatetaan, tällöin eri taajuuksien erottelu on tarkempi, mutta hintana on aikaresoluution huononeminen. Varsin hiljattain kehitetty matemaattinen menetelmä, *väreanalyysi* [7], antaa eräänlaisen kompromissiratkaisun taajuus/aikaresoluutio-ongelmaan, mutta tämän esityksen puitteissa emme voi lähteä syvemmälle matkalle väreiden maailmaan.

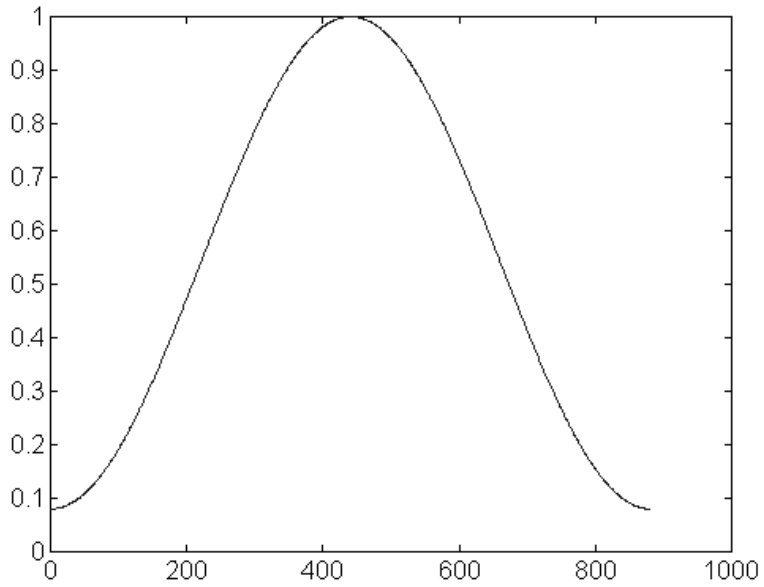
Kehyksen pituuden lisäksi Fourier-analyysissa on kiinnitettävä huomiota myös siihen, mitä *ikkunafunktiota* käytetään. Ikkunafunktiolla tarkoitamme jotakin funktiota, jolla alkuperäinen kehys kerrotaan. Toisin sanoen ikkunafunktiolla painotetaan alkuperäistä kehystä. Hyvällä ikkunafunktiolla kertominen pehmentää DFT:n antamaa spektriä siten, ettei siitä kuitenkaan katoa oleellista informaatiota.

Edellä olemme käsitelleet kehyksen käsitettä hiukan epätäsmällisesti määrittelemällä sen alkuperäisen signaalin lyhyeksi osasignaaliksi. Määritellään kehys nyt uudelleen ikkunafunktiolla kerrotuksi osasignaaliksi. Yksinkertaisin ikkunafunktio on *suorakulmainen ikkuna* [8, 32]:

$$w_{\text{rect}}(n) = \begin{cases} 1 & , 0 \le n \le N - 1 \\ 0 & , \text{muualla} \end{cases} \tag{4.4.1}$$

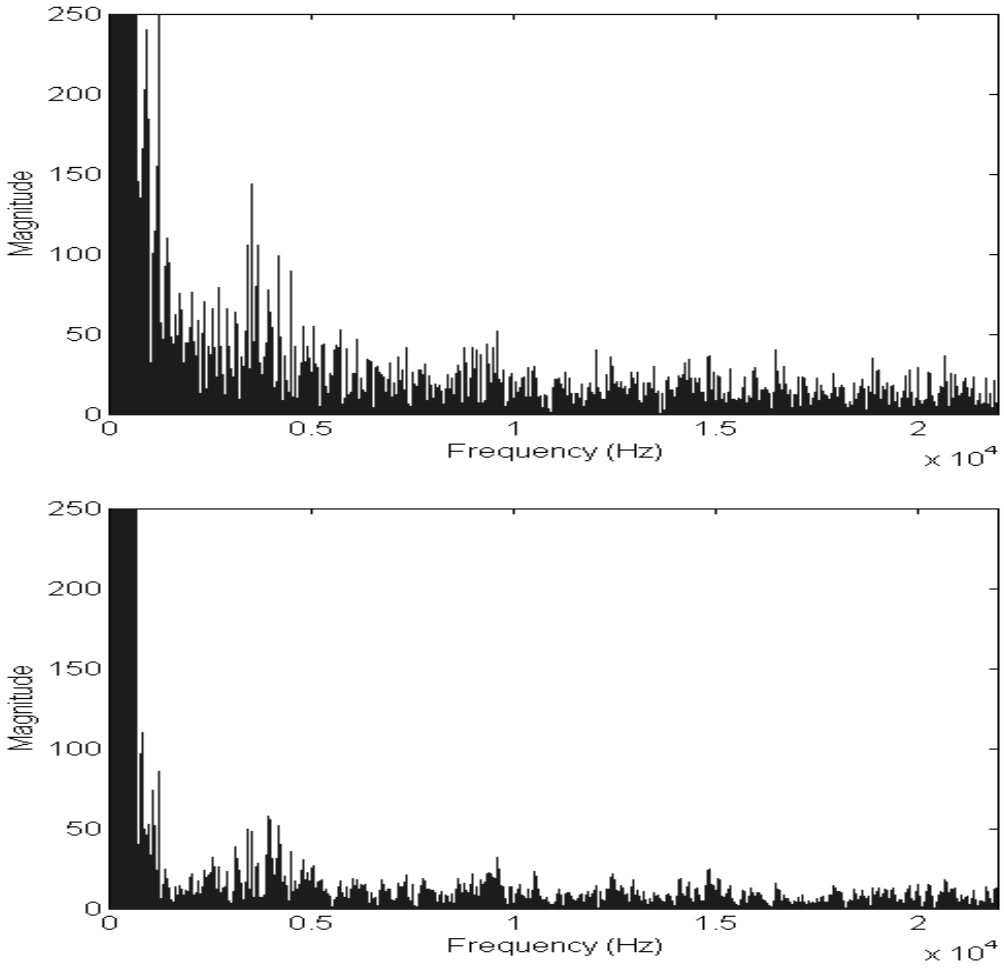
Tällä ikkunalla kertominen säilyttää alkuperäisen kehyksen sellaisenaan. Tavallisesti tätä ikkunafunktiota ei kuitenkaan käytetä, koska se tekee spektristä kovin kohinaisen. Signaalinkäsittelyn piirissä on kehitetty monenlaisia ikkunafunktioita, joilla on paremmat tasoitusominaisuudet kuin suorakulmaisella ikkunalla. Tässä yhteydessä esittelemme erään yleisimmin käytetyistä, *Hammingin ikkunan*. Hammingin ikkuna määritellään seuraavasti [8, 32]:

$$w_{\text{hamm}}(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & , 0 \le n \le N - 1 \\ 0 & , \text{muualla.} \end{cases} \quad (4.4.2)$$




**Kuva 4.1.** Hammingin ikkunan aaltomuoto.

Hammingin ikkunan aaltomuoto on esitetty kuvassa 4.1. Hammingin ikkuna, kuten yleensä muutkin ikkunafunktiot, on symmetrinen kehyksen puolivälin suhteen. Suorakulmaisella ikkunalla on epäjatkuvuuskohdat päätepisteissään, kun taas Hammingin ikkuna pienenee tasaisesti kohti nollaa. Hamming-ikkunoinnin pääasiallinen tarkoitus on tasoittaa spektrissä esiintyviä “häiriöitä”. Kuvassa 4.2 nähdään, kuinka eri ikkunafunktiot vaikuttavat spektriin.




**Kuva 4.2.** Saman kehyksen amplitudispektri laskettuna kahta eri ikkunafunktiota käyttäen. Ylhäällä on suorakulmaisen ikkunan, alhaalla Hammingin ikkunan antama spektri. Hammingin ikkuna pehmentää spektriä ja poistaa siitä "häiriöpiikit" säilyttäen kuitenkin oleellisen informaation.

### *4.5 Kepstrianalyysi*

Nyt olemme käyneet läpi algoritmin 4.2.1 kaksi ensimmäistä askelta. Sinänsä spektrien amplitudeja voitaisiin jo käyttää piirrevektoreiden alkioina, mutta pian osoittautuu, että lisämuunnoksilla piirrevektorien dimensiota saadaan vielä paljon pienemmäksi.

Fourier-analyysi on hyvä signaalianalyysin apuväline silloin kun taajuus on useampien taajuuksien lineaarikombinaatio. On kuitenkin tilanteita, jolloin taajuuksien väliset riippuvuudet eivät ole lineaarisia. Muistamme edellisestä luvusta, että puheen syntyminen lähde-suodin-mallin mukaisesti esitettynä on epälineaarinen prosessi: puhesignaali on kahden toisistaan riippumattoman signaalin konvoluutio. *Kepstrianalyysi* tarjoaa työkalun luonteeltaan epälineaarisesti syntyneiden signaalien

analysointiin [8]. Pohjimmiltaan kepstrin laskeminen on dekonvoluutio-operaatio, joka myös dekorreloi sille “syötteenä” annettavia arvoja [30].

### 4.5.1 Kepstrin määritelmä

Olkoon puhesignaali $s(n)$ jonkin ”nopeasti vaihtelevan” signaalin $e(n)$ (äänilähteen) ja ”hitaasti vaihtelevan” signaalin $h(n)$ (ääniväylän) konvoluutio eli

$$s(n) = e(n) * h(n) \tag{4.5.1}$$

Nyt esitämme kysymyksen ”kuinka saamme konvoloidusta signaalista selville nämä kaksi erillistä komponenttia?”. Tähän vastaus kuuluu: kepstrin avulla. Signaalin $s(n)$ *reaalinen kepstri* (RC) määritellään formaalisti seuraavalla relaatiolla [8]:

$$c_s(n) = \mathsf{T}^{-1}\{\log | \mathsf{T}\{s(n)\}| \}, \tag{4.5.2}$$

missä lineaariset operaattorit $\mathsf{T}$ ja $\mathsf{T}^{-1}$ tarkoittavat DFT:tä ja sen käänteismuunnosta. Teemme siis ensin signaalille Fourier-muunnoksen, otamme sen amplitudispektristä logaritmin ja siitä käänteisen DFT:n.

Tarkastellaan aluksi sisemmän operaation $\log | \mathsf{T}\{s(n)\}|$ merkitystä. Otetaan käyttöön apumerkintä $C_s(\omega) = \log |\mathsf{T}\{s(n)\}|$. Edellisestä luvusta palautamme mieleen, että puhesignaalin spektri $|S(\omega)|$ voidaan esittää muodossa

$$|S(\omega)| = |E(\omega)|\,|H(\omega)| \tag{4.5.3}$$

Kun tästä otetaan logaritmi, saadaan

$$\begin{aligned} \log(|S(\omega)|) &= \log(\,|E(\omega)|\,|H(\omega)|\,) \\ &= \log|E(\omega)| + \log|H(\omega)| \end{aligned} \tag{4.5.4}$$

Mitä tämä tarkoittaa? Olemme muuntaneet signaalien tulon kahden uuden “signaalin” summaksi eli linearisoineet esitysmuodon. Koska nyt olemme tekemisissä kahden “signaalin” lineaarikombinaation kanssa, voimme soveltaa Fourier-analyysin tekniikoita tässä uudessa alueessa. Helposti nähdään, että $C_s(\omega)$ on reaalinen, parillinen ja $2\pi$-jaksollinen funktio. Jaksollisuudesta seuraa, että $C_s(\omega)$:lle kannattaa yrittää löytää “viivaspektri”, jonka “taajuudet” ovat Fourier-sarjan kertoimet $\alpha_n$ [8]:

$$\alpha_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} C_s(\omega)e^{-i\omega n}d\omega \tag{4.5.5}$$

Kirjoitettaessa määritelmä 4.5.2 auki ja verrattaessa tätä Fourier-sarjan kertoimiin havaitaan pienellä vaivalla käyttäen hyväksi $C_s(\omega)$:n parillisuutta, että nämä ovat yksi ja sama asia. Näin ollen kepstri voidaan ajatella “signaalin” $C_s(\omega)$ “viivaspektrinä”.

