PUHEENKOODAUKSEN VAIKUTUS PUHUJANTUNNISTUKSEEN

Timo Viinikka

28.12.2004

Joensuun yliopisto
Tietojenkäsittelytiede
Pro gradu -tutkielma

TIIVISTELMÄ

Ihmisen tuottama puhe muunnetaan digitaaliseen muotoon, jotta sen käsittely tietokoneella olisi mahdollista. Puheenkoodaus, joka joissain tapauksissa täytyy tehdä resurssien säästämiseksi, hävittää informaatiota puheesta. Puhujantunnistus kärsii tästä koodauksesta, eikä koodatun materiaalin tunnistustarkkuus yllä enää alkuperäisellä materiaalilla suoritetun tunnistuksen tasolle. Puheenkoodaukseen on olemassa erilaisia menetelmiä, joista vaatimusten perusteella valitaan tilanteeseen sopivin. Tässä tutkielmassa perehdytään erilaisiin koodausmenetelmiin ja vertaillaan niistä muutamaa hyödyntäen olemassa olevaa puhujatietokantaa.

Avainsanat: puheenkoodaus, puhujantunnistus, GSM, ADPCM

SISÄLLYSLUETTELO

1 JOHDANTO

Nyky-yhteiskunnassa, jossa ihmisten liikkuminen on tullut yhä vapaammaksi ja helpommaksi, on henkilöiden tunnistamisesta on tullut tärkeä tekijä liikkumisen kontrolloinnissa. Varsinkin rikollisuuden ja terrorismin jatkuva kasvu on pakottanut valtiot kehittämään erilaisia tapoja seurata edes jollain lailla maahantuloa. *Biotunnistus* tekee tuloaan biopassin muodossa juuri tällä hetkellä. Yksi biotunnistuksen laji on ihmisen äänen perusteella tehtävä tunnistus, jossa analysoimalla puhetta päätellään kuka puhuja on. Puhetta voidaankin käyttää yhtenä tekijänä tunnistettaessa henkilöä. Aivan yhtä luotettavasti kuin sormenjäljistä ei ihmistä voi nykytekniikalla puheen perusteella päätellä. Menetelmänä puhujantunnistus on kuitenkin oiva lisä ja täydentäjä perinteisten tunnistusmenetelmien joukkoon (Kinnunen, 2003).

Ennen kuin puhujaa voidaan ryhtyä tunnistamaan, tarvitaan häneltä ääninäyte. Nyky-yhteiskunta on tehnyt tämänkin seikan yhä helpommaksi ja yleisemmäksi. Matkapuhelimet ovat levinneet 1990-luvulla räjähdysmäisesti ympäri maailman. Esimerkiksi rikostutkinnassa puhelinkuuntelun avulla voidaan saada ratkaisevia tietoja rikoksista ja niihin osallistuneista henkilöistä. Samalla kertyy aineistoa puhujarekisteriin, aivan kuten sormenjälkirekisteriin, jota voidaan myöhemmin käyttää hyväksi muidenkin rikosten tutkinnan yhteydessä.

Matkapuhelin sopii esimerkkitapaukseksi kuvattaessa tapahtumaketjua, joka johtaa puhujantunnistukseen. Henkilön puhuessa matkapuhelimeen hänen tuottamansa analogisen äänen vastaanottaa puhelimessa oleva pieni mikrofoni. Ääni muutetaan digitaaliseksi ja koodataan sopivaan muotoon ja lähetetään matkapuhelinverkkoon. Vastaanotettaessa verkosta tuleva koodattu puhe täytyy se koodata uudelleen takaisinpäin eli dekoodata, jotta siitä tulisi taas ymmärrettävää. Tämä koodausprosessi hävittää informaatiota puheesta. Kaikki puhelinta käyttäneet tietävät, että puhe on yhä ymmärrettävää, muttei enää yhtä selkeää kuin normaali puhuttu puhe. Matkapuhelimen tapauksessa koodaus tehdään GSM-koodausta käyttäen (Hillebrand, 2002), mutta olemassa on myös useita muita puheenkoodaukseen tarkoitettuja koodausmenetelmiä, jotka hieman

eri tavoin pyrkivät vähentämään puheen viemiä resursseja ilman, että puheen oleelliset ominaisuudet tästä kärsisivät.

Tehtiinpä puheen koodaus millä menetelmällä tahansa, katoaa aina informaatiota tässä prosessissa. Mutta vaikuttaako se puhujantunnistukseen, jossa ohjelmallisesti lasketaan puheen erilaisia ominaisuuksia ja pyritään niiden avulla tunnistamaan puhuja, siihen tämä tutkielma pyrkii antamaan vastauksen.

Tutkielman rakenne koostuu seuraavista luvuista. Aluksi käsitellään äänen- eli audionkoodausta yleisesti luvussa 2. Mistä ääni muodostuu, kuinka analoginen ääni muunnetaan digitaaliseksi ja kuinka digitaalista ääntä voidaan koodata, jottei se veisi tarpeettoman paljon resursseja esimerkiksi siirrettäessä sitä tietoverkossa. Luku 3 sisältää puheenkoodauksen teoriaa ja erilaisia menetelmiä, joita puheen koodaamiseen käytetään. Luvussa käsitellään GSM-koodauksen lisäksi *pulssikoodausmenetelmä* (PCM), *lineaarista ennustavaa koodausta* (LPC) ja Code Excited Linear Prediction (CELP). Luvussa 4 kerrotaan puhujantunnistuksesta. Millaisia käytännön sovellutuksia on olemassa, kuinka puhujantunnistusprosessi etenee ja mitä etuja sekä haittoja puhujantunnistukseen liittyy. Luvussa 5 on tutkittu GSM- ja ADPCM-koodausta ja niiden vaikutuksia puhujantunnistukseen. Nimensä mukaisesti ADPCM-koodaus on muunnos PCM-koodausstandardista, joka on yleisin käytössä olevista koodausmenetelmistä (Chen, 1992). Tutkimuksen puhujatietokantannaksi on valittu TIMIT.

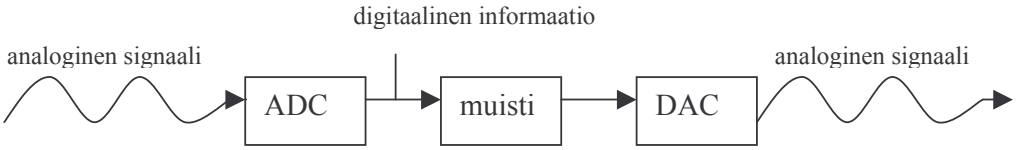
2 AUDIONKOODAUS

Ääni on ilmassa kulkevia aaltoja, joihin ihmiskorva reagoi lähettäen havaitsemansa ärsykkeet aivoihin, joissa ne tulkitaan kuulemaksemme ääneksi. Ääni on ilmanpaineen vaihtelua, joka syntyy, kun ilma värähtelee jonkin esineen liikkeen tai asian seurauksena. Tämä ilmanpaineen vaihtelu on mahdollista muuttaa esimerkiksi jännitevaihteluksi. Tällaista ääntä kutsutaan *analogiseksi*.

Tietokoneessa ääni on aina digitaalisessa muodossa ja äänen käsittely tapahtuu aina digitaalisesti. Analoginen ääni saadaan tietokoneen käsiteltäväksi mikrofonin tai jonkin muun äänilähteen välityksellä. Ilmassa olevat ääniaallot muuttuvat esimerkiksi mikrofonissa sähköiseksi, analogiseksi värähtelyksi, joka syötetään tietokoneen analogia/digitaalimuuntimeen.

2.1 ÄÄNEN DIGITOINTI

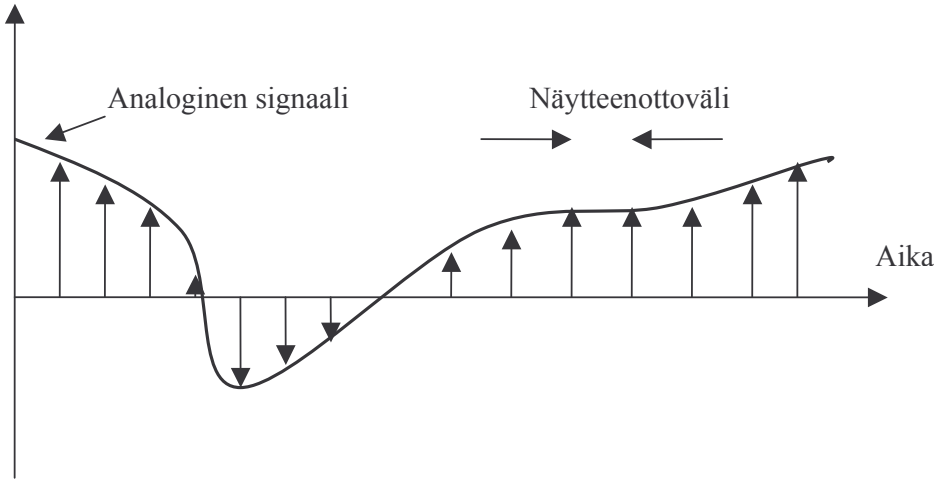
A/D-muunnin (*Analog/Digital Converter*) digitoi tulevan analogisen signaalin digitaaliseksi (kuva 1). A/D-muunnin ottaa analogisesta signaalista näytteitä halutuin tasavälein muodostaen poikkileikkauksen, josta saadaan aallon korkeus eli *amplitudi* katkaisuhetkellä (Holm 1998). Tämä tieto tallennetaan numeerisessa muodossa koneen muistiin. Tässä vaiheessa näytteenottotaajuus vaikuttaa äänen taajuuskaistaan eli äänen kirkkauteen. A/D-muunnoksen tasojen määrä eli *kvantisointi* määrää digitoidun äänen *dynamiikan* eli hiljaisten ja voimakkaiden äänien suhteen. Nämä molemmat ominaisuudet vaikuttavat digitoidun äänen sisältämän informaation määrään. Otettaessa tulevasta analogisesta äänisignaalista näytteitä riittävän tiheästi, saadaan siitä hyvälaatuinen toisto digitaalisessa muodossa. Näin saadun digitaalisen äänen käsittely on mahdollista hyvin monipuolisesti käyttämällä käyttötarkoituksen mukaan valittuja malleja ja algoritmeja.



Kuva 1. A/D- ja D/A-muunnoksen informaatio analogisen ja digitaalisen signaalin välillä.

A/D-muunnin (*Analog-To-Digital Converter ADC*) on laite, joka muuntaa analogisen äänisignaalin digitaaliseksi, koneen ymmärtämäksi informaatioksi. Vastaavasti *D/A-muunnin* (*Digital-To-Analog Converter DAC*) muuntaa digitaaliseen signaalin analogiseksi. A/D-muunnin vastaanottaa analogisen jännitepulssin ja muuntaa sen diskreettien lukujen sarjoiksi, jotka ovat yleensä 8-, 16-, 32- tai 64 bittisiä sanoja. Tätä prosessia kutsutaan *digitoinniksi* (*digitizing*) tai *näytteistykseksi* (*sampling*) (Holm, 1998). Toiseen suuntaan signaalin muuntaminen tapahtuu D/A-muuntimella, joka kääntää bittijonon jatkuvaksi analogiseksi jännitteeksi. Tallennetun digitaalisen äänen laatu ja vastaavuus alkuperäiseen verrattuna riippuu siitä, kuinka tarkka kuva alkuperäisestä signaalista on tallennettu. Tallennukseen laatuun vaikuttavat eniten näytteistystaajuus ja näytekoko.

Näytteistystaajuus (*sampling rate*) kertoo kuinka tiheästi näytteitä on alkuperäisestä signaalista otettu (kuva 2). Näytteistystaajuus ilmoitetaan *taajuutena*, eli *herzeinä* (*n* kertaa sekunnissa), mikä tarkoittaa näytteiden määrää sekunnissa (Holm, 1998). Yksittäinen näyte on siis analogisen signaalin tila tietyllä ajanhetkellä ja saadut näytteet yhdessä muodostavat diskreetin aikasignaalin.



Kuva 2. Näytteenotto analogisesta signaalista (Kettunen, 2003).