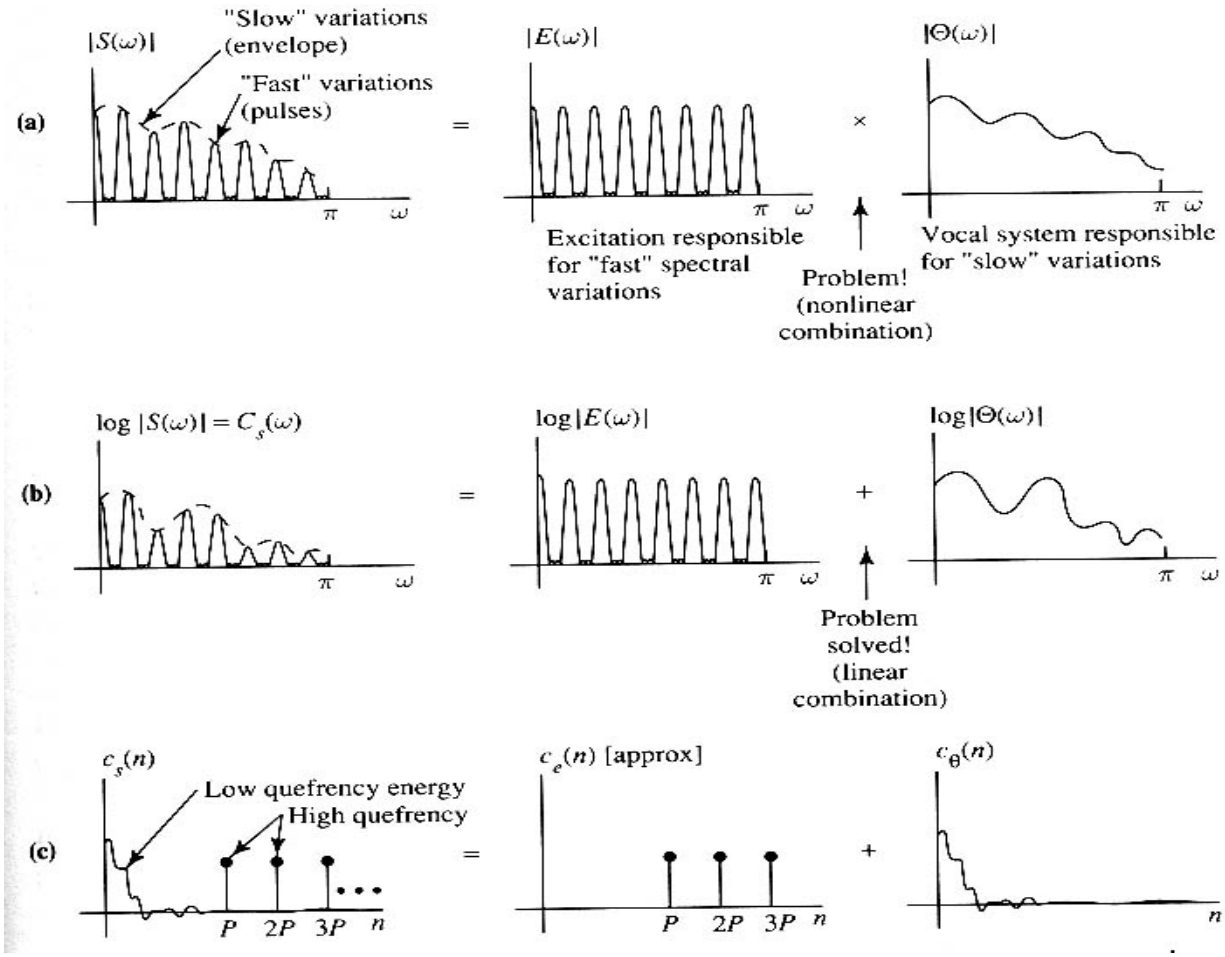
Jokaista kehystä kohti saadaan siis joukko kepstraalisia kertoimia. Kertoimien voidaan ajatella kuvaavan muutoksia *spektrissä* eli kepstri sisältää tietoa tietyn kehyksen “spektraalisesta muodosta”. Tämä taasen sisältää tietoa fysiologisista piirteistä, ja myös prosodisista piirteistä, mikäli käsitellään useampaa kuin yhtä kehystä. Eri äänteille spektrin muoto on tietenkin erilainen. Riittävän suurella opetusdatalla luokittelija oppii implisiittisesti eri äänteisiin liittyvien spektrien “muodot”, jotka ovat erilaiset eri puhujilla. Tämä on kepstriin perustuvan tunnistuksen idea, mutta siihen palaamme luokittelun yhteydessä tarkemmin.

### 4.5.2 Mel-kepstri

Kepstriä voidaan käyttää jo sellaisenaan kuvaamaan kunkin puhujan yksilöllisiä piirteitä. Kepstri dekorreloi amplitudispektriä, mutta laskettaessa kertoimet erikseen jokaisessa spektrin pisteessä saadaan $N$ kepstraalista kerrointa, eikä piirreavaruuden dimensio siis pienene. Tässä vaiheessa kuvaan tulevat edellisessä luvussa esittämämme psykoakustiikan opit.

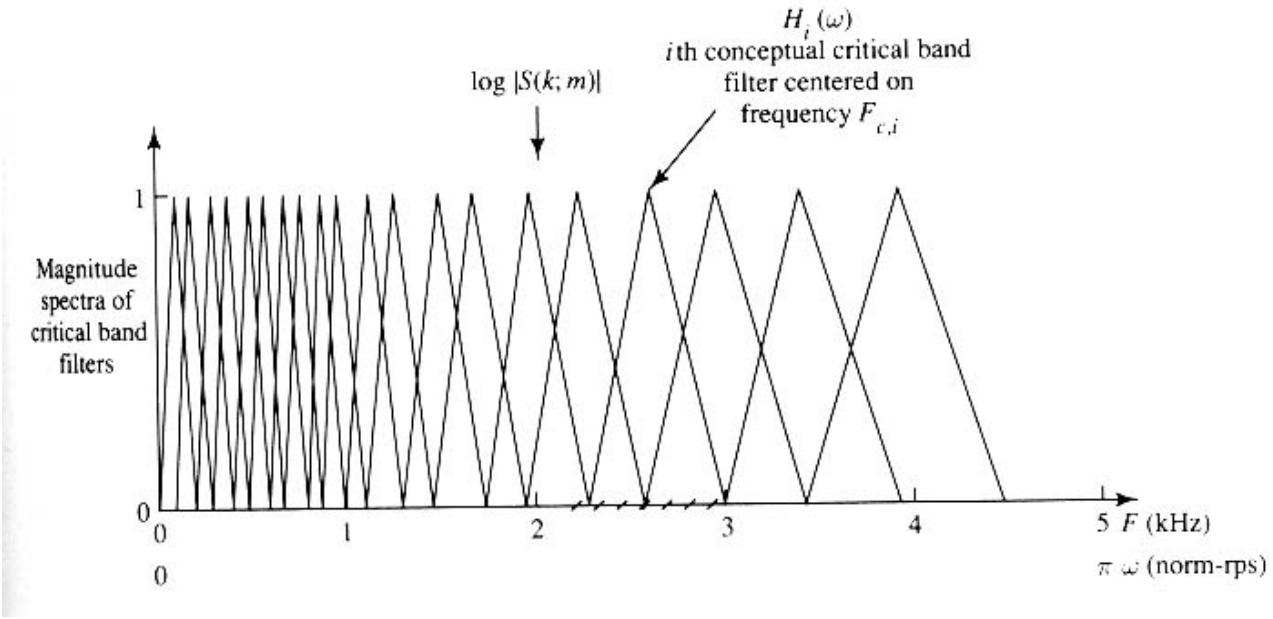
Koska tietyn taajuuden havaitsemiseen vaikuttavat tämän taajuuden kriittisellä kaistalla olevat muut taajuudet, intuitiivisesti tuntuisi, että kutakin kriittistä kaistaa kohti riittäisi spektrissä yksi ainoa

amplitudi, joka laskettaisiin kriittisen kaistan taajuuksista sopivasti painotetulla keskiarvolla. Lisäksi hyödynnämme tietoa, että ihmiskorva on herkempi taajuuksien muutoksille matalilla taajuuksilla. Relaatiota taajuuden ja äänenkorkeuden välillä kuvasi siis mel-asteikko.




**Kuva 4.3.** [8]. Kepstrin intuitiivinen idea. Kahden signaalin konvoluutio muuntuu uudessa alueessa kahden uuden signaalin summaksi, josta saadaan Fourier-analyysin tekniikoilla kummankin alkuperäisen signaalin ”muotoparametrit” erilleen.

Sen sijaan, että laskisimme suoraan kepstraaliset kertoimet spektrin amplitudeista, asetamme ensin taajuusasteikolle mel-asteikon mukaisesti kolmiosuodattimia, jotka simuloivat kriittisten kaistojen sisältämää *logaritmista kokonaisenergiaa* [8]. Kukin suodin saa arvon 1 keskikohdassaan ja pienenee lineaarisesti nollaan siten, että suodattimen “leveys” on kriittinen kaistanleveys suodattimen keskitaajuuden suhteen. Koska mel-arvo ja kriittinen kaistanleveys kasvavat taajuuden myötä, on korkeammilla taajuuksilla vähemmän suotimia ja niiden leveys on suurempi. Kuvassa 4.4 on havainnollistettu kolmiosuodatinten sijoittamista taajuusasteikolle.




**Kuva 4.4.** [8]. Ihmisen kuulojärjestelmää simuloivien kolmiosuodatinten sijoitus taajuusasteikolle mel-asteikon mukaisesti.

*Mel-kepstri* [8, 30, 37] lasketaan seuraavasti: jokaista kriittistä kaistaa kohti lasketaan yksi arvo, joka saadaan painotettuna keskiarvona kaistan sisältämistä taajuuksista. Painofunktiona käytetään edellä kuvattua kolmiosuodatinta. Näistä arvoista otetaan logaritmi ja siitä käänteinen Fourierin muunnos. Proseduuri on siis muuten aivan samanlainen kuin tavallisen kepstrin määrittämisessäkin, mutta kertoimet lasketaan suodatinten ulostuloista eikä kaikista spektrin amplitudeista. Tämä muunnos on paljon käytetty puheanalyysisovelluksissa, koska se pienentää merkittävästi piirteiden määrää ja usein myös parantaa tunnistusta [37].

Algoritmin 4.2.1 viimeisessä askeleessa mainitaan *diskreetti kosinimuunnos* (DCT). Tämä muunnos määritellään suodatinten ulostuloille $S_k$, $k = 1 , \ldots , K$ seuraavasti [30]:

$$c_n = \sum_{k=1}^{K} (\log S_k) \cos\left[ n(k - \frac{1}{2}) \frac{\pi}{k} \right], \qquad n = 0, \ldots , M, \tag{4.5.6}$$

missä $M$ haluttu kepstraalisten kerrointen määrä. Yleensä valitaan $M \approx K/2$ [30], jolloin alkuperäisen piirreavaruuden dimensio on pienentynyt todella merkittävästi. Yleensä kerroin $c_0$ jätetään huomiotta

[14], koska se vastaa signaalin kokonaisenergiaa eikä siis sisällä tietoa puhujasta itsestään. Kerrointa voidaan käyttää kuitenkin vaikka piirteiden normalisointiin.

### 4.5.3 LP-kepstri

Edellä olemme tarkastelleet kepstrin johtamista määritelmän mukaan DFT-spektristä. Kepstrin määrittämiseen on kuitenkin toinenkin menetelmä, jossa kertoimet määritetään suoraan käymättä ollenkaan taajuusalueessa. Tämä menetelmä perustuu ns. *lineaariprediktioon* (LP) [8, 35]. Lineaariprediktiossa signaalin arvo $s(n)$ esitetään edellisten arvojen lineaarikombinaationa [11]:

$$s(n) = \sum_{k=1}^{N} a_k s(n-k), \tag{4.5.7}$$

missä kertoimet $a_k$ ovat *LP-kertoimet*. Kertoimet määrätään siten, että prediktiovirheiden neliösumma minimoituu [8, 35]. Nämä kertoimet voidaan määrittää useallakin tavalla, esimerkiksi ns. *autokorrelaatio-* tai *kovarianssimenetelmiä* käyttäen [8, 35]. Näihin menetelmiin emme tässä esityksessä mene tarkemmin.

Lineaariprediktion perusidea on approksimoida puheen tuottamista rekursiivisella suodatinrakenteella [35], jonka kertoimet pyritään määrittämään. Kepstri johdetaan LP-kertoimista edelleen käyttäen seuraavaa rekursiokaavaa [11, 15]:

$$\begin{aligned} c_1 &= a_1 \\ c_n &= \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) a_k c_{n-k} + a_n, \qquad 1 < n < M, \end{aligned} \tag{4.5.8}$$

missä $M$ on haluttu kerrointen lukumäärä. LP-analyysin tuloksena saatava kepstri ei ole täsmälleen sama kuin spektristä johdettu, mutta ei ole selkeätä näyttöä, että jompikumpi tapa olisi tarkempi tunnistuksen kannalta. LP-kepstrin laskeminen on kuitenkin nopeampi kuin kahden FFT:n laskeminen [15], mistä syystä LP-kepstri on enemmän käytetty. LP-kepstriä puhujan tunnistukseen ovat käyttäneet esimerkiksi [11, 14, 15, 24, 38]. Myös pelkkiä LP-kertoimia on käytetty piirteinä

[33]. Furui [15, 16] muistuttaa kuitenkin, että pelkät LP-kertoimet ovat herkkiä spektrissä oleville häiriöille, joten niiden sijasta pitäisi käyttää mieluummin kepstriä.

Furui [16] mainitsee, että myös ensimmäisen ja toisen kertaluvun differentioitua kepstriä voidaan käyttää piirteinä puheanalyysissa. Nämä kertoimet, joita kutsutaan *delta-* ja vastaavasti *delta-delta-kertoimiksi*, kuvaavat spektrin dynaamisia muutoksia [8, 16]. Toisinaan käytetään myös eri piirteiden yhdistelmiä, esimerkiksi kepstri + delta-kepstri tai kepstri + äänen korkeus.

### *4.6 Muut lähestymistavat*

Kepstri ja sen edellä kuvatut johdannaiset ovat selkeästi eniten käytetty piirreparametrijoukko puhujan tunnistuksessa. Tämä johtuu faktasta, että kepstri kuvaa spektrin olennaisimpia *muotoja*, joiden voidaan olettaa pysyvän samalla henkilöllä tietyn äänteen kohdalla lähes samoina toistokerrasta riippumatta. Furui [15] viittaa lisäksi tutkimukseen, jossa on osoitettu, että kepstristä voidaan aina johtaa uusi piirrejoukko, joka on invariantti kiinteille taajuusvasteen häiriöille! Toisin sanoen, jos taajuusvastetta vääristää koko ajan sama suodin (esimerkiksi siirtokanava), ei tunnistuksen pitäisi ainakaan teoriassa heiketä.

Kepstrin eräänä heikkoutena mainitaan mm. että mikäli opetus- ja testiolosuhteet ovat erilaiset, tunnistustarkkuus putoaa merkittävästi [36]. Tällainen tilanne voi olla esimerkiksi, jos puhujantunnistusjärjestelmä on opetettu puhtaalla puhedatalla mutta tunnistettavassa signaalissa on syystä tai toisesta paljon kohinaa. Wenndt ja Shamsunder [36] kokeilivat ns. *bispektriä* tämän ongelman ratkaisemiseksi ja saivat parempia tunnistustuloksia kohinaisissa olosuhteissa kuin kepstrillä. Kepstrin on havaittu antavan erinomaisia tuloksia laboratorio-olosuhteissa, mutta tulokset ovat heikentyneet jos puhe on kovin huonolaatuista [36].

### 4.6.1 Prosodiset piirteet – uusi lähestymistapa?

Tutkijat ovat kiinnittäneet harvinaisen vähän huomiota prosodisten (suprasegmentaalisten) piirteiden käyttöön puhujan tunnistamisessa. Prosodiset piirteet liittyvät ihmisen puhe*tapaan*, joka lienee varsin yksilöllinen; ihmiset puhuvat eri nopeuksilla ja painotuksilla. Prosodisten piirteiden käytön niukkuuteen lienee parikin syytä. Ensinnäkin prosodisten piirteiden selvittäminen on huomattavasti hankalampi tehtävä kuin spektraalisten piirteiden ja se vaatii luotettavaa puheen segmentointialgoritmia. Toinen syy on, että toisen ihmisen puhetapaa pystyy verrattain helposti imitoimaan. Uudessa lähestymistavassa voitaisiin kuitenkin yhdistää sekä spektraaliset että prosodiset piirteet ja tehdä näiden molempien perusteella päätös henkilöllisyydestä. Tämä saattaisi parantaa tunnistettavuutta esimerkiksi silloin, kun siirtokanava (esim. puhelinverkko) vääristää taajuusvastetta, mutta säilyttää kuitenkin puheen rytmi-informaation. Mathew & al. [27] havaitsivat, että spektraalisten ja suprasegmentaalisten piirteiden yhdistäminen paransi tunnistustarkkuutta. Tämän esityksen viimeisessä luvussa hahmottelemme uuden yksinkertaisen menetelmän rytmi-informaation selvittämiseksi puhesignaalista.

# 5 VEKTORIKVANTISOINTIIN PERUSTUVA LUOKITTELU

Luokittelu on olennainen osa hahmontunnistusprosessia sekä opetus- että tunnistusvaiheissa. Opetusvaiheessa piirrevektoreiden avulla konstruoidaan kullekin *luokalle* matemaattinen malli, joka kuvaa tämän luokan yksilöllisiä piirteitä. Puhujan tunnistusongelmassa luokat ovat puhujia. Tunnistusvaiheessa luokittelija luokittelee tunnistettavat piirrevektorit tiettyihin luokkiin käyttäen jotain sopivaa luokittelukriteeriä. Tässä luvussa käsittelemme luokittelua puhujan tunnistuksessa ja perehdymme tarkemmin vektorikvantisointiin (VQ), joka on puheen- ja puhujan tunnistuksessa sekä puheenkoodauksessa hyväksi havaittu, tehokas ja helposti toteutettavissa oleva algoritmi.

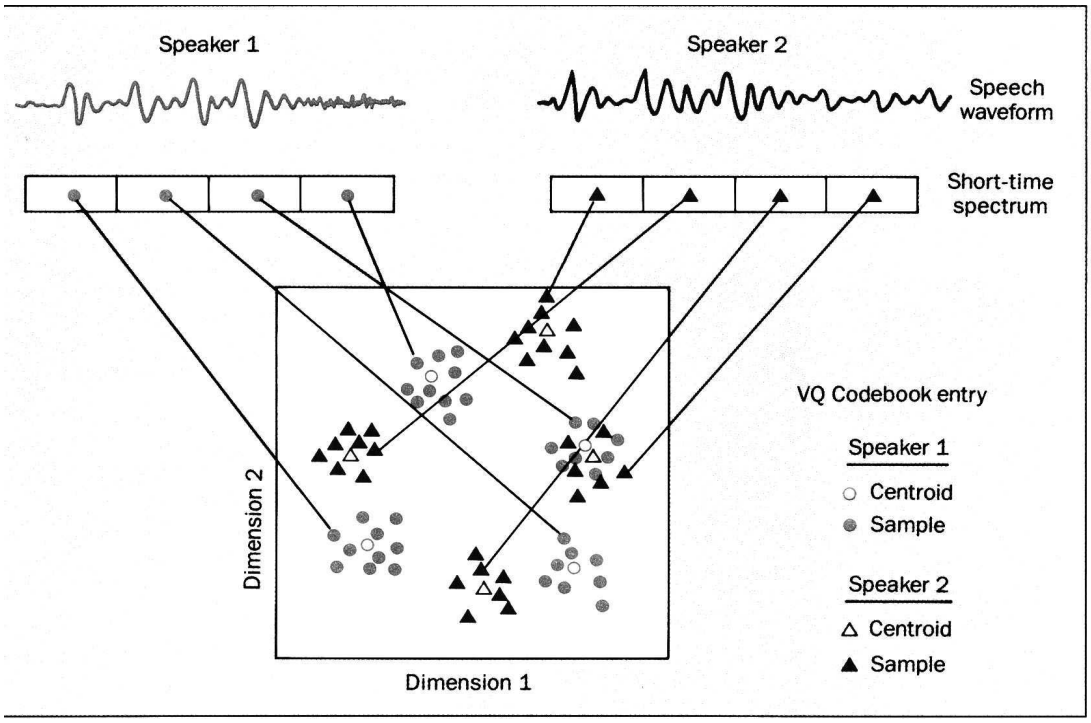
Luvun sisältö on jaoteltu seuraavasti. Kohdassa 5.1 täsmennämme ja havainnollistamme VQ:n toimintaa. Kohdassa 5.2 perehdymme opetusvaiheeseen eli siihen, kuinka puhujalle konstruoidaan VQ-malli. Kohdassa 5.3 taas katsotaan, kuinka varsinainen tunnistaminen tapahtuu. Lopuksi pohdimme VQ:hun kohdistettua kritiikkiä ja muita vaihtoehtoisia luokittelijoita.

## *5.1 Mitä on vektorikvantisointi?*

Erilaisia luokittelualgoritmeja ja –arkkitehtuureita on olemassa koko joukko aina tilastollisesta analyysista neurolaskentaan. Monet luokittelijat, erityisesti neurolaskentaan perustuvat, ovat usein rakenteeltaan monimutkaisia ja niiden täydellinen käsittely vaatii paljon matemaattista teoriaa, johon me emme voi tässä esityksessä paneutua. *Vektorikvantisointi* (VQ) tarjoaa yksinkertaisen mutta tehokkaaksi havaitun ja helposti toteutettavissa olevan mahdollisuuden tunnistaa puhujia. Sen sijaan, että VQ käyttäisi monimutkaista tilastollista analyysia, se toimii enemmänkin “ad hoc” –periaatteella, jonka idea ja matemaattinen formulointi on helppo ymmärtää intuitiivisesti.

Kutakin luokkaa eli puhujaa kuvaa suuri joukko piirrevektoreita. Pohjimmiltaan VQ on tiedonpakkausalgoritmi, joka tarjoaa näppärän tavan tiivistää alkuperäinen vektorijoukko huomattavasti pienemmäksi joukoksi uusia vektoreita, ns. *koodivektoreita* [18]. Koodivektorit valitaan siten, että ne ovat jonkin kriteerin mukaan optimaaliset; voidaan esimerkiksi minimoida neliösumma ryhmään kuuluvien vektorien etäisyyksistä ryhmän koodivektoriin. Koodivektoreista muodostettua joukkoa kutsutaan yleensä *koodikirjaksi*.

Oletetaan, että piirrevektorit ovat $p$-ulotteisen vektoriavaruuden $\mathbf{R}^p$ alkioita. Olkoon opetusvektoreiden joukko $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L\}$ ja merkitään koodikirjaa $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$, missä $K << L$. Formaalisti ajatellen vektorikvantisoija on kuvaus $q$, joka liittää jokaiseen syötevektoriin $\boldsymbol{x}_k$ yksikäsitteisen vektorin $q(\boldsymbol{x}_k)$ joukosta $C$ [26]. Syötevektori voi olla opetusjoukosta tai sen ulkopuolelta, Kvantisoijan täydellinen kuvaus sisältää koodikirjan $C$ lisäksi syötevektorien osittelun alueisiin $S_i$ siten, että $S_i = \{\boldsymbol{x}_k \mid q(\boldsymbol{x}_k) = c_i\}$.




**Kuva 5.1.** [33]. Vektorikvantisoinnin havainnollistus 2-ulotteisessa tapauksessa. Puhesignaalin kehyksistä johdetut piirrevektorit muodostavat piirreavaruuteen ryhmiä, joita edustaviksi vektoreiksi on valittu ryhmien keskiarvot.

Opetusvaiheessa jokaiselle puhujalle muodostetaan oma koodikirja, joka kuvaa tämän puhujan piirteitä. Kukin koodivektori edustaa nyt jotain puheen akustista yksikköä. Toisin sanoen tietyn puhujan koodikirja kuvaa sitä, miten juuri hänen elimistönsä ilmaisee jonkin tietyn yksikön. Opetusvaihe on sekä identifiointi- että verifiointitehtäville sama: konstruoidaan kunkin puhujan koodikirja. Tehtävät poikkeavatkin toisistaan vain tunnistusvaiheen osalta.

### *5.2 Opetusvaihe*

Tarkastellaan seuraavaksi, miten koodikirja muodostetaan tietylle puhujalle. Koodikirjan kunkin koodivektorin tulisi kuvata tietyn akustisen yksikön “lokeroa” piirreavaruudessa. Ongelmana on kuitenkin, että luokiteltaessa tiettyä vektoria emme tiedä, mihin ryhmään sen tulisi kuulua luokan sisällä ja millainen luokan sisäinen rakenne ylipäätään on. Tämä johtaa väistämättömästi menetelmään, jossa tietyn luokan opetusdata jaetaan *automaattisesti* ”lokeroihin” piirreavaruudessa. Tällaista automaattista luokittelua sanotaan *ryhmittelyksi* tai *ohjaamattomaksi oppimiseksi* [8].

Erilaisia ryhmittelyalgoritmeja on olemassa koko joukko. Tässä perehdymme laajalti käytettyyn *K-means* (*C-means, ISODATA*) –algoritmiin. *K*-means –menetelmässä ryhmän koodivektorin esitys on ryhmän vektorien keskiarvovektori. *K*-means lähtee liikkeelle sopivasta alkuarvauksesta ryhmittelylle ja korjaa sitten iteratiivisesti keskiarvoja siten, että ryhmittelyn hyvyys paranee tai pysyy vähintään samana joka kierroksella. Iterointi lopetetaan, kun mikään keskiarvo ei muutu. Seuraavassa on esitetty tyypillinen muoto *K*-means –algoritmista [8].

ALGORITMI 5.2.1 (*K*-means)

(1) Olkoon opetusvektoreiden joukko $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L\}$. Osittele $S$:n vektorit $K$:hon ($K << L$) ryhmään $\{S_1, \ldots, S_K\}$ ja laske ryhmien keskiarvot $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$;

(2) Olkoon $d$ jokin *etäisyysmitta* kahden piirreavaruuden vektorin välillä;
Liitä kukin opetusvektori $\boldsymbol{x}_j \in S$ siihen ryhmään $S_k$, jolle $d(\boldsymbol{x}_j, \boldsymbol{c}_k)$ on pienin;

(3) Laske (2):n tuloksena syntyneen koodikirjan uudet keskiarvot $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$;

(4) Jos jokin keskiarvo muuttui, siirry kohtaan (2). Muutoin lopeta;

*K*-means-algoritmissa on kaksi parametria, jotka pitää kiinnittää etukäteen: haluttu ryhmien määrä (*K*) sekä etäisyysmitta (*d*) piirreavaruudessa $\mathbf{R}^p$. Sopiva ryhmien määrä löydetään monesti kokeilemalla ja/tai tarkastelemalla sovellusalueen teoriaa. Etäisyysmittana useimmiten euklidinen etäisyys,

$$d_e(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{k=1}^{p} (\boldsymbol{x}_k - \boldsymbol{y}_k)^2}\,, \tag{5.2.1}$$

on luonteva ja hyvä etäisyysmitta. Koska neliöjuuri on monotonisesti kasvava funktio, on sama käytetäänkö minimietäisyyden laskemiseen euklidista etäisyyttä vai sen neliötä. Muitakin etäisyysfunktiota on kuitenkin käytetty. Esimerkiksi Finan & al. [14] käyttävät kahden kepstrivektorin välisen etäisyyden laskemiseen *korttelietäisyyttä*:

$$d_c(\boldsymbol{x},\boldsymbol{y}) = \sum_{k=1}^{p} | \boldsymbol{x}_k - \boldsymbol{y}_k | . \tag{5.2.2}$$

Puheanalyysissa käytetään monesti *K*-means-algoritmin muunnelmaa, jota kutsutaan kehittäjiensä mukaan *Linde-Buzo-Gray* (*LBG*) –algoritmiksi [26]. Algoritmista käytetään myös nimitystä *yleistetty Lloydin algoritmi* (*GLA*). Algoritmi on esitetty seuraavassa.