Mitä tiheämmin näytteitä otetaan, sitä laadukkaampi ja alkuperäistä tarkemmin vastaava näyte kokonaisuudessaan saadaan (taulukko 1). Koska ääni on ajan mukaan eteneviä paineaaltoja, määrää näytteistystiheys sen, kuinka etenkin korkeat äänet tallentuvat. Äänen korkeus riippuu sen värähtelytaajuudesta. Korkeat äänet värähtelevät tiheämmin kuin matalat ja jos näytteenottotaajuus on esimerkiksi 11 kHz ja äänen taajuus 15 kHz, ei kaikista värähdyksistä saada lainkaan näytettä. Nyquistin teoreeman mukaan tietyntaajuisen äänen toistumiseen vaaditaan vähintään kaksinkertainen, ja korkealaatuiseen toistumiseen vähintään nelinkertainen näytteistystaajuus (Holm, 1998). Näytteistystaajuudet ilmoitetaan aina kanavakohtaisesti, eli esimerkiksi stereoäänestä otettu 8000 näytettä sisältää yhteensä 16 000 näytettä.

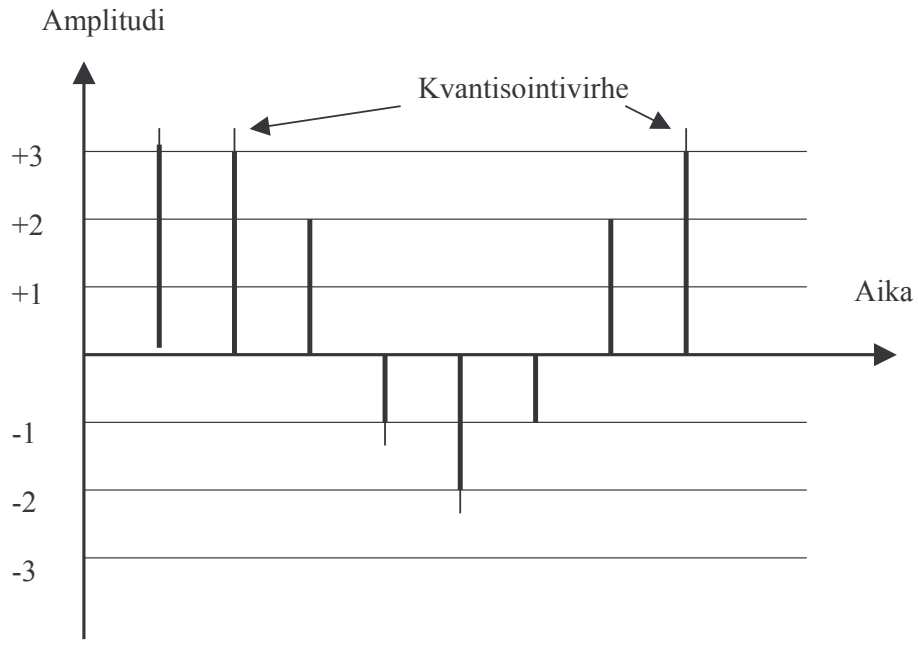
Taulukko 1. Yleisimmät näytteistystaajuudet.

Näytteistystaajuus	Kuvaus
44,1 kHz	CD-äänilevyn standardi
22,1 kHz	keskiaaltoradion taso
11,025 kHz	puheen tallennus, hyvän puhelinlinjan taso
5,5 kHz	vastaa huonon puhelinlinja tasoa

Toinen digitoidun äänen laatuun vaikuttava tekijä on näytteen koko. Signaalin *erottelukyky* eli *resoluutio* (*resolution*) ilmaiseen kuinka paljon bittejä on käytetty kuvaamaan äänisignaalin kunkin pisteen amplitudia. Mitä suurempi resoluutio on, sitä suurempi määrä erillisiä amplitudiarvoja voidaan esittää (Holm, 1998). Esimerkiksi normaalissa audio-CD:ssä on käytössä 16-bitin resoluutio, mikä mahdollistaa jokaisen audiosignaalin pisteen amplitudin esittämistä 65535 eri arvolla.

Resoluution bittimäärä määrää myös digitaalisen signaalin dynamiikan. *Dynamiikka* ilmaisee äänisignaalin äänenvoimakkuudeltaan hiljaisimman ja kovimman kohdan suhteen. Jotta päästäisiin mahdollisimman lähelle ihmiskorvan dynamiikkaa vastaa jokainen bitti 6 desibeliä (dB), eli 8-bittinen audio mahdollistaa 48 dB dynamiikan ja 16-bittinen CD-audio 96 dB:n dynamiikan.

Äänen digitointiin liittyy keskeisesti myös käsite kvantisointi (kuva 3). *Kvantisointia* (*quantization*) esiintyy silloin, kun analogisen lähdesignaalin arvot sijoittuvat johonkin erottelukyvyn mahdollistavien arvojen väliin ja täsmälleen oikeaa digitaaliarvoa ei ole. Tällöin analoginen arvo pyöristetään eli kvantisoidaan lähimpään mahdolliseen digitaaliseen arvoon (Holm, 1998). Kvantisoinnissa muodostuu eräänlainen porrasefekti digitaalisten arvojen välille, joka on mahdollista havaita digitaalisen äänen luonnottomuutena, esimerkiksi rakeisena tai suhisevana äänenä. Tätä virhettä kutsutaan *kvantisointikohinaksi* (Kettunen, 2003). Nämä virheet vähenevät resoluution kasvaessa. Kvantisoinnissa täytyy siis huomioida ääneen mahdollisesti muodostuvat virheet. Suuriamplitudisella signaalilla virheiden mahdollisuus on hyvin pieni ja ilmenee analogisen kohinan tapaan. Sen sijaan matalatasoisilla signaaleilla virheet ilmenevät äänen *säröytymisenä* (*distortion*).



Kuva 3. Kvantisoinnissa tapahtuva vääristyminen (Kettunen, 2003).

2.2 ÄÄNEN KOODAUS

Tiedon koodauksella tarkoitetaan muunnosta, jossa jokin datasyöte muunnetaan alkuperäisestä eroavaan toiseen, koodattuun muotoon. Toimenpide voidaan tehdä myös käänteisesti, jolloin dekoodausalgoritmilla muunnetusta muodosta saadaan alkuperäinen data. Koodaus ja sen toimivuus perustuu datan sisältämään ennustettaviin piirteisiin, joita koodausalgoritmissa voidaan hyödyntää (Vaarala, 1998). Perinteisimmät koodausalgoritmit eivät ota kantaa datan sisältöön, vaan ne käsittelevät kaikki syötteet samalla lailla. Parempia pakkaussuhteita voidaan saavuttaa, kun käsiteltävästä datasta voidaan olettaa tai siitä tiedetään jotain. Entistä parempia pakkaussuhteita saavutetaan, kun huomioidaan datan luonne ja käyttö, eli koodausalgoritmille kerrotaan mikä datassa on oleellista ja sallitaan sen hävittää informaatiota pakkauksen aikana. Lisäksi koodausalgoritmeja kehitettäessä otetaan huomioon käyttökohde, jonka mukaan algoritmia voidaan entisestään kehittää tehokkaammaksi.

Äänen koodauksella pyritään pienentämään äänisignaalin viemiä resursseja, esimerkiksi signaalin vaatimaa bittien määrää. Koodauksen tarkoituksena on tietomäärää pienentämällä parantaa ja tehostaa tiedon siirtämistä tai tallennusta (Painter, 2000). Äänen koodauksen avulla pyritään lisäksi säilyttämään signaalin tuottama kuuloaistimus mahdollisimman samanlaisena verrattuna alkuperäiseen signaaliin. Äänen koodaukseen onkin kehitetty useita erilaisia tapoja näiden tavoitteiden saavuttamiseksi mahdollisimman tehokkaasti ja laadukkaasti

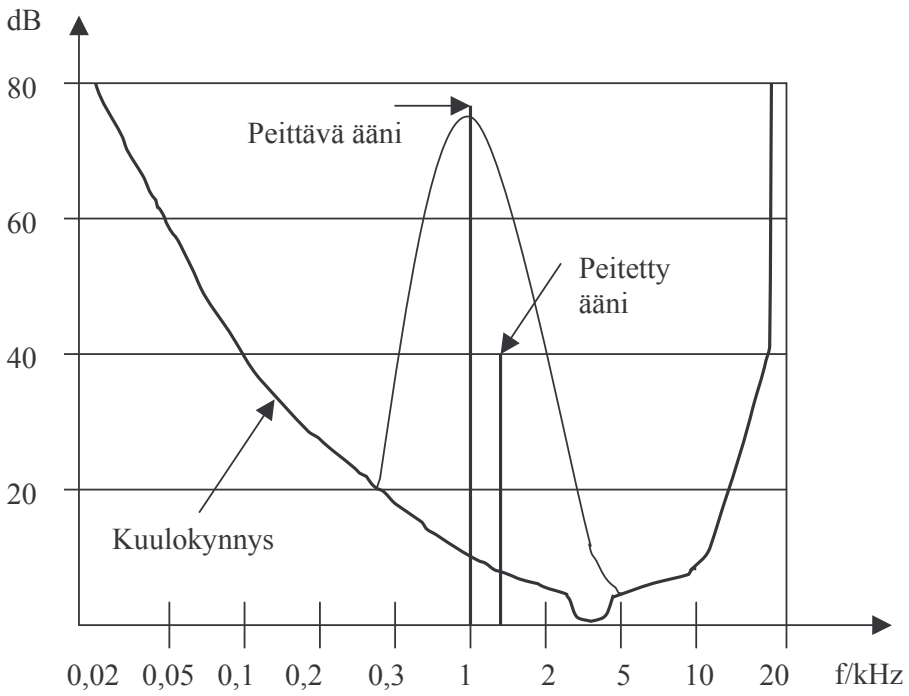
Alkuperäiselle signaalille tehdään ensin jokin yleinen *muunnos* (*transformation*), kuten vaikkapa taajuuskaistoihin jako, minkä jälkeen pakkausalgoritmin on helpompi havaita säännönmukaisuuksia tai käyttötarkoitusta ajatellen merkityksettömiä osia. Audiosignaalista voidaan jättää esimerkiksi hiljaisia tai muita ihmisen kuulemattomia hetkiä tallentamatta, poistaa usein toistuvia komponentteja tai muuttaa koodausta esimerkiksi tallentamalla kahden näytteistetyn arvon välinen erotus varsinaisten arvojen sijaan. Lisäksi hyödynnetään ihmisen kuulokynnystä. Kuulokynnys on äänentaso, jolla ihminen pystyy kuulemaan tietyntaajuisen äänen (Holm, 1998). Parhaimmillaan ihmisen kuulokynnys on 4 kHz:n taajuudella ja muutenkin erityisen alhainen 1-5 kHz:n taajuusalueella, jolloin kuulokynnyksestä ei ole hyötyä audionkoodauksessa. Sen sijaan korkeimmilla äänillä, joiden taajuus on yli 16 kHz, ihmisen kuulokynnys nousee jyrkästi. Näiden lisäksi audionkoodauksessa hyödynnetään niin sanottua kriittistä kaistaa. Ihmiskorvan fyysisistä ominaisuuksista johtuen ihminen analysoi ääntä siten, että tietyn kriittisen kaistan alueelle osuvat äänet käsitellään yhtenä kokonaisuutena. Esimerkiksi MPEG-koodauksessa (Riederer, 2000) hyödynnetään tätä ihmiskorvan ominaisuutta jakamalla signaali lineaarisesti taajuuskaistoihin.

Laadultaan hyvää tai erinomaista audionkoodausta saavutetaan erilaisilla taajuuspainotetuilla *koodekeilla*. Koodekki on ohjelma, joka koodaa alkuperäisen datan haluttuun muotoon käyttämällä yhtä tai useampaa algoritmia lopputuloksen saavuttamiseksi. *Taajuuspainotettuja koodekkeja* (*frequency domain codecs*) ovat muun muassa *alikaistakoodausta* (subband coding, SBC) ja *mukautuvaa muunnoskoodausta* (*adaptive transform coding, ATC*) käyttävät koodekit (Holm, 1998). Erot eri koodekkien välillä muodostuvat lähinnä spektrikomponenttien

määrästä, kvantisointistrategioista ja koodausvirheiden peittämiseen käytettyjen menetelmien eroista. Spektrikomponentit sisältävät äänisignaalin ominaisuuksia eli signaalista laskettuja matemaattisia arvoja, joita toiset koodekit osaavat hyödyntää enemmän kuin toiset.

2.2.1 Taajuuspainotettu koodaus

Taajuuspainotetussa koodauksessa alkuperäisen signaalin sisältämää tiedon määrää pyritään vähentämään muokkaamalla siitä lyhyempimuotoista äänispektritietoa eli jakamalla signaali pienempiin osiin. Lisäksi käytetään hyväksi korvan peitto-ominaisuuksia. Peittoääni syntyy, kun hiljaisempi, mutta muutoin selvästi kuultava signaali muuttuu kuulumattomaksi voimakkaamman signaalin vaikutuksesta (kuva 4). Taajuuspainotetut koodekit tarjoavat myös menetelmän taajuuskomponentin häiriöiden muokkaamiseen ja vaimentamiseen niin, ettei niitä lähetetä edelleen.