ALGORITMI 5.2.2 (*LBG* tai *GLA*)

(1) Olkoon opetusvektoreiden joukko $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L\}$; Valitse sopiva koodivektoreiden joukko
$\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$;

(2) Olkoon $d$ jokin *virhemitta* kahden piirrevektorin vektorin välillä;
Liitä kukin opetusvektori $\boldsymbol{x}_j \in S$ siihen ryhmään $S_i$, jolle $d(\boldsymbol{x}_j, \boldsymbol{c}_i)$ on pienin;

(3) Laske (2):ssa syntyneen ryhmittelyn tuloksena syntynyt keskimääräinen kvantisointivirhe $D$ tässä iteraatiossa ($t$):

$$D^t = \frac{1}{L} \sum_{k=1}^{L} \min_{j=1}^{K} d(\boldsymbol{x}_k, \boldsymbol{c}_j) \tag{5.2.3}$$

(4) Jos $\frac{D^{t-1} - D^t}{D^t} < \varepsilon$, lopeta. Muutoin jatka;

(5) Laske (2):n tuloksena syntyneen koodikirjan uudet koodivektorit siten, että koodikirja on optimaalinen; Palaa kohtaan (2);

LBG-algoritmi muistuttaa pitkälti *K*-means –tyypin algoritmia ja usein näitä kahta käytetäänkin synonyymeina. LBG poikkeaa *K*-meansista lähinnä vain käsitteellisessä mielessä. *K*-meansissa käytettävä etäisyysmitta on useimmiten euklidinen etäisyys tai muu metrinen etäisyysfunktio, kun taas LBG:ssä käytetty virhemitta voi olla myös ei-metrinen mitta [8]. Esimerkiksi Soong & al. [33]

käyttivät piirteinä LP-kertoimia, joiden similariteettimittana käytettiin ns. *LPC-uskottavuussuhdetta*. Tämä mitta määritellään vektoreiden $\boldsymbol{x}$ ja $\boldsymbol{y}$ välille seuraavasti:

$$d_{LR}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{y}^T \boldsymbol{R}_x \boldsymbol{y}}{\boldsymbol{x}^T \boldsymbol{R}_x \boldsymbol{x}} - 1, \tag{5.2.4}$$

missä $\boldsymbol{R}_x$ on vektoriin $\boldsymbol{x}$ liittyvä syötedatan *autokorrelaatiomatriisi* [8]. Etäisyysmitan lisäksi toinen ero LBG:n ja *K*-meansin perusversion välillä on, että LBG sallii myös kvantisointivirheen mielessä epäoptimaalisen tuloksen; iterointi päättyy, kun virheen muutos edellisestä iteraatiosta alittaa halutun kynnysarvon.

### 5.2.1 Optimaalisuus

Yleisesti tarkasteltuna kvantisoijan optimaalisuutta koodikirjan iteroinnin aikana tarkastellaan kahdella tavalla [21]: *osittelun* ja *koodikirjan* optimaalisuudella. LBG:n askel (2) optimoi annetun osittelun nykyisen koodikirjan suhteen. Viimeinen askel taas optimoi koodikirjan, kun annettuna on nykyisen iteraation osittelu.

LBG:n viimeisessä askeleessa ei ole kiinnitetty, *minkä* kriteerin suhteen nykyisen iteraation koodikirjan tulee olla optimaalinen. Monesti optimaalisuuskriteeriä ei tarvitse eksplisiittisesti edes käyttääkään koodikirjaa muodostettaessa. Usein halutaan minimoida virheiden neliösumma opetusvektorin $\boldsymbol{x}_i$ ja sitä vastaavan koodivektorin $\boldsymbol{c}_j = q(\boldsymbol{x}_i)$ välillä:

$$d(\boldsymbol{x}_i, q(\boldsymbol{x}_i)) = d(\boldsymbol{x}_i, \boldsymbol{c}_j) = \sum_{k=1}^{p} (\boldsymbol{x}_{ik} - \boldsymbol{c}_{jk})^2. \tag{5.2.5}$$

Neliösumman mielessä optimaalinen koodikirja saadaan laskemalla ryhmän vektorien *keskiarvovektori*:

$$\boldsymbol{c}_j = \frac{\sum_{\boldsymbol{x} \in S_j} \boldsymbol{x}}{|S_j|}, \tag{5.2.6}$$

missä $S_j$ on koodivektoria $\boldsymbol{c}_j$ vastaava ryhmä ja $|S_j|$ sen alkioiden lukumäärä. Näin ollen *K*-means pyrkii neliösumman mielessä optimaalisen koodikirjan muodostamiseen.

### 5.2.2 Ryhmittelyn alkuarvauksen valinta

LBG-algoritmissa ryhmittelylle tarvitaan jokin alkuarvaus. Keskiarvoiksi voidaan valita periaatteessa mielivaltaiset vektorit, esimerkiksi *K* ensimmäistä vektoria opetusjoukosta tai satunnaisesti valitut *K* vektoria. Puhetta käsiteltäessä peräkkäisistä kehyksistä johdetut piirrevektorit eivät kuitenkaan ole kovin järkevä valinta, koska ne ovat vahvasti korreloituneet ja ovat lähellä toisiaan piirreavaruudessa [26].

Koko LBG:n idea on valita aluksi jokin koodivektoreiden joukko ja sitten iteroida tätä koodikirjaa ryhmien sisäisillä eli *lokaaleilla* muutoksilla siten, että kokonaiskvantisointivirhe pienenee tai pysyy vähintään samana jokaisella iteraatiolla. Koska muutokset ovat lokaaleja, alkuarvauksen valinta vaikuttaa oleellisesti kvantisoijan hyvyyteen [21].

Linde & al. [26] ehdottavat koodivektorien alkuarvojen määrittämiseen seuraavaa ylhäältä-alas –periaatteella toimivaa algoritmia, joka tuottaa $K = 2^m$ ($m = 0, 1, 2, \ldots$) ryhmää.

Algoritmi 5.2.3 (Alkuarvaus vektoreille hajoitustekniikalla)

(1) Laske kaikkien opetusvektoreiden $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ keskiarvovektori;
Olkoon tämä koodikirjan ainut vektori;

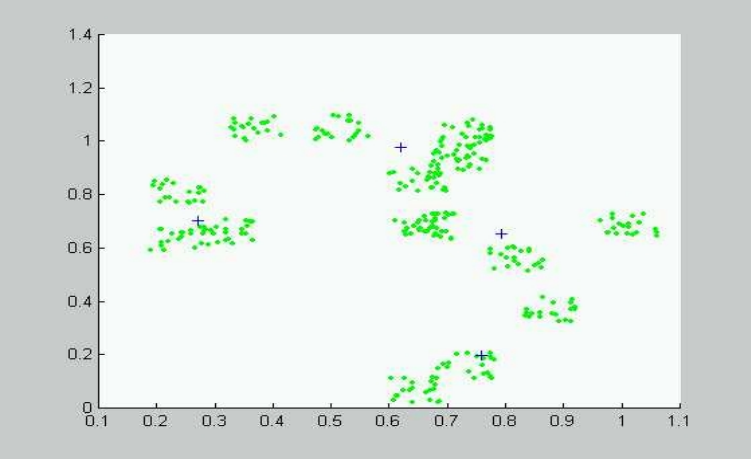
(2) Korvaa jokainen koodivektori $\boldsymbol{c}$ kahdella uudella
vektorilla $\boldsymbol{c} + \boldsymbol{\varepsilon}$ ja $\boldsymbol{c} - \boldsymbol{\varepsilon}$, missä $\boldsymbol{\varepsilon}$ on sopiva (pituudeltaan pieni) vektori;

(3) Jos vektoreiden määrä on haluttu, lopeta; Muutoin jatka kohdasta 2;

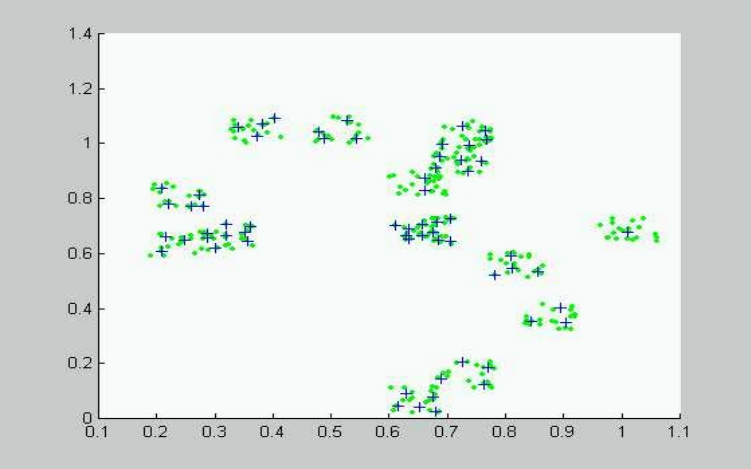
Koodikirjan muodostaa siis alussa kaikkien opetusvektoreiden keskiarvo. Tämän jälkeen jokaisella iteraatiolla kukin vektori hajotetaan kahdeksi (entisen koodivektorin suhteen) symmetriseksi vektoriksi, kunnes vektorien määrä on haluttu. Vektoriksi $\boldsymbol{\varepsilon}$ voidaan valita esimerkiksi

piirrevektoreiden varianssi. Tavallisia koodikirjan kokoja puhesovelluksissa ovat esim. *K*=32 ja *K*=64. Usein hajotustekniikka ei kuitenkaan käytetä, sillä satunnaisesti valitut *K* vektoria ovat osoittautuneet paremmaksi heuristiikaksi. Hajotusmenetelmän tuottamat aloitusvektorit eivät ole opetusjoukosta, mikä voi johtaa jopa siihen, että joihinkin ryhmiin ei liity iteroinnin jälkeen yhtään vektoria!

Kuvissa 5.2 (a) ja (b) on havainnollistettu sitä, miten ryhmien määrän valinta vaikuttaa koodikirjan keskineliövirheeseen (*MSE*). Opetusvektoreita on merkitty palloilla, koodivektoreita +:lla. Kohdassa (a) koodivektoreita on vain 4, kun taas ryhmien todellinen määrä on selvästi suurempi. Kuvassa (b) koodivektoreita taas on silmämääräisesti vähintään saman verran kuin ryhmiäkin. Havaitaan, että keskineliövirhe pienenee, kun ryhmien määrää kasvatetaan. Tietenkin on muistettava, että jos koodivektorien määrä on hyvin suuri, VQ:n hyöty tiedonpakkausalgoritmina pienenee.




**Kuva 5.2. (a)** Liian vähän koodivektoreita (*MSE* = 0.1354).




**Kuva 5.2. (b)** Enemmän koodivektoreita, parempi sovitus (*MSE* = 0.0161).

Soong & al. [33] kokeilivat erilaisia koodikirjan kokoja puhujan tunnistuksessa ja tulokset näyttivät, että koodikirjan koko 64 oli sopiva koko: tätä suuremmat koodikirjat eivät parantaneet tunnistustarkkuutta merkittävästi. On kuitenkin syytä muistaa, ettei tämä ole mikään universaali tulos, vaan esimerkiksi eri kielissä on eri määrät foneemeja. Myös piirteiden valinnalla on merkitystä.

### 5.2.3 Koodikirjojen päivittäminen

Tutkimukset osoittavat, että puhujan äänessä tapahtuu ajan mittaan muutoksia [33]. Puhuja voi lisäksi lukea lauseet eri tavalla opetusvaiheessa kuin tunnistusvaiheessa, mutta tämä ei tietenkään saa vaikuttaa tunnistamiseen. Soong & al. [33] tutkivat monien eri tekijöiden vaikutusta puhujan tunnistukseen. He käyttivät puhujien mallintamiseen LBG-algoritmilla muodostettuja VQ-koodikirjoja. He tekivät mm. seuraavan arvattavissa olevan havainnon: mitä pidempi aika opetus- ja tunnistusvaiheiden välillä oli, sitä huonompia tunnistusprosentteja saatiin. Sen lisäksi että puhujan äänessä tapahtuu variaatiota ajan kuluessa, myös opetus- ja tunnistusvaiheiden äänitysolosuhteet voivat olla erilaiset. Esimerkiksi puhelimen yli käytettävissä sovelluksissa siirtokanavan ominaisuudet vaikuttavat luonnollisesti taajuusvasteeseen. Lisäksi joko opetus- tai tunnistusvaiheessa voi olla taustahälyä tai muita vastaavia häiriötekijöitä, joka puuttuu toisesta vaiheesta.

Edelliset seikat osoittavat, että koodikirjoja on päivitettävä ja adaptoitava olosuhteiden mukaisiksi. Yksinkertaisin päivitysalgoritmi voisi toimia vaikka seuraavaan tapaan. Muistissa pidetään kaikkia piirrevektoreita ja niitä käsitellään *FIFO*-periaatteella: kun uusia vektoreita tulee sisään, vanhimmat poistetaan ja uusi koodikirja lasketaan muistissa olevista vektoreista. Tämä lähestymistapa ei kuitenkaan ole muistinkäytön kannalta kovin edullinen. Brunelli ja Falavigna [4] käyttävät menetelmää, jossa alkuperäisiä koodivektoreita käännetään adaptointivektoreiden suuntaan tiettyjen parametrien mukaan. Tässä esityksessä ei ole tarpeen mennä algoritmin yksityiskohtiin.

### *5.3 Tunnistusvaihe*

Oletetaan nyt, että kunkin puhujan koodikirjat on muodostettu jollakin ryhmittelyalgoritmilla. Kukin koodikirja koostuu vektoreista $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$. Tehtävänä on luokitella puhesignaalista johdettu piirrevektorijono $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$, so. päättää kuka puhuja on tuottanut kyseiset vektorit.

Luokittelulle tarvitaan aina jokin *luokittelukriteeri*. Nyt luonteva valinta on keskimääräinen kvantisointivirhe tunnistettavan vektorijoukon ja tietyn koodikirjan välillä, joka lasketaan käyttäen kaavaa 5.2.3. Virhe lasketaan siis tietylle koodikirjalle seuraavasti. Ensin vektori sijoitetaan koodikirjan siihen ryhmään, jolle sen etäisyys (kvantisointivirhe) ryhmän koodivektorista on pienin. Tämä toistetaan kaikille tuntemattomille vektoreille, yksittäiset kvantisointivirheet summataan ja lopuksi summa jaetaan vektorien määrällä. Näin saatu mitta kuvaa sitä, kuinka hyvin luokiteltava vektorijoukko keskimäärin muistuttaa tiettyä luokkaa eli puhujaa.

Edellistä keskivirhettä, tai yleisemmin jotain toistakin *similariteettimittaa*, käytetään nyt luokittelukriteerinä seuraavasti [11, 16, 33]. *Identifiointitehtävässä* lasketaan tunnistettavien piirrevektorien kvantisointivirhe kaikkien koodikirjojen suhteen ja tunnistettava puhuja on se, jonka koodikirjalle virhe on pienin. Jos kyseessä on avoin joukko, eli tunnistettava puhuja voi puuttua tietokannasta, etsitään samalla tavalla pienin virhe ja jos tämä virhe on etukäteen asetettua kynnysarvoa pienempi, puhuja on identifioitu. Jos virhe ei alita kynnysarvoa, annetaan tulos “ei päätöstä”.

*Verifioinnissa* tunnistaminen tapahtuu samaan tapaan. Nyt riittää laskea piirrevektoreiden keskivirhe ainoastaan väitetylle puhujalle (koodikirjalle). Jos virhe on ennalta asetettua kynnysarvoa pienempi, annetaan positiivinen tunnistustulos, muutoin negatiivinen.

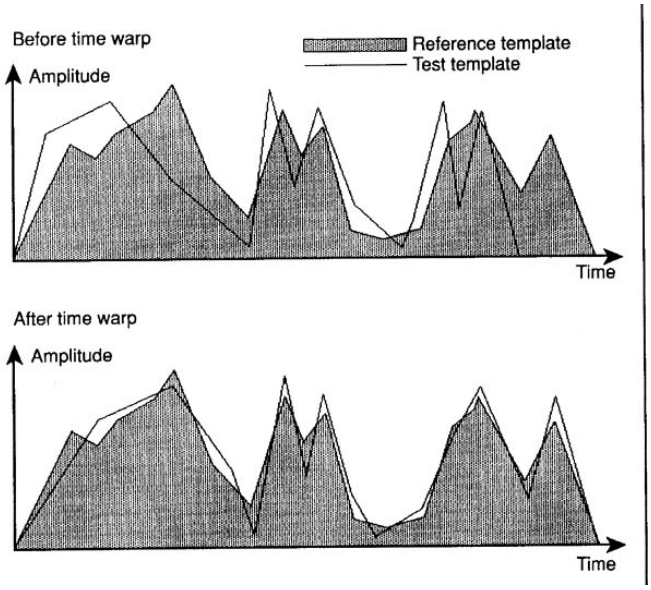
Eräs tärkeä kysymys verifioinnissa on, kuinka hyväksymisen/hylkäämisen kynnysarvo tulisi valita [38]. Jos kysymys on esimerkiksi kriittisestä turvallisuussovelluksesta, on erittäin tärkeää kiinnittää huomiota tunnistusvirheiden minimoimiseen. Puhujan verifiointijärjestelmä voi tehdä kahdenlaisia virheitä: *väärien hyväksymistä* ja *oikeiden hylkäämistä*. Ensin mainittu tarkoittaa, että järjestelmä antaa positiivisen tunnistuspäätöksen, vaikka puhuja onkin väärä. Jälkimmäisessä taas annetaan negatiivinen tunnistuspäätös, vaikka tunnistettava on väitetty puhuja. On selvää, että erityisesti

kriittisissä turvasovelluksissa väärien hyväksyminen on paljon vakavampi virhe kuin oikeiden hylkääminen. Monesti kynnysarvo asetetaan siten, että tällä arvolla edellä mainitut virheet ovat yhtä suuret (ns. *yhtäsuuri virhemäärä* eli *EER*) [esim. 15, 16]. Vertailtaessa eri verifiointijärjestelmiä ja niiden hyvyyttä, EER-virheiden tarkastelu on hyvä tätä tarkoitusta varten. Kynnysarvon laskeminen ”jälkikäteen” eli mittaamalla suoraviivaisesti näitä virheitä ei sovellu kuitenkaan kovin hyvin käytännön tilanteisiin. Onkin esitetty menetelmiä, joissa kynnysarvo lasketaan *a priori* opetusvaiheen aikana [15, 38].

### *5.4 Kritiikkiä ja muita menetelmiä*

LBG on helposti toteutettavissa oleva ja tehokas datan ryhmittelyalgoritmi, mistä syystä sitä käytetäänkin paljon. Kuitenkin algoritmia kohtaan on esitetty runsaasti kritiikkiä. Algoritmin pahimpana puutteena pidetään yleisesti sitä, että ryhmittelyn hyvyys riippuu vahvasti alkuarvauksesta, koska koodikirjan iterointi perustuu vain lokaaleihin muutoksiin [21]. Vaikka koodikirja olisi opetusvektoreille optimaalinen, se ei takaa, että opetusjoukon ulkopuolisten vektoreiden luokittelu olisi optimaalinen [18]. Näistä syistä on esitetty koko joukko algoritmeja, joilla LBG:llä muodostettuja koodikirjoja optimoidaan globaalisti. Näitä menetelmiä ovat esimerkiksi *oppiva vektorikvantisointi* (LVQ) [23] ja *ryhmävektorikvantisointi* (GVQ) [18]. VQ ei ole kuitenkaan ainoa eikä välttämättä paras lähestymistapa luokitteluun. Seuraavassa käydään lyhyesti läpi muita yleisesti käytettyjä luokittimia.

***Dynaaminen aikasovitus.*** *Dynaamista aikasovitusta* [8, 15] eli *DTW*:tä on käytetty puheentunnistuksessa, joten se soveltuu myös tekstistä riippuviin menetelmiin puhujan tunnistuksessa. Menetelmässä tunnistettavaa hahmoa (kehystä) kuvaa signaalin aaltomuoto. Järjestelmällä on muistissaan eri luokkien aaltomuodot, joihin se vertaa tunnistettavaa signaalia. Sen sijaan että laskettaisiin suoraan näytearvojen virheiden summa, suoritetaan ensin dynaamisen aikasovitus eli ”venytetään” tai ”kutistetaan” signaalia ajan suhteen siten, että virheiden summa minimoituu. Kehys tunnistetaan siksi luokaksi, jolle virheiden summa on pienin. Lopullinen päätös puhujasta tehdään samoilla periaatteilla kuin vektorikvantisoinnissakin. DTW:n ideaa on havainnollistettu seuraavassa kuvassa.




**Kuva 5.3.** [31]. DTW:n periaate.

***Kätketyt Markovin mallit***. *Kätketyt Markovin mallit* (HMM) [8] lienee tällä hetkellä eniten käytetty menetelmä puheentunnistuksessa, joten niitä käytetään tekstistä riippuvissa puhujantunnistusmenetelmissä [16]. HMM on äärellinen tilasiirtymäautomaatti, jolle annetaan syötteenä joukko piirrevektoreita Tuloksenaan yksittäinen HMM antaa todennäköisyyden, että tämä HMM on "generoinut" tutkittavat vektorit. Jokaista akustista yksikköä kohden, oli se sitten fooni, foneemi, sana tai lause, muodostetaan opetusvaiheessa oma Markovin malli. Tunnistusvaiheessa piirrevektorit syötetään kaikille HMM:ille ja valitaan se HMM, joka on kaikkein todennäköisimmin tuottanut nämä vektorit. Näin lauseen sisältö saadaan selville. Jokaiselle puhujalle muodostetaan opetusvaiheessa oma HMM-joukkonsa. Yleensä HMM:ää käytetään "esikäsittelijänä" puheen segmentointiin ja varsinaisen päätöksen puhujan identiteetistä tekee jokin toinen luokittelija [30].

***Tilastolliset menetelmät***. Tilastollisissa menetelmissä luokat esitetään niiden tilastollisten ominaisuuksien avulla, esimerkiksi todennäköisyysjakauman tiheysfunktioina (*Bayesilainen päätösanalyysi*). Puhujan tunnistuksessa on käytetty mm. spektraalisten piirteiden keskiarvoja ja variansseja, jotka on laskettu pitkältä aikaväliltä [16]. Suosittuja malleja ovat myös ns. *Gaussilaiset mikstuurit* (GMM) [esim. 6].

***Neurolaskenta***. Neurolaskennan idea on esittää eri luokat keinotekoisen neuroverkon *rakenteen* avulla, joka määritetään opetusvaiheessa. Erilaisia verkkoarkkitehtuureita on olemassa useita. Tunnistusvaiheessa tutkittavat vektorit syötetään verkolle, joka tekee päätöksen. Neurolaskennan

soveltuvuutta puhujan tunnistukseen ovat testanneet mm. [13, 24, 27, 29, 30, 38]. Farrell & al. [11] vertailevat monipuolisesti neurolaskentaa sekä useita muita luokittelijoita.

***Eri luokittelijoiden yhdistelmät.*** Yleisessä hahmontunnistuksessa on kokeiltu eri luokittelijoiden yhdistelmiä. Myös puhujan tunnistuksessa on testattu tällaisten arkkitehtuureiden mahdollisuuksia. Esimerkiksi Olsen [30] kokeili HMM:n ja RBF-neuroverkon yhdistelmää. Mathew & al. [27] yhdistivät kaksi erityyppistä neuroverkkoa. VQ:n yhteydessä on kokeiltu myös mallintaa yhtä luokkaa useammalla kuin yhdellä koodikirjalla [28].

# 6 KOKEELLISET TULOKSET

Tutkielman tekoon sisältyi kokeellinen osa, jossa testattiin itse kerätyllä puhujatietokannalla esitetyn teorian toimivuutta sekä identifiointi- että verifiointitehtävissä. Tässä luvussa kuvaamme puhedatan ominaisuudet, testatut parametrit sekä saadut tulokset.

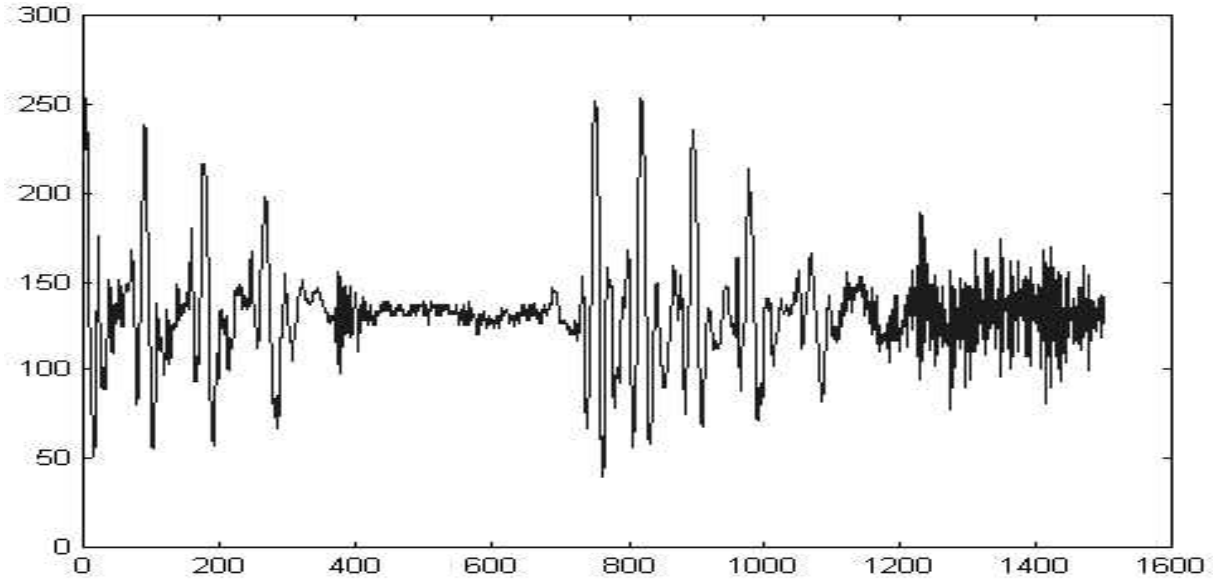
## *6.1 Puhedata ja sen esikäsittely*

Testaamisessa käytetty puhe kerättiin puhujatietokannaksi Suomen TV-kanavilta television kuulokeliitännän kautta monojohdolla ja talletettiin Microsoft wav -tiedostomuotoon. Kaikki puhenäytteet ovat suomenkielisiä ja mahdollisimman neutraalia puhetta, joissa ei ole voimakkaita tunnesävyjä tai painotuksia. Kaikissa näytteissä on kohtalaisen paljon taustakohinaa, joka on aiheutunut mm. verkkohurinasta ja alkuperäisten äänitysolosuhteiden huonoudesta.