Kuva 4. Ihmisen kuulokynnys ja korvan peitto-ominaisuus (Hanhijärvi, 1998).

Tulevan signaalin spektri jaetaan aluksi taajuuskaistoihin eli taajuuskomponentteihin. Kukin taajuuskomponentti on kvantisoitu erikseen kuvan 3 mukaisesti halutulla määrä bittejä. Kvantisointiin käytettävien bittien määrä taajuuskomponentissa vaihtelee sen mukaan, kuinka tärkeä komponentti on kyseessä (Holm, 1998). Mitä tärkeämpi komponentti on, sitä tarkemmin se kvantisoidaan. Komponentin tärkeys riippuu sen sisältämän äänen taajuudesta. Esimerkiksi puheen ymmärtämisen kannalta oleellisin taajuusalue on 50 Hz ja 3400 Hz väliin jäävä alue.

Taajuuspainotteisessa koodauksessa esikaiuilla on erittäin tärkeä merkitys. *Esikaiku* (*pre-echo*) voi syntyä esimerkiksi silloin, kun hiljaista kohtaa seuraa jokin voimakas perkussio- eli lyömäsoittimen aiheuttama ääni samassa koodausalueessa. Tällöin koodauksessa voi tapahtua verraten suuri hetkellinen kvantisointivirhe, joka voi muodostua selvästi kuuluvaksi varsinkin, jos kvantisoinnissa käytetään alhaisia bittimääriä. Esikaiut voidaan välttää käyttämällä lyhyempiä koodausalueita.

2.2.2 Alikaistakoodaus

Alikaistakoodauksessa lähdesignaali syötetään analysoivaan suodinpankkiin. *Analysoiva suodinpankki* (*analysis filter bank*) sisältää *n* kaistansuodinta taajuudellisesti rinnakkain niin, että sarja alikaistasignaaleja voidaan lisätä yhdistellen niitä halutulla tavalla ja näin tuottaa lähellä alkuperäistä signaalia oleva koostettu signaali (Holm, 1998). Jokaisen suotimen ulostulevaa signaalia on häivytetty määrällä *n*, joka on summa alkuperäiseen signaaliin verrattavista alikaistasignaaleista. Jokaisen suotimen signaali on kvantisoitu erikseen.

Vastaanottopuolella jokaisen alikaistan näytteistystaajuutta lisätään vastaamaan alkuperäistä signaalia. Tämä toteutetaan lisäämällä signaaliin tarvittava määrä nollanäytteitä. Mikäli jälleenmuokkaussuotimet ovat täydelliset, saadaan summasignaalista vastaava kuin kvantisoimaton lähdesignaali oli. Alikaistakoodausta käytetään muun muassa MPEG-audiokoodauksessa.

2.3 MPEG-AUDIOFORMAATTI

Tietotekniikassa koko sen historian ajan kukin laitevalmistaja on käsitellyt audiodataa omalla tavallaan ja luonut oman tiedostotyyppinsä sen tallentamiseksi. Niinpä erilaisia formaatteja audion tallentamiseksi on runsaasti ja kaikki formaatit eivät ole keskenään yhteensopivia. Yleensä audioformaatit ovat muunnettavissa toiseksi, tosin joissain tapauksissa informaatiota hukkaavasti (Holm, 1998).

Audiodataa sisältävät tiedostoformaatit voidaan jakaa kahteen kategoriaan: itsekuvaileviin ja raakoihin tiedostoformaatteihin. *Itsekuvailevat* (*self-describing*) tiedostoformaatit sisältävät erillisen otsakkeen eli headerin, jota *raa'at* (*raw*) tiedostoformaatit eivät sisällä. Tässä otsakkeessa on määritelty mahdolliset laiteparametrit ja muuta koodauksessa käytettävää informaatiota. Raa'oissa formaateissa laiteparametrit ja itse koodaus on yhdistetty, eivätkä nämä otsakkeettomat formaatit salli parametrivariaatiota. Itsekuvailevat tiedostoformaatit muodostavatkin oman koodausperheensä, jossa otsakekentät kertovat tapauskohtaisesti käyttötarkoituksen.

Otsakekentissä määritellään koodauksessa tarvittavia tietoja, kuten erilaisia näytteistysparametreja. Näiden lisäksi otsakekentissä voidaan kertoa myös muuta informaatiota, kuten äänen kuvaus- ja tekijänoikeustietoja.

MPEG (Moving Pictures Experts Group) on *ISO*:n (*International Standard Organization*) eli kansainvälisen standardoimisliiton komitea, joka kehittää liikkuvan kuvan ja audion pakkaamisen standardeja (Watkinson, 2000). MPEG/audio on äänen kompressiostandardi, joka eroaa useista vokaalimalleihin perustuvista puheelle optimoiduista koodausmenetelmistä siten, ettei se oleta mitään lähteen luonteesta. MPEG/audio koodaa kaiken, mitä ihmiskorva voi kuulla karsien kuuloaistille epäolennaisen informaation. Standardissa on useita kompressiomoodeja eli audiota on mahdollista pakata tehokkaammin riippuen käyttötarkoituksesta ja käytettävissä olevista resursseista, kuten esimerkiksi verkon nopeudesta.

MPEG-komitea luo standardeja versioittain (taulukko 2). MPEG-1 ja MPEG-2 standardeja käyttävät useimmat äänen ja kuvan pakkausta tarvitsevat sovellukset. MPEG-4 -standardi on tarkoitettu alhaisia bittivirtoja käyttäville sovelluksille esimerkiksi Internetissä. MPEG-7 –standardi on kehitetty eritoten multimediasovelluksia varten. MPEG-21 –standardi on vielä kehitysasteella, joka toteutuessaan toisi viitekehyksen (framework), jonka kautta erilaisten multimedialähteiden käyttäminen verkon yli helpottuisi ympäri maailmaa erilaisilla laitteilla.

Taulukko 2. MPEG-standardit (MPEG Home Page, 2003).

Standardi	Kuvaus
MPEG-1	Koodaustapa liikkuvalle kuvalle ja äänelle. Resoluutio enintään 1.5Mbit/s.
MPEG-2	Kehittyneempi koodaustapa liikkuvalle kuvalle ja äänelle. Resoluutio enintään 20Mbit/s.
MPEG-4	Koodaustapa audio-visuaalisille objekteille.
MPEG-7	Koodaustapa erityisesti multimediasovelluksia varten.
MPEG-21	Kehitysasteella oleva multimedia-framework.

MPEG-1 ja MPEG-2 –standardien audio-osa määrittelee MPEG-audionkoodauksen perusrakenteen. MPEG-audionkoodaus koostuu kolmesta *kerroksesta* (*layer*), joita kutsutaan myös kolmeksi algoritmiksi. Kerrokset ovat tehokkuusjärjestyksessä, joista Layer 3 on tehokkain ja Layer 1 tehottomin ja samalla yksinkertaisin. Layer 3 tarvitsee koodaamiseen alhaisemman bittimäärän saavuttaakseen yhtä laadukkaan lopputuloksen kuin Layer 1. MPEG-audionkoodauksen perustana on *MUSICAM*-koodaus (Hanhijärvi, 1998), joka kehitettiin ennen MPEG-standardeja. MPEG-audionkoodauksessa tuleva PCM-audiosignaali muutetaan suodinpankin avulla ensiksi taajuusjakoiseksi ja jaetaan alikaistoihin, jotka kvantisoidaan vaihtelevalla bittinopeudella. Seuraavassa vaiheessa muodostetaan psykoakustinen malli käyttäen apuna *nopeaa Fourier-muunnosta* (*Fast Fourier Transform, FFT*). Sen avulla voidaan arvioida ihmisen kuuloaistin tilaa tietyssä kohtaa syötesignaalia ja laskea sitä tietoa hyödyntäen

tehotasot ja määrittää peittoäänten teho. Psykoakustisen mallin avulla arvioidaan taajuuskaistojen kuulokynnys ja se, kuinka paljon kohinaa kullakin kaistalla voi olla ilman, että kuulija huomaa mitään. Näin yritetään minimoida havaittu kvantisointikohina.

MPEG-dekoodekki on koodekkia yksinkertaisempi, koska sen ei tarvitse muodostaa malleja. Dekoodekin tarvitsee ainoastaan purkaa paketit, rakentaa kvantisoitu spektri ja muuntaa taajuusjakoinen koodi aikatasoon.

Käytettäessä Layer 3:a, monimutkaistuu koodekin rakenne huomattavasti verrattuna kahteen tehottomampaan Layeriin. Tämä johtuu pakkaussuhteen kasvamisesta juuri näiden Layerien välillä. Layer 3:n ominaisuuksiin kuuluu laadukas äänenlaatu hyvällä pakkaussuhteella. Pakkaussuhde voi olla esimerkiksi 12:1, jolloin saadaan vielä CD-tasoista ääntä ilman havaittavaa äänenlaadun heikkenemistä. Toisaalta tarkasteltaessa Layer 3-pakatun äänen taajuusanalyysiä tietokoneella, saatetaan huomata hyvinkin suuri heikennys verrattuna alkuperäiseen ääneen. Alhaisten bittitaajuuksiensa vuoksi Layer 3 sopii audiodatan siirtoon ja korkean äänenlaatunsa puolesta kaikenlaisen musiikin tallennukseen. MPEG-1:n Layer 3:lla tallennettua ääntä kutsutaan maailmanlaajuisesti mp3:ksi.

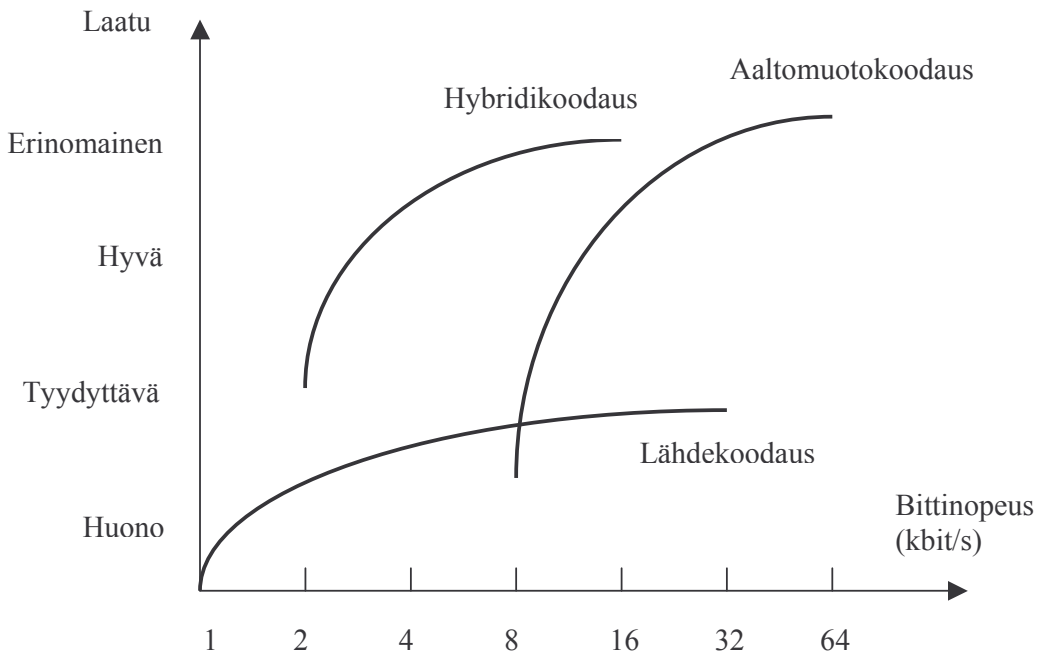
MPEG-4 –standardi on tarkoitettu erittäin alhaisia bittitaajuuksia käyttäville ja tarvitseville järjestelmille, kuten vaikkapa multimediasovelluksille langattomassa verkossa (Ylitalo, 1997). MPEG-4 –standardissa *bittitaajuus* voi vaihdella 2 – 64 kbit/s välillä. Bittitaajuus kertoo kuinka monta bittiä käytetään yhden sekunnin pituisen äänen koodaukseen. Alhaisten bittitaajuuksien lisäksi uutta verrattuna aiempiin MPEG-1- ja MPEG-2 –standardeihin on MPEG-4:n interaktiivisuus. Vastaanottaja voi esimerkiksi kesken tiedonsiirron vaihtaa bittitaajuutta ja muuttaa tarvittaessa tulevan signaalin stereosignaalista monosignaaliksi kasvattaen näin bittitaajuutta. Joustavamman pakkauksen ansiosta MPEG-4 sopiikin hyvin erilaisiin laitteisiin (Riederer, 2004). MPEG-4 –standardi ei ainoastaan pakkaa tietoa mahdollisimman tehokkaasti, vaan myös siirtää sitä kuvaamalla keskeisen sisällön. Tieto on paketoitu jäsennetyksi o*bjekteiksi* (*object*), jotka kuvaavat tiedosta audiovisuaaliset *otokset* (*scheme*). Otoksia ovat esimerkiksi videoleikkeet,

tietokonegrafiikka, luonnolliset ääninauhoitteet ja synteettinen ääni. MPEG-4 bittivirta tuotetaan pala kerrallaan missä kukin pala vastaa yhtä äänilähdettä. Kuva ja ääni kuvataan omalla kuvauskielellä, *bifs*:llä (*binary format for scene description*). Kuvauskieli asettaa yleiset säännöt äänelle, vuorovaikutteisuudelle, kolmiulotteiselle tilavaikutelmalle ja räätälöityjen efektien latautuselle. Näiden tarkka toteutus on ohjelmoijan itse päätettävissä, kunhan hän noudattaa kuvauskielen asettamia rajoja.