Ääni näytteistettiin 11025 Hz:n taajuudella ja 8-bittisellä resoluutiolla. Datasta poistettiin manuaalisesti hiljaiset osuudet, eli katsottiin aaltomuotoesityksestä silmämääräisesti, milloin signaalin energia oli selvästi liian pieni ollakseen muuta kuin hiljaisuutta tai taustakohinaa. Signaalien amplitudit normalisoitiin keskimääräisen *RMS-tehon* (*äänekkyyden*) mukaan. Taulukossa 6.1 on yhteenveto puhujatietokannasta. Tietokanta on vapaasti ladattavissa FTP-osoitteesta ftp://ftp.cs.joensuu.fi/pub/Software/SoundDB/MySpeakerDB.zip. Esimerkki eräästä puhujatietokannan opetussignaalin aaltomuodosta on kuvassa 6.1.

| | |
|---|---|
| Puhujien määrä | 40 (29 miestä + 11 naista) |
| Keskimääräinen signaalin pituus puhujaa kohden | 2.0 s (1.0 s opetukseen + 1.0 s. testaukseen) |
| Näytteistys | 11.025 kHz, 8 bittiä |

**Taulukko 6.1.** Yhteenveto puhujatietokannasta.




**Kuva 6.1.** Tietokannan opetussignaalin ”h12_o.wav” aaltomuotoa. Näytearvot ovat positiivisia kokonaislukuja väliltä 0-255.

### *6.2 Parametrit*

Edellä olemme havainneet, että puhujan tunnistukseen liittyy lukuisia parametreja, jotka osin riippuvat myös toisistaan. Kaikkien mahdollisten kombinaatioiden testaaminen on käytännössä mahdotonta. Seuraavat parametrit oletamme heti alussa kiinnitetyiksi. Piirteinä käytetään mel-kepstriä ja luokittelijana standardia *K*-means –algoritmia, etäisyysmittana euklidinen etäisyys ja alkuarvauksena satunnaiset *K* vektoria opetusjoukosta. Ikkunafunktiona käytetään Hammingin ikkunaa, vierekkäisten kehysten peittoaste on 50 % ja kolmiosuodatinten määrä kepstriä laskettaessa on $\lfloor 3 \cdot \ln F_s \rfloor = 27$. Kepstrikerrointa $c_0$ ei huomioida. Seuraavassa taulukossa on kuvattu parametrit, joita testattiin.

| Parametri | Arvot |
|---|---|
| Ylipäästösuodatus (HPF) | $H(z) = 1 - az^{-1}$, a = 0.95, 0.96, 0.97, 0.98 |
| Kehyksen pituus | $N$ = 64, 128, 256, 512 |
| Kepstrikerrointen määrä | $M$ = 1, 5, 10, 15, 20, 25 |
| Koodikirjan koko | $K$ = 1, 2, 4, 8, 16, 32 |

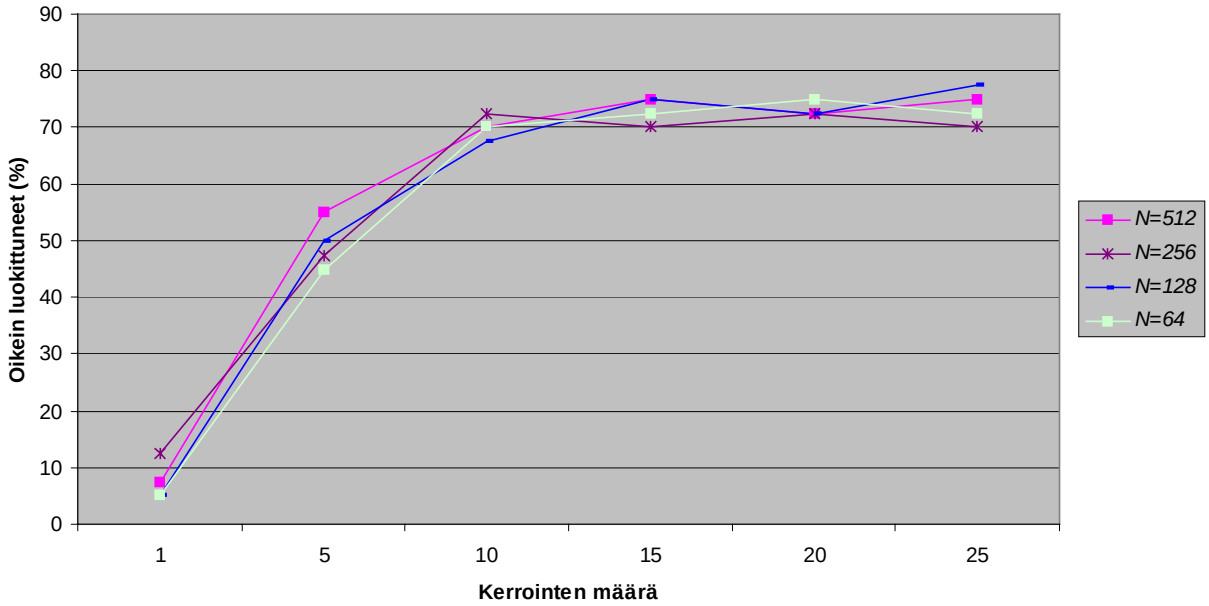
**Taulukko 6.2.** Testatut parametrit.

### *6.3 Tulokset*

Kaikki testit identifiointimoodissa suoritettiin seuraavalla tavalla. Jokaista puhujaa kohden oli keskimäärin 2.0 sekuntia puhetta. Tämä jaettiin 1.0 sekunnin osiin siten, että toista osaa käytettiin luokittelijan opettamiseen ja toista testaamiseen. Jokaisen puhujan testidataa verrattiin kaikkien muiden puhujien koodikirjoihin ja laskettiin, montako prosenttia puhenäytteistä luokittui oikein. Seuraavissa alakohdissa 6.3.1 - 6.3.3 on kuvattu eri parametrien vaikutukset ja alakohdassa 6.3.4 verifiointimoodissa suoritetun testin tulokset.

#### 6.3.1 Kerrointen lukumäärä ja kehyksen pituus

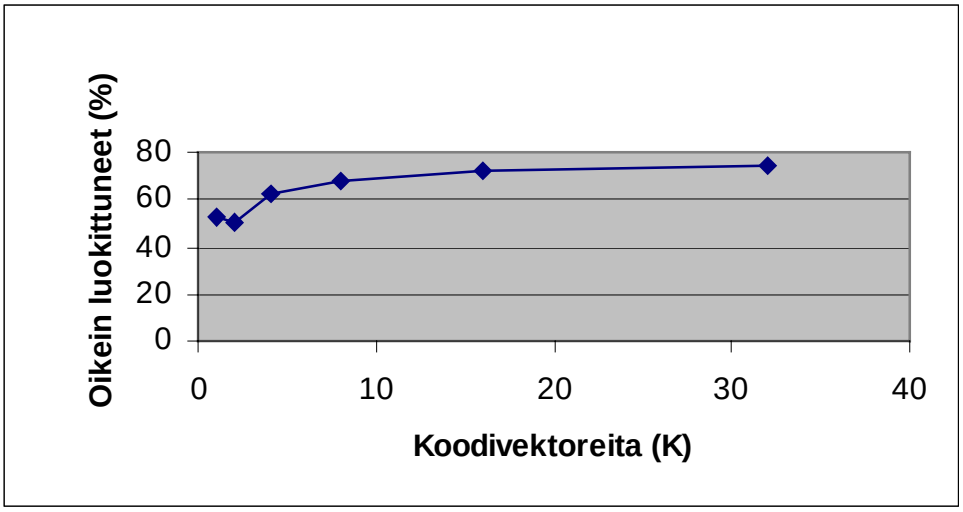
Kuvassa 6.2 on esitetty graafisesti, kuinka kepstrikerrointen lukumäärä (*M*) ja kehyksen pituus (*N*) vaikuttavat tunnistukseen. Tässä testissä koodikirjan koko pidettiin vakiona ($K = 32$) eikä signaalille suoritettu suodatusta. Havaitaan selvästi, että kertoimien lukumäärän kasvattaminen parantaa tunnistusta ainakin tiettyyn rajaan saakka. Sen sijaan kehyksen pituudella ei näyttäisi olevan juurikaan merkitystä. Lisäksi kokeiltiin yhdistää kepstri ja sen 1. ja 2. kertaluvun differenssit (eli 2*M*- ja 3*M*-pituiset piirrevektorit), mutta tämä ei parantanut tunnistusprosentteja.




**Kuva 6.2.** Kepstrikertoimien lukumäärän vaikutus eri pituisilla kehyksillä.

### 6.3.2 Koodikirjan koko

Edellisen testin perusteella kerrointen lukumääräksi valittiin nyt *M*=12 ja kehyksen pituudeksi *N*=256 ja testattiin eri koodikirjojen kokojen vaikutusta. Kuvasta 6.3 havaitaan sama arvattavissa oleva tulos, jonka Soong & al. [33] saivat omissa testeissäänkin: koodikirjan koon kasvattaminen



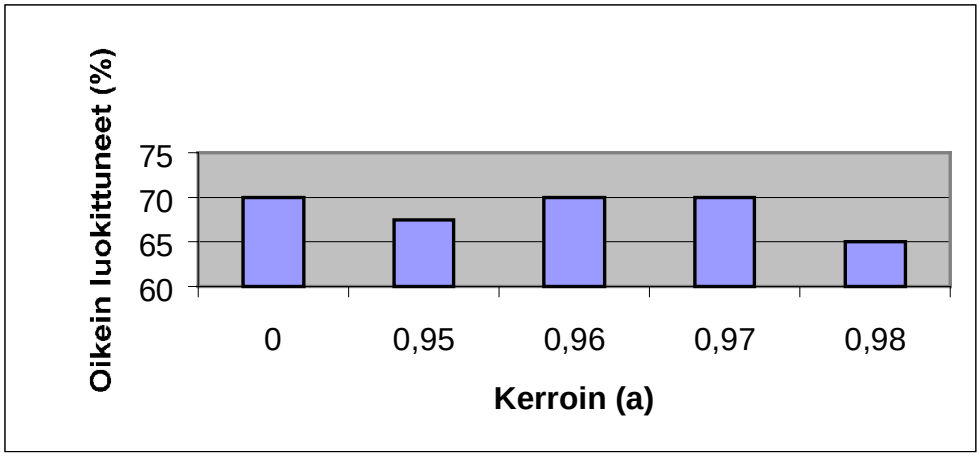

**Kuva 6.3.** Koodikirjan koon vaikutus.

parantaa tunnistusprosenttia. Tässä yhteydessä emme voineet testata suurempia koodikirjoja kuin *K*=64, sillä 1 sekunnin pituisesta signaalista saadaan näillä parametreilla vain vajaat 90 vektoria, eikä siis ole järkeä käyttää suurempia koodikirjoja kuin vektoreiden lukumäärä on. Vähäinen piirrevektorien määrä selittäneekin osittain, miksi tunnistustarkkuus paranee hitaanlaisesti vaikka koodikirjan kokoa kasvatetaan eksponentiaalisesti; VQ:sta saatava todellinen hyöty nähtäisiin vasta suuremmilla vektorijoukoilla.

### 6.3.3 Signaalin esisuodatus

Tässä testissä tutkittiin, mikä vaikutus erilaisilla ylipäästösuodattimilla on tunnistusprosenttiin. Aiemmin totesimme, että puhesignaalin korkeat taajuudet sisältävät paljon puhujakohtaista tietoa, joten ylipäästösuodatuksella pyritään vaimentamaan matalien taajuuksien liiallista osuutta. Suodatus suoritettiin sekä opetus- että testisignaaleille. Pidimme muut parametrit kiinnitettyinä seuraavasti: *K*=32, *M*=12, *N*=256. Tulokset on koottu kuvaan 6.4.

Kuvasta havaitaan, ettei suodatus paranna tunnistusta, vaan jopa huonontaa sitä keskimäärin hiukan. Tähän voi olla osasyynä se, että data oli varsin kohinaista, joten ylipäästösuodatus ainoastaan korostaa korkeataajuista kohinaa. Lisäksi on muistettava, että koska ennen näytteistystä ei suoritettu analogista anti-aliasing-suodatusta, aliasing-ilmiö vääristää amplitudispektriä ja saattaa siten vaikuttaa negatiivisella tavalla tunnistukseen.