Bittitaajuuden vaihteluvälin suuruudesta johtuen MPEG-4 –koodauksessa luonnollisen äänen koodaukseen käytetään kolmenlaisia koodekkeja (Ylitalo, 1997). Alhaisimmilla bittitaajuuksilla (2 - 16 kbit/s) käytetään parametrista koodausta. Keskitaajuuksilla (6 – 24 kbit/s) voidaan käyttää ennustavaa CELP-koodausta. Bittitaajuuksien ylittäessä 16 kbit/s koodauksessa käytetään taajuustason käyttöön perustuvia AAC- ja TwinVQ -koodekkeja. Koska koodekkien käyttämät bittitaajuudet menevät osittain päällekkäin, on keskitaajuuksilla mahdollista käyttää käyttötarkoitukseen parhaiten sopivaa koodekkia.

3 PUHEENKOODAUS

Puheenkoodauksella tarkoitetaan puhetta sisältävän äänisignaalin digitalisointia ja koodausta ilman, että puhe muuttuu epäselväksi tai sen laatu kärsii liikaa. Useimmiten tähän pyritään lisäksi mahdollisimman pienellä bittimäärällä (Kettunen, 2003). Puhesignaalin, kuten kaiken äänisignaalin digitalisoinnin suorittaa koodekki. Hyvän puhekoodekin ominaisuuksia ovat mahdollisimman pieni puheenlaadun aleneminen, pieni bittimäärä, pieni viive, lähetyksen pieni virhealttius, nopeus, kohinansieto ja useamman peräkkäisen koodauksen mahdollisuus. Tällaista koodekkia ei kuitenkaan ole mahdollista toteuttaa, sillä osa ominaisuuksista on toisensa poissulkevia. Koodekin suunnittelu ja toteutus onkin yleensä kompromissi, johon päädytään ottamalla huomioon koodekin tuleva käyttötarkoitus ja -ympäristö.



Kuva 5. Rekonstruoidun signaalin puheen laatu bittinopeuden funktiona kolmella eri koodausmenetelmällä.

Yleisimmät puheenkoodausmenetelmät voidaan jakaa karkeasti kahteen pääryhmään (Kettunen, 2003): *aaltomuotokoodaukseen* ja *lähdekoodaukseen*. Lisäksi on kehitetty näiden kahden pääryhmän hyvät ominaisuudet yhdistävä *hybridikoodaus*. Kuvassa 5 on esitetty koodausmenetelmien eroja puheenlaadussa bittinopeuden funktiona.

Aaltomuotokoodauksessa pyritään säilyttämään analogisen signaalin aaltomuoto mahdollisimman samanlaisena verrattuna alkuperäiseen signaaliin. Koodaus perustuu analogisen signaalin näytteistämiseen ja näytteiden kvantisointiin. Aaltomuotokoodauksen avulla on mahdollista saavuttaa erittäin hyvä puhesignaalin laatu, koska rekonstruoitava signaali on suora kopio alkuperäisestä (Kettunen, 2003). Peruskoodina aaltomuotokoodauksessa on PCM, jonka variaatioita muut aaltomuotokoodausmenetelmät ovat.

Lähdekoodauksessa sen sijaan keskitytään parametrien koodaukseen aaltomuodon sijaan. Parametrien, esimerkiksi herätteen tyyppi ja formaattitaajuudet, avulla voidaan rekonstruoida alkuperäinen signaali. Lähdekoodauksessa puhesignaalia voidaan käsitellä pienillä bittinopeuksilla, mutta samalla menetetään suuri osa signaalin laadusta (Kettunen, 2003). Yleisin lähdekoodauksen menetelmä on *Linear Predictive* Coding (*LPC*).

Hybridikoodauksessa on yhdistetty näiden kahden menetelmän hyvät ominaisuudet eli aaltomuotokoodauksen hyvä puheenlaatu ja lähdekoodauksen tehokkuus. Hybridikoodeja on useampi ja niitä kehitetään jatkuvasti lisää, sillä sovellusten välillä liikkuvan tiedon määrä kasvaa koko ajan, jolloin vaaditaan yhä tehokkaampia toteutuksia (Kettunen, 2003). Hybiridkoodauksen perustekniikka on *Code Excited Linear Prediction* (*CELP*).

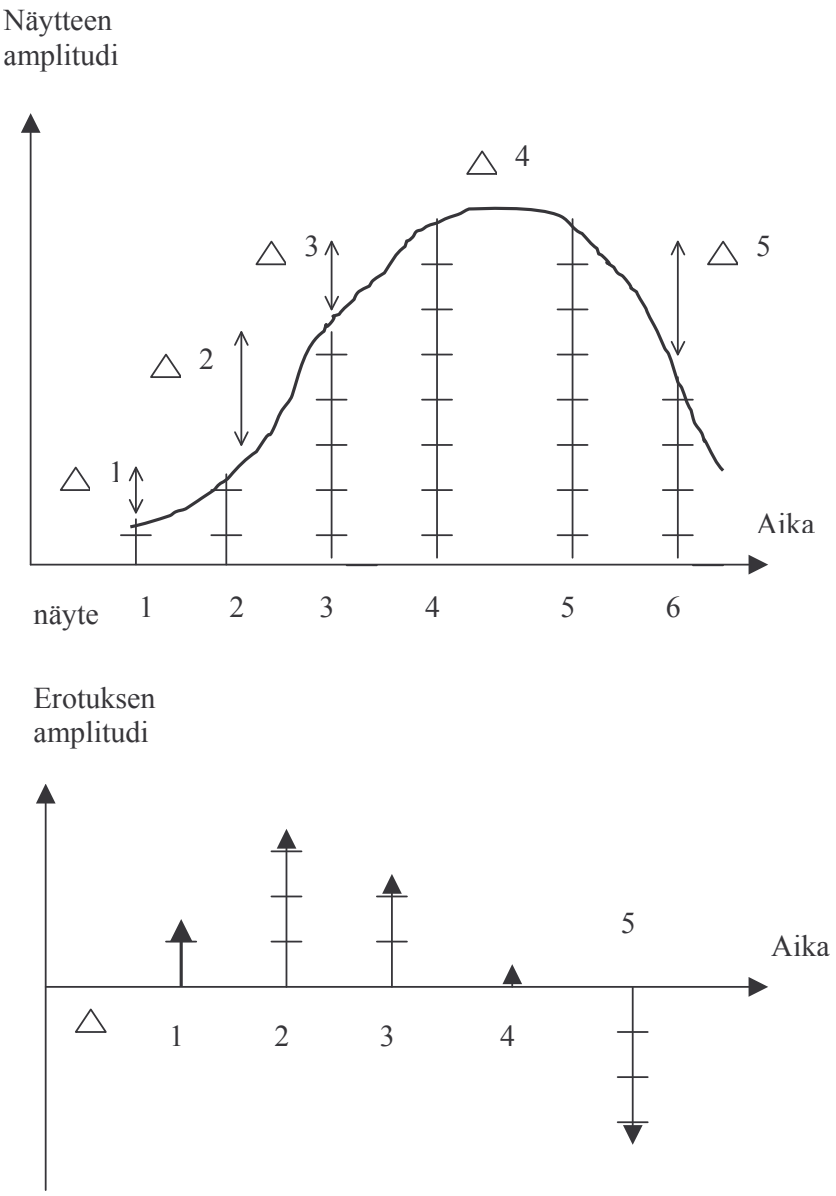
3.1 PULSSIKOODAUSMENETELMÄ (PCM)

Pulssikoodausmenetelmä (PCM Pulse Code Modulation) on yleisin ja vanhin analogisen äänen muunnosstandardi. Se kuuluu aaltomuotokoodeihin ja sitä käytetään analogisen signaalin digitoinnissa muun muassa digitaalisissa

puhelinverkoissa ja tehtäessä digitaalisia äänitallenteita. Pulssikoodausmenetelmä kehitettiin jo 1930-luvulla, mutta sen käyttö yleistyi vasto 1960-luvulla transistoritekniikan kehityttyä riittävästi (Kettunen,2003). PCM:stä on olemassa useampi standardi, joista ensimmäisen muodostavat A-law PCM Euroopassa ja μ-law USA:ssa. A-law ja μ-law algoritmit perustuvat ihmisäänen logaritmiseen aistimukseen, eli analogista ääntä kvantisoidaan pienillä tasoalueilla tiheämmin ja suuremmilla tasoalueilla harvemmin. Näin päästään vähemmän tilaa vievään lopputulokseen verrattuna tasavälein tapahtuvaan kvantisointiin, jonka PCM myös mahdollistaa. A-law ja u-law koodaavat dataa 64kbit/s taajuudella.

Toinen standardi on *adaptiivinen PCM* (*APCM Adaptive Pulse Code Modulation*). Perinteisessä PCM koodauksen kvantisoinnissa ei pystytä huomioimaan näytteiden muutoksia. Ongelman ratkaisuksi on kehitetty adaptiivinen PCM, jossa kvantisointitasot asetetaan muuttumaan näytteiden mukaisesti (Kettunen, 2003). APCM:n perusajatuksena on pienentää kvantisointiaskelta, jos kvantisoitavan signaalin taso on matala. Vastaavasti suurille amplitudeille kvantisointiaskelta suurennetaan. Verrattuna perinteiseen PCM:ään tällä menetelmällä saavutetaan parempi signaali/kohina-suhde ja siinä voidaan käyttää pienempää taajuuskaistaa, jolloin myös signaalin lähetystehokkuus paranee.

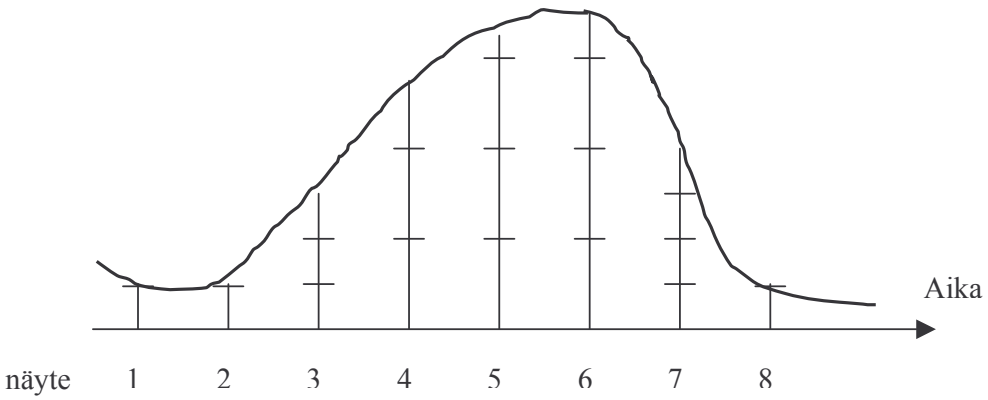
Kolmas standardi on *differentiaalinen PCM* (*DPCM Differential Pulse Code Modulation*). Puhesignaalista otettujen vierekkäiset näytteiden välillä on suuri korrelaatio. DPCM:n perusidea on lähettää kvantisoitujen näytteiden sijasta vierekkäisten näytteiden erotus (kuva 6), jolloin signaalin esittämisessa tarvittava dynaaminen alue pienenee. Tämän seurauksena kvantisointiin tarvitaan pienempi resoluutio ja samalla kvantisointikohina pienenee (Gold, 2000).



Kuva 6. Näytteiden välisen erotuksen (delta) määrittäminen analogisesta signaalista (Kettunen, 2003).

ADPCM (*Adaptive Differential PCM*) (Chen, 1992) yhdistää kahden viimeksi mainitun tekniikan ominaisuudet eli se on differentiaalisen ja adaptiivisen PCM:n yhdistelmä. ADPCM käyttämä bittitaajuus on 32kbit/s. ADPCM-koodauksessa differentiaaliseen tekniikkaan lisätään sekä adaptiivinen ennustaja että kvantisointi. ADPCM määrittää näytteiden eron ja mukautuu signaalin

ominaisuuksiin muuntamalla kvantisointi- ja ennustusparametreja (Gold, 2000). Kuvassa 7 on havainnollistettu ADPCM:n tapaa toteuttaa signaalin näytteenotto ja kvantisointi. Kuten edellisessäkin tekniikassa, erotuksen koodaamiseen tarvitaan vähemmän bittejä kuin koko näytteen koodaamiseen tarvittaisiin.



Kuva 7. Adaptiivisen DPCM:n näytteenotto ja kvantisointi, jossa kvantisointiaskel muuttuu näytteen kasvaessa riittävän suureksi (Kettunen, 2003).

Kaikessa digitaaliaudiossa on olemassa peruselementit ADC, DAC, bittiresoluutio ja näytteistystaajuus. Niin myös digitaaliäänittimissä. Digitaaliäänittimiin liittyy myös se, kuinka audio tallennetaan digitaaliselle tallennusvälineelle. Yleisin digitaalinen koodaustekniikka on PCM. Tallennuksessa bittivirta konvertoidaan ketjuksi kapeita suorakulmaisia pulssiaaltoja, jotka edustavat bittiarvoja. Tämä pulssien virta kirjoitetaan tallennusvälineelle. Kun tallennusvälineeltä toistetaan siihen tallennettua audiodataa, prosessi on käänteinen ja pulssit konvertoidaan bittiarvoiksi. Useimmiten digitaalinen informaatio kirjoitetaan tallennusvälineelle niin, että informaatiosta on olemassa varmuuskopio virheiden varalle.

3.2 LINEAARINEN ENNUSTAVA KOODAUS (LPC)

Lineaarinen ennustava koodaus (Linear Predictive Coding, LPC) on puhesignaalin parametrinen koodausmenetelmä. Siinä koodataan ainoastaan signaalin perusominaisuudet, kuten ennustajavakio ja herätefunktio (Kettunen, 2003). LPC perustuu ennustajaparametreihin, joiden avulla puhesignaalista saatujen

näytteiden ennustaminen on mahdollista edellisten näytteiden lineaarikombinaation avulla. Sovelluksissa käytetään 8-16 kbit/s nopeutta, jolloin puheen laatu ei huonone liikaa. Koodauksessa tarkastellaan audiodataa pienissä osissa ja sovitetaan kukin osa lähinnä sitä vastaavaan analyyttiseen malliin (Gold, 2000). Tämän jälkeen puhe erotetaan pois ja parametrit lähetetään sopivassa muodossa. LPC-purkaja osaa näiden parametrien avulla muodostaa synteettistä puhetta, joka on ymmärrettävää, mutta konemaista.

3.3 CODE EXCITED LINEAR PREDICTION (CELP)

Code Excited Linear Prediction (CELP) on etenkin avoimissa järjestelmissä käytettävä digitaalisten lähetysten puheenkoodaustapa, jonka kehityksestä ja standardoinnista vastaa *International Telecommunication Union* (*ITU*). CELP-algoritmi perustuu vekrorikvantisointiin ja koodikirjan käyttöön (Chen, 1992, Ylitalo, 1997). Ensimmäiset korkealaatuiseen puheenkoodaukseen pystyvät koodekit käyttivät korkeaa bittitaajuutta, kuten 64 kbit/s tai 32 kbit/s. Nykyisin CELP:n eri puheenkoodaustapoja ovat muun muassa matalaviiveinen versio *Low Delay Code Excited Linear Prediction (LD-CELP*), joka digitoi puhetta 16 kbit/s sekä 8 kbit/s digitoiva CS-ACELP (ITU, 2003). LD-CELP on kehitetty ratkaisuksi ongelmaan, joka syntyy algoritmin tutkiessa dataa. Tällöin syntyy aina viivettä, joka hidastaa koodausta. Viive taas voi joidenkin sovellusten kannalta muodostua liian pitkäksi, jolloin sovelluksen reaaliaikainen käyttö ei ole enää mahdollista.

3.4 GSM-KOODAUS

Global System for Mobile Communication (*GSM*) on matkapuhelinverkossa käytettävä puheen digitaalinen koodaussysteemi. Yleisesti GSM-järjestelmän kehittäminen alkoi 1980-luvun alkupuolella, kun CEPT (*Conference of European Posts and Telegraphs*) perusti vuonna 1982 ryhmän, jonka tehtävänä oli suunnitella yleiseurooppalainen matkapuhelinjärjestelmä (Sunila, 1997). Vuonna 1989 vastuu GSM:n kehittämisestä siirtyi ETSI:lle (*European Telecommunication Standards Institute*), ja ensimmäiset spesifikaatiot julkaistiin vuonna 1990 (ETSI, 2003). Kokonaisuudessaan GSM sisältää yhteensä viisi erilaista puheen

koodaustapaa: *täyden taajuuden* (*full rate, FR*), *puolitaajuuden* (*half rate, HR*) ja *parannetun täyden taajuuden* (*enhanced full rate, EFR*) ovat kolme ensimmäistä ja vanhinta koodausstandardia. Kaksi uusinta standardia ovat *adaptive multi-rate (AMR*) ja *adaptive multi-rate wideband (AMR-WB)* (Hillebrand, 2002). Näistä neljä ensimmäistä on standardoitu vastaavassa järjestyksessä GSM 06.10, GSM 06.20, GSM 06.60 ja GSM 06.90.

Täyden taajuuden GSM –koodaus FR standardoitiin 1989. Standardi mahdollistaa 22,8 kbit/s bittitaajuuden, mutta koodauksessa käytetään 13 kbit/s bittitaajuutta. Jäljelle jäävää 9,8 kbit/s aluetta käytetään virheiden estämiseen (Besacier, 2000).

Puolitaajuuden GSM-koodaus HR kehitettiin vastaamaan lisääntyneisiin käyttäjämääriin laskemalla bittitaajuutta ja standardoitiin 1995 (Hillebrand, 2002). Tällöin samalla kaistanleveydellä pystyttiin palvelemaan kaksinkertainen määrä käyttäjiä verrattuna täyden taajuuden GSM–koodaukseen. Standardi mahdollistaa 11,4 kbit/s bittitaajuuden, mutta koodauksessa käytetään 5,6 kbit/s bittitaajuutta. Jäljelle jäävä 5,8 kbit/s käytetään virheiden estämiseen. Puheenlaadussa puolen taajuuden GSM–koodaus yltää vastaavaan laatuun kuin täyden taajuuden GSM-koodauskin lähes kaikissa olosuhteissa.

Kolmas GSM–standardi, parannettu täyden taajuuden GSM–koodaus EFR, standardoitiin 1996. Se tarjoaa huomattavan parannuksen puheen laatuun täyden taajuuden GSM–koodaukseen verrattuna (Besacier, 2000). Standardin mahdollistama 22,8 kbit/s bittitaajuus jakautuu siten, että 12,2 kbit/s käytetään puheen koodaukseen ja loput 10,6 kbit/s virheiden estämiseen.

Kaksi uusinta GSM-koodausmenetelmää AMR ja AMR-WB standardoitiin 1999 ja 2001. Yksi merkittävimmistä eroista verrattuna aiempiin koodausmenetelmiin on siinä, että nämä menetelmät mahdollistavat useita bittitaajuuksia, joista voidaan valita käytettävä taajuus tilanteen mukaan (Hillebrand, 2002).

Kaikki GSM-koodekit perustuvat lineaarisen ennustavaan koodaukseen LPC (Hillebrand, 2002). Neljän vanhimman koodekin näytteenottotaajuus on 8kHz ja uusimmassa AMR-WB:ssä se on 16kHz. Koodekit toimivat äänitaajuuksilla 50Hz –

3,4kHz. Uusin AMR-WB edustaa edistyneempää tekniikkaa tältäkin osin, sillä se toimii aina 7kHz saakka. Tämä parantaa koodauksen äänenlaatua säilyttämällä äänen luonnollisuuden paremmin verrattuna muihin GSM-koodekkeihin.

3.5 MENETELMIEN VERTAILUA

Edellä kuvattuja koodausmenetelmiä ja niiden tehokkuutta ja tuottamaa puheenlaatua on esitelty taulukossa 3. Vertailusta havaitaan, että logaritminen PCM tuottaa erinomaisen puheenlaadun, mutta samalla joudutaan käyttämään suuria bittinopeuksia. Sen vuoksi PCM:ää käytetään yleisesti puhelinverkoissa, joissa puheenlaatu on ensisijaisen tärkeää. Muilla koodausmenetelmillä ei saavuteta yhtä hyvää puheenlaatua, mutta menetelmillä on muita etuja. Esimerkiksi LPC on käytössä puhetta syntetisoitaessa, koska tällöin puheen sisältämä informaatio on tärkeämpää kuin hyvä puheen laatu. Pienen bittinopeuden ansioista puhe voidaan siirtää nopeasti ja pienemmillä resursseilla.

Taulukko 3. Puheenkoodausmenetelmien vertailua (Kettunen, 2003).

Menetelmä	Bittinopeus	Laatu
tasavälinen PCM	64 kbit/s	tyydyttävä
logaritminen PCM	64 kbit/s	erinomainen
APCM	56 kbit/s	hyvä
DPCM	48 kbit/s	hyvä
ADPCM	32 kbit/s	hyvä
LPC	2,4 – 16 kbit/s	huono
CELP	4,8 kbit/s	hyvä

Taulukossa 3 esitettyjen koodausmenetelmien lisäksi on matkapuhelinympäristöissä laajalti käytössä GSM-koodaus. Tutkimuksen mukaan (Phythian, 1997) GSM sopii paremmin puheen koodaukseen kuin LPC tai CELP. GSM-koodaus ei heikennä signaalin ominaisuuksia, kuten huipputasoa tai aaltoaluetta, yhtä merkittävästi kuin LPC ja CELP.

4 PUHUJANTUNNISTUS

Matkapuhelinten räjähdysmäinen yleistyminen viimeisen reilun kymmenen vuoden aikana on nostanut puheteknologian tärkeäksi osaksi tietotekniikkaa ja samalla yhä enemmän tutkimus- ja kehitysresursseja on suunnattu puheteknologian saralle. Puheentunnistus, puhujantunnistus ja puhesynteesi (puheen tuottaminen tietokoneella) eivät kuitenkaan ole yksinkertaisia toimenpiteitä puheen luonteesta johtuen, jolloin analysoitavasta puhesignaalista muodostuu vaihteleva ja monimuotoinen (Koskenniemi, 2001). Äänisignaali sisältää normalisoidun ja kirjoitetun kielen ilmaukset ei-triviaalilla tavalla. Äänteet, niiden kestot ja taajuusjakaumat eivät ole juuri koskaan riippumattomia niitä ympäröivästä sanasta tai lauseyhteydestä.

Ongelmia tuottaa saman puhujan eri tuotosten välille syntyvät erot, jotka ovat melko satunnaisia ja johtuvat puhujan fyysisestä tai psyykkisestä tilasta. Myös eri puhujien väliset huomattavat erot, joissa voidaan osittain havaita systemaattisuutta, aiheuttavat ongelmia. Jokainen ihminen puhuu omalla tavallaan ja lisäksi esimerkiksi murteet ja painotus vaihtelevat eri ihmisten välillä hyvin paljon. Näiden lisäksi on ollut välttämätöntä kehittää menetelmiä, joilla puheen kannalta ulkopuoliset tai käytettävästä tekniikasta johtuvat häiriötekijät saadaan erotettua varsinaisesta puheesta (Neumeyer, 1995). Taustamelu ja äänisignaalin sisältämä kohina tai särö ovat turhaa tietoa, joka kulkevat äänisignaalissa. Tämä turha tieto puhujantunnistussovelluksen on osattava erottaa varsinaisesta puheesta.

4.1 PUHUJANTUNNISTUSSOVELLUKSET

Puhujantunnistustekniikkaa käytetään monien eri alojen sovelluksissa. Sovellukset voidaan jakaa käyttötarkoituksen mukaan viiteen pääryhmään (Kinnunen, 2003):

- henkilön tunnistaminen
- oikeustiede/rikostutkimus
- puheentunnistus
- usean puhujan ympäristöt

- puhekäyttöliittymät

Henkilön tai sovelluksen käyttäjän tunnistaminen on yksi luonnollisimmista puhujantunnistuksen käyttökohteista. Puhujantunnistusta voidaan käyttää korvaamaan tavanomaista käyttäjätunnukseen ja salasanaan perustuvaa tunnistusta.