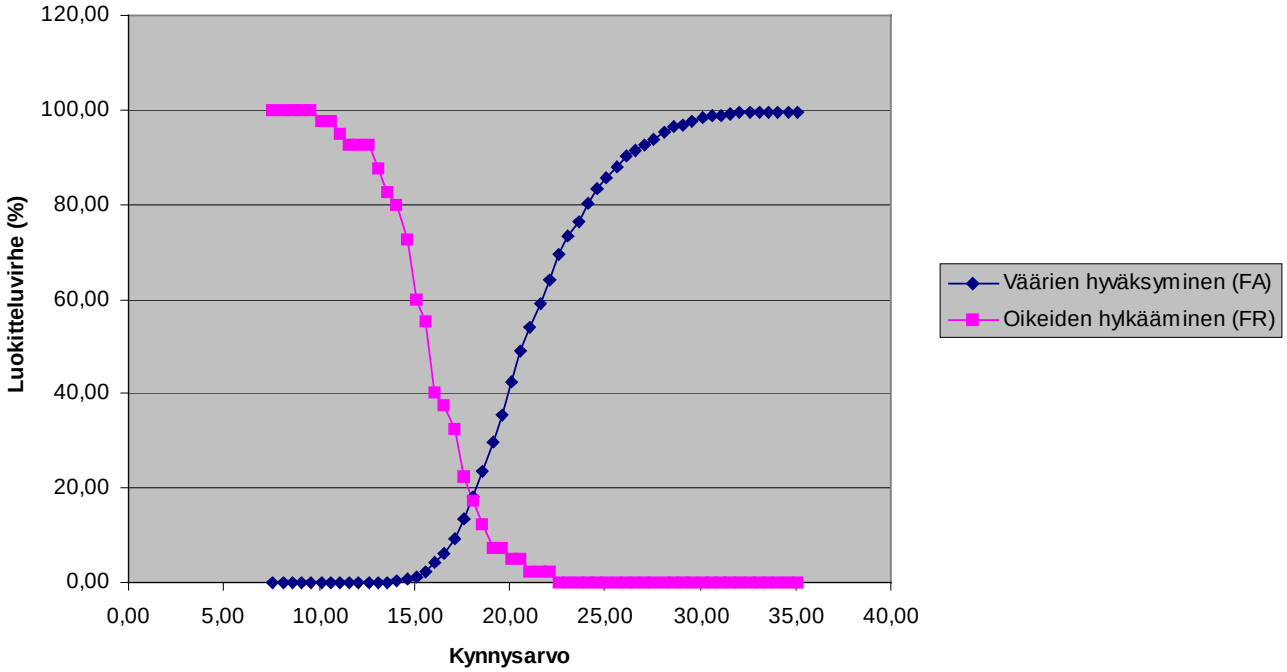



**Kuva 6.4.** Ylipäästösuodattimen $H(z) = 1-az^{-1}$ vaikutus tunnistukseen. Kerroin $a$=0 tarkoittaa, ettei suodatusta ole tehty.

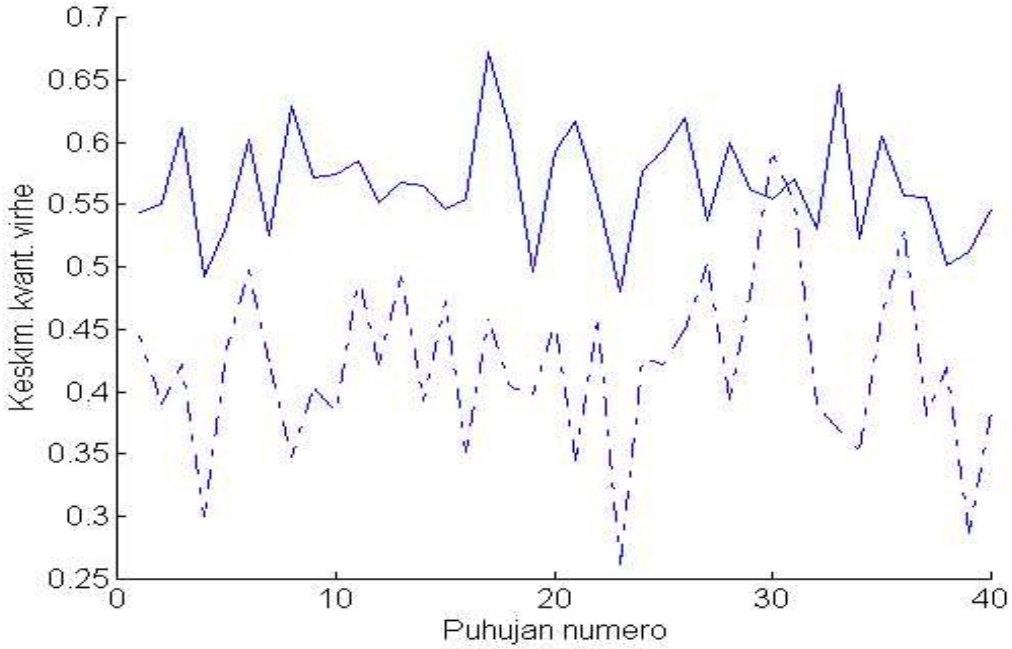
### 6.3.4 Verifiointi

Edellisissä testeissä tutkimme eri parametrien vaikutusta puhujan identifiointiin. Tässä alakohdassa kuvaamme testin, jolla puhujan tunnistamista suoritettiin verifiointimoodissa. Edelleen $K$=32, $M$=12 ja $N$=256. Verifioinnin kynnysarvo määritettiin siten, että oikeiden hylkäämis- (FR) ja väärien hyväksymis (FA) -virheet olivat yhtä suuret. Kuvassa 6.5 on esitetty graafisesti kynnysarvon vaikutus luokitteluvirheisiin. EER-virheeksi saatiin 17.2 %.

Kuvassa 6.6 on vielä esitetty tapauksen $K$=32, $M$=12, $N$=256, testauksessa saadut kvantisointivirheet normalisoituna välille [0, 1]. Ylempi käyrä kuvaa eri puhujien (luokkien) välisiä virheitä ja alempi saman puhujan sisäisiä virheitä. Vastaavat keskiarvot ovat 0.56 ja 0.42. Tulos on samansuuntainen kuin Soongin & al. [33] saama: luokkien välinen kvantisointivirhe on selvästi luokkien sisäistä suurempi, niin kuin kuuluu ollakin.




**Kuva 6.5.** FA- ja FR-virheiden muuttuminen verifioinnin kynnysarvon funktiona (EER = 17.2 %).




**Kuva 6.6.** Luokkien välisten (ylempi käyrä) ja luokkien sisäisten (alempi käyrä) keskimääräiset kvantisointivirheet normalisoituna välille [0, 1].

# 7 POHDINTA

Tässä viimeisessä luvussa vertaamme kokeellisesti saatuja tuloksia muiden saavuttamiin tuloksiin. Lisäksi pohdimme, mistä erot johtuvat ja miten tunnistustarkkuutta voitaisiin edelleen parantaa. Viimeisessä alakohdassa ehdotamme uutta menetelmää tempoinformaation hyödyntämiseksi tunnistuksessa.

## *7.1 Muiden saavuttamia tuloksia*

Tässä kohdassa vertailemme keskenään muutamia tuloksia, joita kirjallisuudessa on saatu. Taulukoihin 7.1 ja 7.2 on koottu identifiointi- ja verifiointituloksia muutamista lähteistä. Tulokset ovat parhaimpia kyseisissä lähteissä saavutetutuista.

Jo näistä taulukoista havaitaan eri menetelmien vertailuun liittyvät ongelmat: käytetään eri piirteitä, erilaatuista puhetta ja eri kokoisia tietokantoja. Lisäksi käytetään usein eri kieliä, eri sukupuolijakaumia, eri pituisia opetus/testilauseita, luetaan erilaisia lauseita (numerot, aakkoset, spontaani puhe…) jne. Campbell ja Reynolds [5] ovat tehneet yhteenvetoa standarditietokannoista, joita puhujan tunnistusjärjestelmien evaluointiin on käytetty ja joita olisi suotavaa käyttää tulevaisuudessakin vertailun helpottamiseksi.

| **Luokittelu** | **Tekstistä riippumattomuus** | **Näytteistys** | **Puhujia** | **Piirteet** | **Oikein luokittuminen** |
|---|---|---|---|---|---|
| VQ [33] | Riippumaton | 6.67 kHz | 100 | LP-kertoimet | 98.5 % |
| VQ [11] | Riippumaton | 8 kHz | 20 | LP-kepstri | 96.0% |
| VQ [29] | Riippuva | 10 kHz | 10 | LP-kepstri | ~ 93 %* |
| VQ [39]; puhdas puhedata | Riippumaton | 8 kHz | 32 | LSP-taajuudet | 100 % |
| VQ [39]; puhelinlaatuinen data | Riippumaton | 8 kHz | 32 | LSP-taajuudet | 69.8 % |
| LVQ [18] | Riippumaton | 8 kHz | 112 | Mel-kepstri | ~ 60 %* |
| GVQ [18] | Riippumaton | 8 kHz | 112 | Mel-kepstri | 86.5 % |
| MLP-neuroverkko [11] | Riippumaton | 8 kHz | 20 | LP-kepstri | 90.0 % |
| MNTN-neuroverkko [11] | Riippumaton | 8 kHz | 20 | LP-kepstri | 96.0 % |
| Bayesin luokittelu [11] | Riippumaton | 8 kHz | 20 | LP-kepstri | 83.0 % |
| Feed-forward-neuroverkko [29] | Riippuva | 10 kHz | 10 | LP-kepstri | ~ 92 %* |
| VQ [TÄMÄ TYÖ] | Riippumaton | 11.025 kHz | 40 | Mel-kepstri | 77.5 % |

**Taulukko 7.1.** Muutamia puhujan identifioinnissa saavutettuja tunnistusprosentteja. Tähdellä merkityt prosentit on arvioitu graafisista esityksistä, koska alkuperäisistä lähteistä puuttuvat tarkat arvot.

| Luokittelu | Tekstistä riippumattomuus | Näytteistys | Puhujia | Piirteet | EER |
|---|---|---|---|---|---|
| DTW [15] | Riippuva | 6.67 kHz | 10 | LP-kepstri | 0.19 % |
| MLP-neuroverkko [13] | Riippumaton | 8 kHz | 20 | LP-kepstri | 0.74 % * |
| RBF-neuroverkko [13] | Riippumaton | 8 kHz | 20 | LP-kepstri | 0.35 % * |
| MNTN-neuroverkko [11] | Riippumaton | 8 kHz | 20 | LP-kepstri | 1.9 % |
| VQ [11] | Riippumaton | 8 kHz | 20 | LP-kepstri | 2.0 % |
| VQ [TÄMÄ TYÖ] | Riippumaton | 11.025 kHz | 40 | Mel-kepstri | 17.5 % |

**Taulukko 7.2.** Verifioinnissa saavutettuja tunnistustuloksia EER-virheellä mitattuna. Tähdellä merkityt virheet on laskettu FA- ja FR-virheiden keskiarvona, koska alkuperäisissä lähteissä ei suoraan annettu EER:ää.

Edelliset taulukot osoittavat myös, ettei mikään luokittelija yllä selvästi muita parempiin tuloksiin. Olipa luokittelija melkein mikä tahansa, muutaman kymmenen puhujan tietokannoilla päästään aina 80-100 % identifiointiin ja parin prosentin suuruusluokkaa olevaan EER-virheeseen verifioinnissa.

## *7.2 Vertailua ja johtopäätöksiä*

Vertailtaessa edellisessä luvussa kokeellisesti saatuja tuloksia taulukoiden 7.1 ja 7.2 tuloksiin havaitaan, että muut ovat keskimäärin saaneet parempia tuloksia. Erityisesti verifioinnissa on päästy huomattavasti saamaamme 17.2 % pienempiin EER-virheisiin. Tulosten vertailu ei kuitenkaan ole kovin tasavertaista, koska puhujien määrä on erilainen ja lähes poikkeuksetta muut ovat käyttäneet huomattavasti pidempiä opetus- ja testisignaaleita, jopa useita minuutteja puhetta. Vaikka parhaimmillaan pääsimmekin vain 77.5 %:een identifiointiprosenttiin ja 17.2 %:n EER-virheeseen verifioinnissa, on tämä olosuhteet huomion ottaen yllättävän hyvä tulos: opetus- ja testidata oli erittäin rajoitettua ja lisäksi kohinaista ja laadultaan kaukana cd-tasoisesta äänestä. Tunnistuksen hyvyyteen saattoi tosin osaltaan vaikuttaa se, etteivät alkuperäiset nauhoitusolosuhteet ole olleet yhdenmukaiset: televisiohaastatteluissa on käytetty esimerkiksi erilaisia mikrofoneja ja taustahälyn osuus on aina hieman erilainen; voi siis olla, että osa puhujista tunnistettiin olosuhdekohtaisten parametrien eikä puhujan itsensä perusteella.

Tunnistukseen liittyvät perusongelmat tulivat myös hyvin tekemissämme testeissä. Kepstri ei toimi yhtä hyvin kohinaisella kuin puhtaalla datalla. Identifioinnin laskennallinen raskaus tuli myös ikävällä tavalla esille kun testit suoritettiin nopeasti koodatuilla Matlab-rutiineilla. Jokaista yksittäistä testiä varten kaikkien puhujien koodikirjat täytyi muodostaa uudelleen. Testausvaiheessa testisignaalista johdettuja piirteitä piti verrata kaikkien 40 puhujan malleihin ja tämä toistettiin kaikille 40 signaalille, eli yhteensä 1600 vertailua yhtä testiä kohden. Tämä vei kohtuuttomasti aikaa, joten testit piti

suorittaa eräajoina mm. öisin. Tästä nähdään että mikäli aiotaan toteuttaa oikea puhujan tunnistusjärjestelmä, on syytä kiinnittää heti alusta alkaen huomiota tehokkaiden tietorakenteiden suunnitteluun.