Oikeustiede on tärkeä ala, jolla nykyisin käytetään puhujantunnistusta. Rikosten selvittämisessä puhenäytteitä voidaan käyttää todisteena ja tunnistaa henkilö sen perusteella. Poliisi pitää yllä puherekisteriä, johon kerätään rikollisten puhenäytteitä aivan vastaavasti kuten sormenjälkinäytteitäkin.

Puheentunnistuksessa, aivan kuten puhujantunnistuksessa, yksi ongelma on puhujan puheen variaatiot eri näytteiden välillä. Puhujantunnistuksessa käytettäviä tekniikoita voidaan käyttää puheentunnistuksessa juuri näytteiden variaatioiden huomioimisessa. Puheentunnistussovellukset voivat puhujan tunnistettuaan valita käytettävän mallin juuri kyseiselle henkilölle sopivaksi.

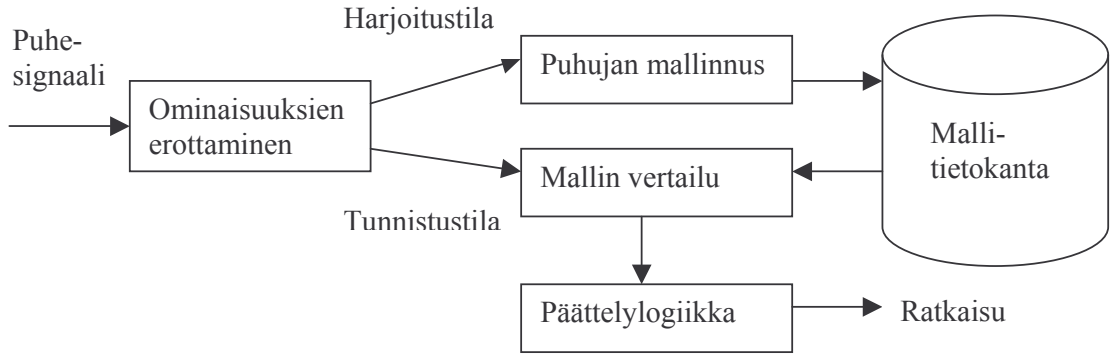
Sovelluksissa, jotka on tehty usean puhujan käytettäviksi, on puhujan tunnistuksella tärkeä rooli. Esimerkkejä ovat vaikkapa videoneuvottelut, paneelikeskustelut ja oikeuden istunnot. Sovellukset havaitsevat puhujat, seuraavat heitä ja heidän puhetta.

Puhujantunnistusta voidaan käyttää myös automaattisesti personoimaan sovelluksen käyttöliittymää tai toimintaa käyttäjän yksilöllisten mieltymysten mukaan. Samoin sovellusta itseään voidaan käyttää äänikomentojen avulla.

4.2 PUHUJANTUNNISTUSPROSESSI

Puhujantunnistusprosessi sisältää kuvassa 8 kuvatut komponentit. Käsiteltävästä digitaalisesti puhesignaalista erotetaan sitä kuvaavia numeerisia arvoja, jotka yhdessä kuvaavat äänisignaalia riittävän tarkasti, jotta puhujantunnistus on mahdollista. Puhujantunnistusjärjestelmä toimii tavallisesti kahdessa eri tilassa.

Harjoitustilassa kunkin puhujan puheesta tallennetaan tietokantaan puhujamalli, johon järjestelmä tunnistustilassa toimiessaan käsiteltävää puhesignaalia vertaa. Järjestelmään toteutetun logiikan avulla tunnistustilassa saadaan lopuksi tunnistuksen tulos.



Kuva 8. Puhujantunnistusprosessin komponentit (Kinnunen, 2003).

Puhujantunnistus voidaan jakaa kahteen erilliseen tehtävään (Kinnunen, 2003): *puhujan identifiointiin* (*speaker identificatio*) ja *puhujan varmentamiseen* (*speaker verification*). Puhujan identifioinnissa tuntematonta puhujaa verrataan tietokantaan, joka sisältää N:n tunnetun puhujan näytteet. Parhaiten tuntematonta puhujaa vastaava tietokannan puhuja palautetaan tunnistettuna henkilönä (*1:N matching)*. Puhujan varmentamisessa varmistetaan se, että puhuja todellakin on henkilö, joksi hän itseään väittää. Tällöin ääninäytettä verrataan tallennettuun näytteeseen ja jos näytteiden samankaltaisuus on riittävä, on puhuja varmennettu oikeaksi henkilöksi (*1:1 matching*). Näistä kahdesta tehtävästä puhujan identifiointi on vaativampi tehtävä varsinkin, jos tietokantaan tallennetun vertailumateriaalin määrä on suuri. Tällöin myös virheellisen identifioinnin mahdollisuus kasvaa.

Puhujan identifiointi voidaan jakaa edelleen kahteen eriluonteiseen tehtävään (Kinnunen, 2003): *avoimen joukon* (*open-set*) ja *suljetun joukon* (*closed-set*) identifiointiin. Identifiointi on avoimen joukon ongelma silloin, kun puhuja ei välttämättä ole tunnettu. Suljetun joukon identifioinnissa puhuja on jokin tiedossa olevan joukon henkilö. Näistä ongelmista avoimen joukon identifiointi on selvästi vaativampi.

Puhujantunnistusta on mahdollista tarkastella puhujan lisäksi myös puhutun puheen suhteen. Tällöin puhujantunnistus voidaan jakaa *tekstiriippuvaiseen* (*text-dependent*) ja *tekstiriippumattomaan* (*text-independent*) tehtävään. Tekstiriippuvaisessa tehtävässä on etukäteen tiedossa mitä puhuja sanoo. Tekstiriippumattomassa tehtävässä tunnistaja ei tiedä, mitä puhuja aikoo sanoa. Näistä tehtävistä tekstiriippuvainen tunnistus tuottaa tarkemman tuloksen.

4.3 PUHUJANTUNNISTUKSEN EDUT JA HAITAT

Puhujantunnistuksen suurin etu on sen luonnollisuus. Puhuminen on ihmisten pääasiallinen kommunikointimuoto, joten puhujan kannalta sovelluksen käyttäminen on äärimmäisen helppoa. Myös sovellusten kannalta puhujantunnistuksen käyttö ei ole kovin erikoinen tai vaativa tehtävä. Mitään erikoisia välineitä ei tarvita, ainoastaan mikrofoni puheen talteen ottamista varten. Samoin osa käytettävistä algoritmitmeista ovat vähän resursseja, kuten aikaa tai muistia, vaativia. Lisäksi oikeissa olosuhteissa puhujantunnistus on tehokas ja tarkka tunnistusmenetelmä (Kinnunen, 2003). Käytettäessä puhujantunnistusta osana tunnistusprosessia, vaikkapa sormenjälkitunnistuksen kanssa, paranee tunnistustarkkuus verrattuna pelkkään sormenjälkitunnistukseen. Puhujantunnistus onkin kolmanneksi yleisin biometrinen tunnistusmenetelmä kasvojen- ja sormenjälkitunnistuksen jälkeen.

Puhujantunnistuksen suurin ongelma on sen epäluotettavuus. Ihmisen puhe ei ole yhtä yksilöivä piirre kuin vaikkapa sormenjälki. Näiden ero johtuu siitä, että sormenjälki on fyysinen ominaisuus, joka saadaan suoraan mitattua ihmisen kehosta. Puhe sen sijaan on monen ruumiinosan yhteistyön tulos, jolloin puhe vaihtelee kerrasta toiseen riippuen olosuhteista. Niin fyysinen kuin henkinenkin tila vaikuttaa ihmisen tuottamaan puheeseen. Samoin ikääntyminen saa aikaan ihmisen tuottamassa äänessä muutoksia (taulukko 4). Kahta täysin samanlaista ääninäytettä ihmisen onkin käytännössä mahdotonta tuottaa.

Toinen ongelma puhujantunnistuksessa on puheen imitoinnin mahdollisuus. Aivan niin suuri ongelma se ei kuitenkaan ole, kuin voisi nopeasti ajatellen luulla, sillä

vaikka imitointi kuulostaakin meistä samanlaiselta kuin alkuperäisen puhujan puhe, niin se on vain meidän subjektiivinen mielipide. Imitoitua puhetta tarkemmin tarkasteltaessa voidaan huomata mahdolliset erot alkuperäisen ja imitoidun puheen välillä. Kolmas virhetilanteita aiheuttava ryhmä on tekniset virheet eli puhujasta riippumattomat syyt, joita aiheuttavat ympäristö ja laitteet.

Taulukko 4. Virheitä ja ongelmia aiheuttavat tekijät puhujantunnistuksessa (Campbell, 1997, Kinnunen, 2003).

Ihmisestä johtuvat	**Ympäristöstä johtuvat**	**Laitteista johtuvat**
Väärin lukeminen tai lausuminen	Taustamelu	Mikrofonin aiheuttama särö
Äärimmäisten tunteiden vaikutus ääneen	Ympäristön akustiikka	A/D muunnoksen aiheuttama kohina
Sairaus	Kaiut	Koodauksen aiheuttama laadun huononeminen
Ikääntyminen		

4.4 PUHEENTUNNISTUS

Puhujantunnistukseen liittyy läheisesti myös puheentunnistus. Luonnollisen kielen käyttäminen yhtenä käyttöliittymävaihtoehtona on tuonut uuden vaihtoehdon komentorivin ja graafisen käyttöliittymän joukkoon. Viime vuosina luonnollisen kielen käyttäminen graafisen käyttöliittymän sijaan tai osana sitä on yleistynyt. Luonnollisen kielen käyttäminen graafisen käyttöliittymän sijaan on perusteltua silloin, kun (Burstein, 1992, Koskenniemi, 2001):

- ei ole mahdollista käyttää näppäimistöä, hiirtä eikä riittävänkokoista näyttölaitetta ole saatavilla tai
- vuorovaikutus sovelluksen kanssa on monimutkaisempaa, kuin mihin valikot luontevasti soveltuvat

Tapauksista ensimmäinen johtaa puhekäyttöliittymän käyttöön ja jälkimmäinen toteutuu, kun sovellus ei muistuta yksinkertaista nappeja painamalla ohjattavaa laitetta, vaan vaatii useampia vuorosanoja toiminnon suorittamiseksi.

Viime vuosina onkin ilmaantunut laajalti käyttökelpoisia puheentunnistusohjelmia, jotka korvaavat graafisen käyttöliittymän ja mahdollistavat vaihtoehtoisen tavan käyttää sovellusta. Puheentunnistussovelluksen tyypin ja käytettävän tekniikan ja algoritmin määräävät sovelluksen käyttökohde. Jokaisesta puheentunnistussovelluksesta voidaan erottaa kolme ominaisuutta, jotka kuvaavat sovelluksen luonnetta ja käyttötarkoitusta (Spanias, 1991).
Puheentunnistussovellus voi:

- olla tarkoitettu ainoastaan tietylle puhujalle tai useammalle puhujalle
- tunnistaa jatkuvaa puhetta tai yksittäisiä sanoja
- käsittää laajan sanajoukon tai suppean sanajoukon

Tunnistettavien henkilöiden määrän kertovasta ominaisuudesta käytetään termiä *tunnistusmuoto* (*recognition mode*). Tunnistettavan puheen muodon ilmaisevasta ominaisuudesta käytetään termiä *syötemuoto* (*input mode*). Puheentunnistussovelluksen lopullisen tehokkuuden ja tarkkuuden määrittelee kolmas ominaisuus eli *sanajoukko* (*vocabulary*). Luonnollisesti sanajoukon koko vaikuttaa tunnistuksen tehokkuuteen, mutta myös sanajoukon luonne voi vaikeuttaa tunnistusta. Esimerkiksi puhuttujen numeroiden tunnistus onnistuu melko hyvin, mutta jos tunnistuskohteena ovatkin puhutut aakkoset, tunnistustarkkuus laskee selvästi. Tämä johtuu sanajoukon kasvusta, mutta erityisesti siitä, että monet kirjaimet, joita numeroissa ei esiinny, ovat hyvin samankaltaisia (b,c,d,g).

Näiden ominaisuuksien avulla voidaan kuvata tämän hetkinen tekninen osaaminen puheentunnistuksessa. Puheentunnistussovellus voi joko

- tunnistaa suppean komentosanajoukon puhujasta riippumattomasti tai
- tunnistaa laajan sanajoukon yksittäiseltä puhujalta, jonka puheelle järjestelmä on automaattisesti viritetty puhenäytteiden avulla

Kumpaakaan yllä mainituista tavoitteista ei saavuteta täysin virheettömästi. Kukin puheentunnistussovellus tunnistaa jonkin todennäköisyyden mukaan puhutut sanat oikein. Tämä todennäköisyys on kuitenkin riittävän hyvä sovellusten käyttöä ajatellen. Viimeisen kymmenen vuoden aikana puheentunnistusmenetelmät ovat kehittyneet tasaisesti (taulukko 5) ja puheentunnistuksen kannalta yhä vaikeampia tehtäviä pystytään ratkaisemaan. Tekniikan kehittymisen vuoksi eri tehtävissä tapahtuvien virheiden määrä pienenee vuosittain keskimäärin noin 15% (Woodland, 1998).

Taulukko 5. Puheentunnistuksen kehittyminen (Woodland, 1998).

Tehtävä	Vuosi	Virheprosentti
Selkeä sanomalehtimateriaali, pieni joukko sanoja	1993	3% - 5%
Selkeä sanomalehtimateriaali, suuri joukko sanoja	1994	5% - 8%
Taustameluinen sanomalehtimateriaali, suuri joukko sanoja	1995	10% - 15%
TV- tai radiolähetys	1997	13% - 20%
Puhelinkeskustelu	1998	25% - 40%

Puhujasta riippuvat (*speaker dependent)* puheentunnistussovellukset on optimoitu tunnistamaan tietyn henkilön puhetta. Muiden henkilöiden puhetta sovellus tunnistaa heikosti tai ei lainkaan. *Puhujasta riippumattomat* (*speaker independent)* puheentunnistussovellukset on suunniteltu tunnistamaan usean eri henkilön puhetta. Puhujasta riippumaton puheentunnistus on vaikeampaa kuin vain tietyn henkilön puheen tunnistus. Samoin jatkuvan puheen tunnistaminen verrattuna yksittäisten sanojen tunnistamiseen on vaikeampaa.

Puheentunnistussovellukset toteutetaan tapauskohtaisesti sovelluksen käyttötarkoitusta silmällä pitäen. Näin sovelluksen on mahdollista käyttää erilaisia tapoja tunnistuksen helpottamiseksi ja onnistumiseksi (Woodland, 1998). Sovelluksissa voidaan rajoittaa sanaston kokoa tai rajoittaa puheen muotoa esimerkiksi vain yksittäisiin sanoihin. Eri puhujien määrää voidaan rajoittaa,

kenties sallitaan vain yksi puhuja, jolle sovellus on optimoitu. Sovelluksen käyttöä voidaan rajoittaa sallimalla vain tietynlaiset mikrofonit, jolloin tekniikasta johtuvia häiriötekijöitä saadaan rajattua. Taustamelun tai kohinan määrälle voidaan asettaa jokin raja, jonka ylityttyä ei edes yritetä puhetta tunnistaa, vaan puhujaa pyydetään toistamaan sanansa. Nämä seikat vaikuttavat keskeisesti puheentunnistuksen tehokkuuteen ja tarkkuuteen (Spanias, 1991). Tavoite kuitenkin on, että kaikista rajoitteista päästäisiin eroon. Olipa tunnistusalgoritmi kuinka hyvä tahansa, niissä ei voida ottaa huomioon inhimillistä, ihmisestä lähtöisin olevaa virhettä. Tällaisia virheitä ovat esimerkiksi väärin luettu tai sanottu sana. Nämä virheet rajoittavat tunnistuksen tehokkuutta (Campbell, 1997).

Puheentunnistussovelluksen käytössä on eksplisiittisesti määritelty kielimalli, jonka sisältämiä ilmauksia äänisignaalista pitäisi tunnistaa. Puhesignaalin tietty osa, esimerkiksi tauosta taukoon, pitäisi liittää johonkin kielimallissa määriteltyyn sanaan. Tarkoituksena on siis löytää se sovelluksen sillä hetkellä odottama sana, joka vastaa riittävän hyvin vastaanotettua signaalin osaa. Vertailun suorittamiseen on olemassa erilaisia menetelmiä, niin loogisia kuin matemaattisiakin.

Vapaamman puheen tunnistuksessa käytetään yleisesti menetelmää, jossa puhesignaalista pyritään tunnistamaan äänteitä tai *difoneja*, eli kahden vierekkäisen äänteen yhdistelmiä. Difoneille ja niiden eri yhdistelmille on etukäteen laadittu sanakirja esimerkiksi äärellisen automaatin muotoisena tilasiirtymäverkkona. Verkon koko vaihtelee riippuen käyttötarkoituksesta ollen komentosanoille suppeampi ja hyvin valikoiva ja vastaavasti vapaalle saneluohjelmalle laaja ja avoin.

Osa menetelmistä ei yritä tunnistaa puhesignaalista äänteitä tai difoneja, eli kahden vierekkäisen äänteen yhdistelmiä, vaan käsittelevät signaaleja kokonaisvaltaisemmin tai käyttävät signaalin venytystä tai tiivistystä siten, että vertailu kielimallin sanoihin onnistuu helpommin ja tarkemmin.

Puheentunnistusprosessin alkuvaiheissa äänisignaalin nopeita paineenvaihteluita kuvaavia kohtia pyritään tiivistämään, jolloin äänteitä hyvin erotteleva tieto saadaan selville. Fourier-muunnoksen avulla ääninäytteen yhden jakson

värähtelykäyrä muunnetaan taajuusjakautumaksi, josta on helpompi erottaa äänteitä kuin värähtelykäyrästä. Muun muassa eri vokaaleilla on keskinäisiltä suhteiltaan tunnusomaiset resonanssitaajuudet eli *formantit*, jotka nähdään helpommin taajuusjakautumasta. Itse puheentunnistusprosessi voidaan sovittaa tämän jälkeen Markovin piilomalliin, jos saadut signaalinkäsittelyn tulokset pakotetaan johonkin laajaan äänteiden aakkostoon eli *koodikirjaan*.

5 TUNNISTUSTARKKUUDEN TESTAUS

Tutkimuksen lähtökohtana on ohjelmallisesti mitata kuinka eri koodausmenetelmät vaikuttavat puhujantunnistuksen tarkkuuteen. Tarkoitus on vertailla kahta eri koodekkia ja niiden eroja tunnistustarkkuudessa. Lisäksi koodaamattoman ja koodatun puheen eroja on tarkoitus tutkia myös erilaisin kuvaajin ja kuuntelemalla puhujien puhenäytteitä samalla havaintoja kirjaten.

5.1 KÄYTETTY PUHEAINEISTO JA TUTKITTAVAT KOODEKIT

Puhujien näytteet sisältäväksi tietokannaksi on valittu englanninkielinen TIMIT (LDC, 2004), jota on käytetty aiemminkin automaattista puhujantunnistusta käsittelevissä tutkimuksissa (Lamel, 1993). Se sisältää yhteensä 6300 englanninkielistä virkettä ja 630 eri puhujaa Yhdysvalloista 8 eri kielialueelta (LDC, 2004). Puhujista 70% on miehiä ja 30% naisia. Myös muita korpuksia on kirjallisuuden perusteella käytetty mutta hyvin vaihtelevasti. TIMIT on nauhoitettu kohinattomassa ympäristössä, jolloin muiden tekijöiden, kuten esimerkiksi mikrofonin, vaikutus ei vaikuta tuloksiin. Lisäksi tutkimuksessa ei ole tarkoitus keskittyä korpuksiin vaan koodauksen vaikutukseen puhujantunnistuksessa, joten korpuksen valinta ei ole keskeinen seikka.

Koodekeiksi on valittu taulukon 6 mukaiset GSM 06.10 ja G.726, joka on ITU:n standardoima ADPCM-menetelmää käyttävä koodekki. Nämä koodekit ovat olleet useissa automaattista puhujantunnistusta käsittelevissä tutkimuksissa mukana (Hirsch, 2002, Kuitert, 1997, Phythian, 1997).

Taulukko 6. Testauksessa käytettävät koodekit ja niiden parametrit.

Koodekki	Bittitaajuus kbit/s
PCM (ei koodausta)	16 / 8
GSM 06.10	8
G.726	8

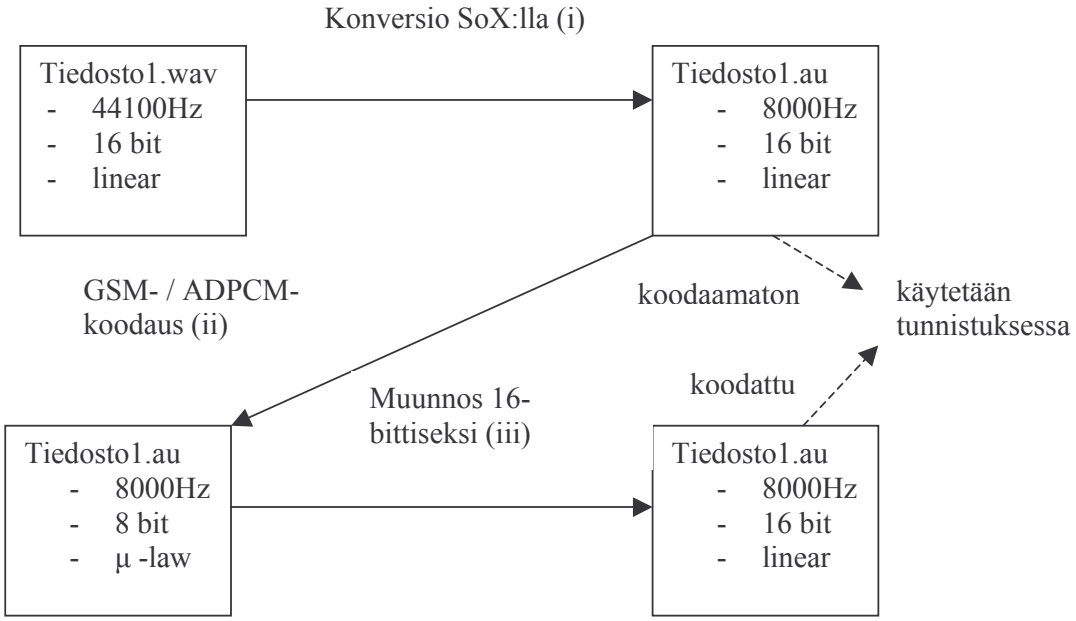
Sen mukaan, ovatko puhenäytteet tallennettu opetus- tai testaustilassa koodaamattomana vai koodattuna, jakautuvat testitapaukset kahteen ryhmään: 1) testiolosuhteet ovat samanlaiset opetus- ja testaustilassa (*matched conditions*) ja 2) testiolosuhteet eroavat toisistaan (*mismatched conditions*). Näin ollen eri testitapauksille saadaan taulukossa 7 esitetyt kombinaatiot. Testiolosuhteiden täsmätessä (tapaus 1) tunnistustarkkuus pitäisi olla parempi kuin tapauksessa 2, jossa testiolosuhteet poikkeavat toisistaan. Kirjallisuuden perusteella useita koodekkeja käsitteleviä tutkimuksia ei ole kovinkaan montaa tehty.

Taulukko 7. Testitapausten kombinaatiot kullekin koodekille.