Seuraavaan taulukkoon on koottu yleisesti hyväksyttyjä – ja intuitiivisesti selviä – tekijöitä, joilla on vaikutusta tunnistustarkkuuteen. Tutkijat ovat saaneet myös näissä toisistaan poikkeavia tuloksia. Useimmat esimerkiksi painottavat, että puhujan äänessä ajan mittaan tapahtuvat variaatiot ovat ”eräs tärkeimmistä kysymyksistä” puhujan tunnistuksessa [4, 30, 33]. Kuitenkin esimerkiksi Furui [15] testasi puhujan verifiointia muuttamalla opetus- ja tunnistusvaiheiden välisen eron 6 päivästä 6 viikkoon eikä havainnut merkittävää muutosta tarkkuudessa.

| **Tunnistusta parantavia tekijöitä** | **Tunnistusta huonontavia tekijöitä** |
|---|---|
| + Opetus- ja testidatan pituuksien kasvattaminen | - Opetus- ja testidatan pituuksien lyhentäminen |
| + Samat olosuhteet opetuksessa ja tunnistuksessa | - Vaihdellaan olosuhteita (laitteisto, siirtokanava jne.) |
| + Mallien jatkuva päivittäminen / adaptoiminen | - Pitkä aikaväli (viikkoja) opetus- ja testivaiheiden välillä |
| + Tunnistettavan halukkuus tulla tunnistetuksi | - Haluttomuus, tahallinen äänen muuttaminen |

**Taulukko 7.3.** Joitakin yleisesti hyväksyttyjä tunnistukseen vaikuttavia tekijöitä.

### *7.3 Ideoita jatkotutkimusta varten*

Sekä kirjallisuudessa esitettyjen että omien empiiristen testien valossa automaattinen puhujan tunnistaminen näyttäisi lupaavalta käyttäjän tunnistustekniikalta. Turvallisuussovelluksia ajatellen ottamalla mukaan useita muitakin modaliteetteja voidaan päästä hyvinkin korkeisiin tunnistusprosentteihin. Esimerkiksi Jourlin & al. [20] kokeilivat akustisten piirteiden yhdistämistä huulien liikkeiden tarkasteluun puheen aikana. He saivat 37 puhujan tietokannalla pudotettua FA-virheen 2.3 %:sta 0.5 %:iin yhdistämällä akustiset ja visuaaliset modaliteetit. Brunelli ja Falavigna [4] yhdistivät puhujan ja kasvojen tunnistuksen. Pelkät akustisiin ja visuaalisiin piirteisiin perustuvat tunnistusprosentit 155 henkilön tietokannalla olivat 88 % ja 91 %, mutta näiden integroiminen nosti tunnistarkkuuden 98 %:iin! Mikäli yksittäisten modaliteettien suhteen ei voida enää päästä tiettyjä rajoja ylemmäs, hedelmällinen lähtökohta olisi tutkia kuinka näitä voidaan yhdistellä luotettavasti.

Furui [16] pohdiskelee koko joukkoa kysymyksiä, jotka ovat vielä avoimia puhujan tunnistuksessa. Esimerkiksi: onko parempia piirteitä kuin kepstri, pitäisikö teknisten kysymysten sijaan tutkia lisää myös ihmisen puhujantunnistusmekanismia, voidaanko kaksi eri puhujaa todellakin erottaa jonkin formaalin similariteettimitan avulla, voidaanko todellisissa tilanteissa koskaan saavuttaa 100 %:n tunnistettavuutta, miten kilpailukykyinen puhujan tunnistaminen on muiden henkilön identifiointimenetelmien kanssa jne.

### 7.3.1 Suprasegmentaaliset piirteet – uusi menetelmä

Tässä alakohdassa ehdotamme uutta suprasegmentaalisiin piirteisiin liittyvää lähestymistapaa puhujan tunnistukseen. Ihmisten puhetapa on erilainen ja tempoinformaation yhdistäminen spektraalisten piirteiden kanssa voisi parantaa tunnistusprosentteja.

Oletetaan seuraavassa, että spektraaliset piirteet on jo selvitetty ja eri puhujien $i$, $i$=1, …, $S$, spektraaliset koodikirjat $C_i = \{\boldsymbol{c}_{i1}, \boldsymbol{c}_{i2}, \ldots, \boldsymbol{c}_{iK}\}$ on konstruoitu jollain VQ-algoritmilla. Oletetaan, että meillä on käytössämme luotettava segmentointialgoritmi, jolla saamme segmentoitua signaalin dynaamisiin segmentteihin, jotka vastaavat jotain akustista yksikköä. Tämän jälkeen kullekin puhujalle lasketaan tietyn yksikön keskipituudet opetussignaalista. Idea on formuloitu seuraavan algoritmin muotoon.

ALGORITMI 7.3.1 (Segmenttien keskipituuksien selvittäminen)

**FOR EACH** puhuja $i$ **DO**
(1) Segmentoi opetussignaali dynaamisiin segmentteihin $F_j$ ;
(2) **FOR EACH** segmentti $F_j$ **DO**
(3) Laske piirrevektorit $F_j$ :lle;
(4) Etsi koodikirjasta $C_i$ lähin koodivektori $\boldsymbol{c}_{ib}$ piirrevektoreille;
(5) Aseta *pituus*[$b$] := segmentin pituus;
**ENDFOR;**
(6) Laske eri segmenttien keskipituudet taulukkoon *keskipituus*[1..$K$];
**ENDFOR;**

Tämän jälkeen tunnistaminen tapahtuu seuraavasti. Lasketaan ensin signaalista erityyppisten segmenttien lukumäärät, jonka jälkeen signaalin pituudelle lasketaan ”ennuste” summaamalla segmenttien lukumäärän ja vastaavien keskipituuksien tulot. Tätä ennustetta verrataan signaalin todelliseen pituuteen ja jos itseisarvon erotus alittaa annetun kynnysarvon, annetaan positiivinen tunnistuspäätös, muutoin negatiivinen.

Edellinen menetelmä on näennäisen helppo ja intuitiivinen, mutta vaatii toimiakseen luotettavan segmentointialgoritmin. Voidaan kokeilla vaikkapa ”ad hoc” –tyylistä segmentointia, joka perustuisi esimerkiksi kepstrikerrointen muutokseen: kun segmentti vaihtuu toiseksi, spektrin muoto ja siten myös kepstrikertoimet muuttuvat. Ehdotettuun menetelmään liittyy kuitenkin myös muita kysymyksiä, jotka pitäisi ratkaista ennen testaamista.

# VIITELUETTELO


[1] Ainsworth W.A.: *Mechanisms of Speech Recognition*. Pergamon Press Ltd., Oxford, England, 1976.

[2] Borden G.J., Harris K.S.: *Speech Science Primer. Physiology, Acoustics, and Perception of Speech*. Second Edition. Williams & Wilkins, Baltimore, 1984.

[3] Borenius J., Jauhiainen T., Lampio E., Nuotio J., Pesonen K., Pyykkö I.: *Akustiikan perusteet*. Insinööritieto Oy, 1981.

[4] Brunelli R., Falavigna D.: ”Person Identification Using Multiple Cues”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(10): 955-966, 1995.

[5] Campbell Jr. J.P., Reynolds D.A.: ”Corpora for the Evaluation of Speaker Recognition Systems”. *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*: 829-832. Phoenix, USA, March 1999.

[6] Castellano P.J., Slomka S., Sridharan S.: ”Telephone Based Speaker Recognition Using Multiple Binary Classifier and Gaussian Mixture Models”. *Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*: 1075-1078. Munich, Germany, April 1997.

[7] Chui C.K.: *An Introduction to Wavelets*. Mass. Academic Press, Boston, 1992.

[8] Deller Jr. J.R., Proakis J.G., Hansen J.H.L.: *Discrete-Time Processing of Speech Signals*. Macmillan Publishing Company, New York, 1993.

[9] Fant G.: *Acoustic Theory of Speech Production*. The Netherlands, Mouton, 1970.

[10] Fant G.: *Speech Sounds and Features*. The MIT Press, Cambridge, Massachusetts, 1973.

[11] Farrell K.R., Mammone R.J., Assaleh K.T.: ”Speaker Recognition Using Neural Networks and Conventional Classifiers”. *IEEE Transactions on Speech and Audio Processing*, **2**(1): 194-205, 1994.

[12] Feuer A., Goodwin G.C.: *Sampling in Digital Signal Prosessing and Control*. Birkhäuser, Boston, 1996.

[13] Finan R.A., Sapeluk A.T., Damper R.I.: ”Comparison of Multilayer and Radial Basis Function Neural Networks for Text-Dependent Speaker Recognition”. *Proc. 1997 IEEE International Conference on Neural Networks*: 1992-1997. Washington, DC, USA, June 1996.

[14] Finan R.A., Sapeluk A.T., Damper R.I.: ”Impostor Cohort Selection for Score Normalization in Speaker Verification”. *Pattern Recognition Letters*, **18**: 881-888, 1997.

[15] Furui S.: ”Cepstral Analysis Technique for Automatic Speaker Verification”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **29**(2): 254-272, 1981.

[16] Furui S.: ”Recent Advances in Speaker Recognition”. *Pattern Recognition Letters*, **18**: 859-872, 1997.

[17] Gagnoulet C., Couvrat M., Jouvet D.: ”Seraphine: A Connected Word Recognition System”. In: Haton J-P. (ed.): *Automatic Speech Analysis and Recognition*: 205-215. D. Reidel Publishing Company, Dordrecht, Holland, 1982.

[18] He J., Liu L., Palm G.: ”A New Codebook Training Algorithm for VQ-based Speaker Recognition”. *Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*: 1091-1094. Munich, Germany, April 1997.

[19] Iivonen A.: *Puheen tuottamismekanismi*. 2. versio. Oulun yliopiston fonetiikan laitoksen monisteita, Oulu, 1973.

[20] Jourlin P., Luettin J., Genoud D., Wassner H.: ”Acoustic-labial Speaker Verification”. *Pattern Recognition Letters,* **18**: 853-858, 1997.

[21] Kaukoranta T., Fränti P., Nevalainen O.: ”Iterative Split-and-merge Algorithm for Vector Quantization Codebook Generation”. *Optical Engineering*, **37**(10): 2726-2732, 1998.

[22] Kersta L.G.: ”Voiceprint Identification”. In: Dowden, Hutchinson & Ross, Inc.: *Speech Intelligibility and Speaker Recognition*: 425-429. Stroudsburg, 1977.

[23] Kohonen T.: *Self-Organizing Maps*. Springer-Verlag, Heidelberg, 1995.

[24] Lei J., Hall L.O.: ”Speaker Recognition with a Self-Configuring Neural Network”. *Proc. 1997 IEEE International Conference on Neural Networks*: 2351-2354. Houston, USA, June 1997.

[25] Levinson S.E., Rabiner L.R.: ”A Task-Oriented Conversational Mode Speech Understanding System”. In: Schroeder M.R.(ed): *Speech and Speaker Recognition*: 149-196. S Karger AG, Basel, 1985.

[26] Linde Y., Buzo A., Gray R.M.: An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications,* **28**(1), 84-95, 1980.

[27] Mathew M., Yegnanarayana B., Sundar R.: ”A Neural Network-Based Text-Dependent Speaker Verification System Using Suprasegmental Features”. *Proc. 6th European Conference on Speech Communication and Technology 1999 (Eurospeech’99)*: 995-998. Budabest, Hungary, Sept. 1999.

[28] Matsui T., Furui S.: ”Text-Independent Speaker Recognition Using Vocal Tract, Pitch Information”. *Proc. International Conference on Spoken Language Processing*: 137-140, Kobe, Japan, Nov. 1990.

[29] Oglesby J., Mason J.S.: ”Speaker Recognition with a Neural Classifier”, *Proc. IEEE Int. Conference on Artificial Neural Networks*, 306-310. London, Nov. 1989.

[30] Olsen J.∅.: *Phoneme Based Speaker Recognition*. PhD Thesis. Center for PersonKommunikation, Aalborg University, Denmark, 1997. Ladattavissa myös URL-osoitteesta http://www.kom.auc.dk/~jo/papers.html (15.12.1999).

[31] Peacocke R.D., Graf D.H.: ”An Introduction to Speech and Speaker Recognition”, *IEEE Computer*, **23**(8): 26-33, 1990.

[32] Proakis J.G., Manolakis D.G.: *Digital Signal Prosessing. Principles, Algorithms and Applications*. Second Edition. Macmillan Publishing Company, New York, 1992.

[33] Soong F.K., Rosenberg A.E., Juang B-H., Rabiner L.R.: ”A Vector Quantization Approach to Speaker Recognition”. *AT&T Technical Journal*, 66: 14-26, 1987.

[34] Suomi K.: *Johdatusta puheen akustiikkaan*. Oulun yliopisto, Logopedian ja fonetiikan laitoksen julkaisuja, N:O 4. Oulu, 1990.

[35] Wakita H.: ”Linear Prediction of Speech and Its Application to Speech Processing”. In: Haton J-P. (ed.): *Automatic Speech Analysis and Recognition*: 1-19. D. Reidel Publishing Company, Dordrecht, Holland, 1982.

[36] Wenndt S., Shamsunder S.: ”Bispectrum Features for Robust Speaker Identification”. *Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*: 1095-1098. Munich, Germany, April 1997.

[37] Young S.: ”A Review of Large-vocabulary Continuous-speech Recognition”. *IEEE Signal Processing Magazine*, Sept. 1996: 45-57, 1996.

[38] Zhang W.D., Yiu K.K., Mak M.W., Li C.K., He M.X.: ”A Priori Threshold Determination for Phase-Prompted Speaker Verification”. *Proc. 6th European*

*Conference on Speech Communication and Technology 1999 (Eurospeech'99)*: 1023-1026. Budabest, Hungary, Sept. 1999.

[39] Zilka R.D., Bistritz Y.: "Text Independent Speaker Identification Using LSP Codebook Speaker Models and Linear Discriminant Functions". *Proc. 6th European Conference on Speech Communication and Technology 1999 (Eurospeech'99)*: 799-802. Budabest, Hungary, Sept. 1999.