Testiolosuhde opetustilan ja tunnistustilan välillä	**Puhenäyte opetustilassa**	**Puhenäyte tunnistustilassa**
Täsmäävät	Koodattu	Koodattu
	Ei koodausta	Ei koodausta
Eivät täsmää	Koodattu	Ei koodausta
	Ei koodausta	Koodattu

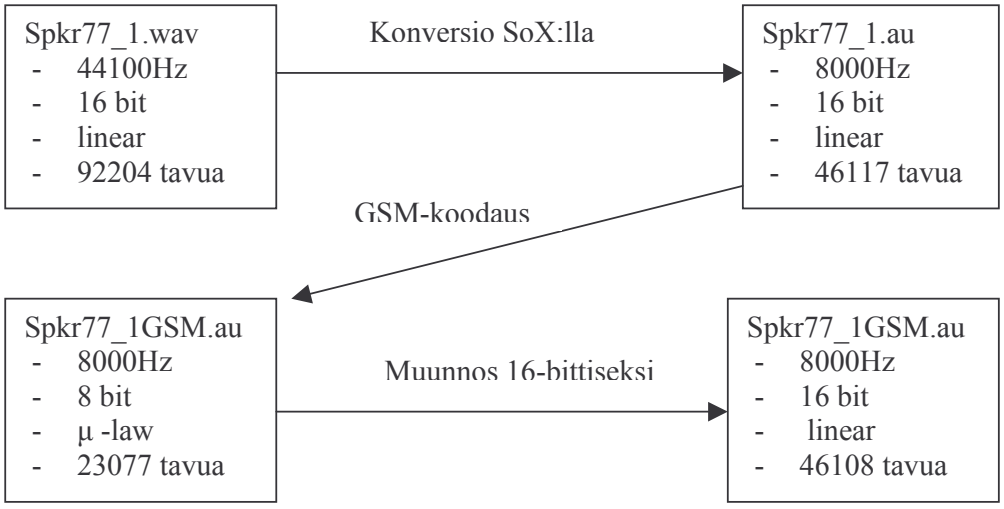
5.2 TIEDOSTOJEN KÄSITTELY JA KÄYTETTY TUNNISTUSOHJELMA

Testiaineisto sisältää kultakin puhujalta joukon lukupuhuntaa, jossa jokaisen alkuperäisen äänitteen tallennusmuotona on käytetty 44,1 kHz näytteenottotaajuista ja 16-bittistä lineaarista WAV-äänitiedostoa. Kukin äänitiedosto käsiteltiin kuvan 9 mukaisesti ensin Sox:lla (i) siten, että näytteenottotaajuudeksi tuli 8000 Hz ja tiedostoformaatiksi Unix-käyttöjärjestelmän käyttämä au-formaatti. Muunnos tehtiin sen vuoksi, että käytetyt koodekit vaativat tiedostoformaatiltaan au-tiedostoja. Tämän jälkeen äänitiedosto koodattiin halutulla koodekilla (ii). Koodauksen jälkeen äänitiedosto oli 8-bittinen, μ-law –algoritmilla koodattu, joten se muunnettiin takaisin lineaariseksi 16-bittiseksi (iii) Matlab-ohjelmalla, jotta äänitiedostot olisivat yhteneväisiä kaikissa testitilanteissa.



Kuva 9. Äänitiedostojen käsittely ennen tunnistusta tutkimuksen 1. vaiheessa.

Esimerkiksi puhujan numero 77 äänitiedoston käsittely ja äänitiedoston ominaisuudet eri vaiheissa etenivät kuvan 10 mukaisesti käytettäessä GSM-koodekkia.



Kuva 10. Esimerkki äänitiedoston käsittelystä ja ominaisuuksista eri vaiheissa tutkimuksen 1. vaiheessa.

Puhujantunnistusohjelmaksi valittiin Joensuun yliopiston kehittämä *Sprofiler* (Karpov, 2003). Ohjelman asetuksiin tai sen toimintaan ei tehty mitään muutoksia tutkimuksen alussa eikä sen aikana. Liitteestä 1 käy ilmi käytetty konfiguraatiotiedosto, jota ohjelma käytti piirteiden irrottamista varten. Ohjelma laskee äänitiedostoista *akustiset piirteet* eli *mel-kepstrikertoimet* (MFCC) ja muodostaa äänitiedostoista puhujakohtaiset piirrevektorit ja mallit vektorikvantisoinnilla (VQ). Tunnistettaessa puhujaa, ohjelma laskee näistä piirrearvoista etäisyysarvot puhujien välillä, vertailee niitä tunnistettavan puhujan malliin ja tunnistaa puhujaksi sen, joka eroaa tuntemattomasta puhujasta vähiten.

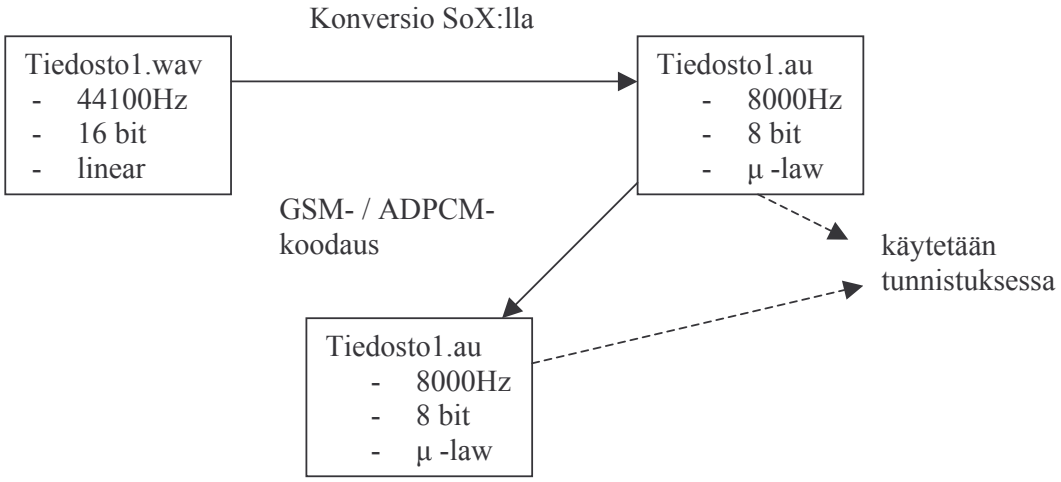
Tutkimuksessa hyödynnettiin kaikki TIMIT:n 630 puhujaa, joille jokaiselle laskettiin piirrevektorit. Yksittäiseen testitapaukseen osallistuvia puhujia ei rajattu otoksella mitenkään, vaan mukana olivat kaikki puhujat. Sprofiler laski automaattisesti tunnistustarkkuuden kaavalla k/n, missä k=oikein tunnistettujen määrä ja n=puhujien kokonaismäärä.

5.3 RESOLUUTION JA KOODAUSALGORITMIN VAIKUTUS TULOKSIIN

Taulukkoon 8 on merkitty eri testitapaukset ja niiden tunnistustarkkuudet. Tuloksista huomataan, että kaikissa täsmäävissä tapauksissa tunnistustarkkuudet ovat hyviä. Sen sijaan ei-täsmäävissä tapauksissa tunnistustarkkuus putoaa rajusti. Lisäksi näissä ei-täsmäävissä tapauksissa vaikuttaa siltä, että käytettäessä opetusvaiheessa koodattua materiaalia ja tunnistusvaiheessa koodaamatonta materiaalia, on tunnistustarkkuus parempi kuin jos opetusvaiheessa käytettäisiin koodaamatonta materiaalia ja tunnistusvaiheessa koodattua materiaalia.

Tunnistustarkkuuden suuren putoamisen vuoksi tutkittiin myös bittimäärän eli resoluution vaikutusta tunnistustarkkuuteen. Tunnistustarkkuus 8 bitin resoluutiolla PCM datalla putosi 96%:sta noin 90%:iin (taulukko 8), joka oli samaa tasoa kuin 16-bittisellä GSM-koodatulla datalla saatu tulos. Tarkempi koodattujen äänitiedostojen tarkastelu paljasti myös toisen eroavaisuuden verrattuna koodaamattomaan materiaaliin. Koodattu äänitiedosto oli μ–law –algoritmilla koodattua eikä suinkaan enää lineaarista. Vaikka muunnos takaisin lineaariseksi

16-bittiseksi dataksi tehtiinkin, tunnistustarkkuus jäi huomattavan alhaiseksi. Tämän vuoksi hyödyllisimmät tulokset koodekkien vaikutuksesta tunnistustarkkuuteen ja koodekkien välisistä eroista tunnistustarkkuudessa saatiin tutkimuksen 2. vaiheessa, kun myös koodamattomat äänitiedostot olivat 8 bittisiä μ-law -muotoisia eikä muunnoksia enää koodauksen jälkeen suoritettu (kuva 11).



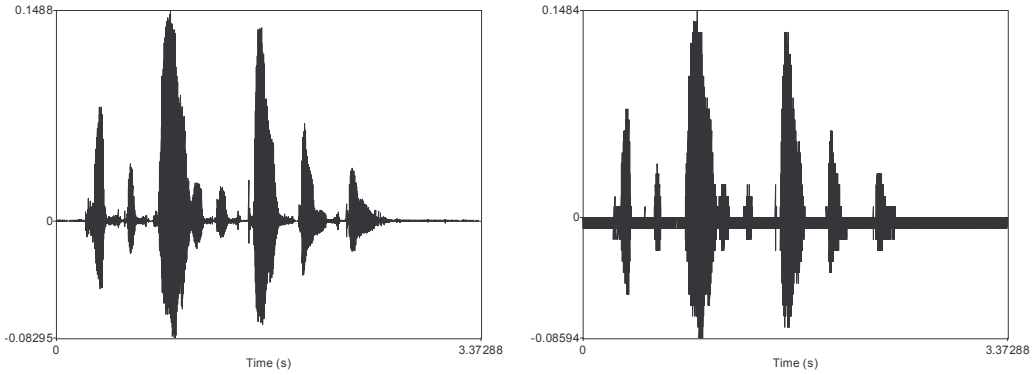
Kuva 11. Äänitiedostojen käsittely ennen tunnistusta 2. vaiheessa.

Saatujen tulosten perusteella näyttää siltä, että ADPCM-koodattu materiaali on lähes yhtä hyvää tunnistuksen kannalta kuin alkuperäinen koodaamaton PCM-data. Tulos on odotettu, sillä ADPCM on yksi PCM-standardin muunnoksista. Sen sijaan GSM-koodatulla materiaalilla tunnistustarkkuus putoaa noin puoleen verrattuna alkuperäisen koodaamattoman tai ADPCM-koodatun datan tunnistustarkkuuteen.

Taulukko 8. Koodekkien tunnistustarkkuus.

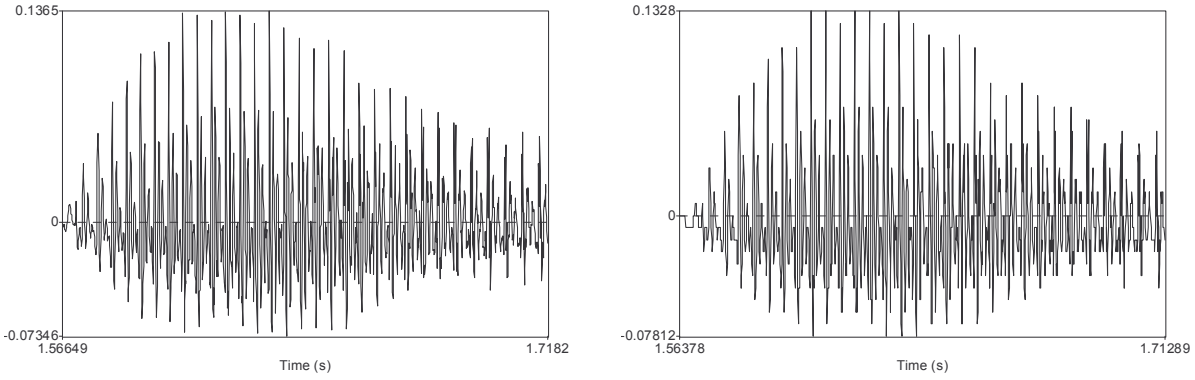
	Tutkittu koodekki	**Koodaus opetustilassa**	**Koodaus tunnistustilassa**	**Tunnistus-tarkkuus**
1	PCM 16-bit, linear	PCM	PCM	96,03%
2	GSM 06.10 16-bit, linear	GSM 06.10	GSM 06.10	91,11%
3		GSM 06.10	PCM	19,52%
4		PCM	GSM 06.10	6,67%
5	PCM 8-bit, linear	PCM	PCM	89,68%
6	GSM 06.10 8-bit, linear	GSM 06.10	GSM 06.10	79,36%
7		GSM 06.10	PCM	6,51%
8		PCM	GSM 06.10	4,44%
9	PCM 8-bit, μ –law	PCM	PCM	95,71%
10	GSM 06.10 8-bit, μ-law	GSM 06.10	GSM 06.10	86,98%
11		GSM 06.10	PCM	46,83%
12		PCM	GSM 06.10	35,87%
13	G.726, 8-bit, μ-law	G.726	G.726	94,92%
14		G.726	PCM	95,40%
15		PCM	G.726	94,76%

Tunnistustarkkuus putosi huomattavasti ei-täsmäävissä olosuhteissa silloin, kun koodattu materiaali oli muunnettu takaisin lineaariseen muotoon koodauksen jälkeen (taulukko 8). Kuitenkaan koodaamattoman ja koodatun äänitiedoston välillä ei havaittu merkittäviä kuuntelemalla havaittavissa olevia eroja. Myös äänitiedostoista ohjelmallisesti piirretyt kuvaajat vaikuttavat hyvin samankaltaisilta, vaikkakin kvantisoinnin vaikutus on havaittavissa koodatun äänitiedoston kuvaajassa. Kuvassa 12 on vertailtu lineaarisen PCM-muotoisen äänitiedoston ja lineaarisen GSM-koodatun äänitiedoston eroja.

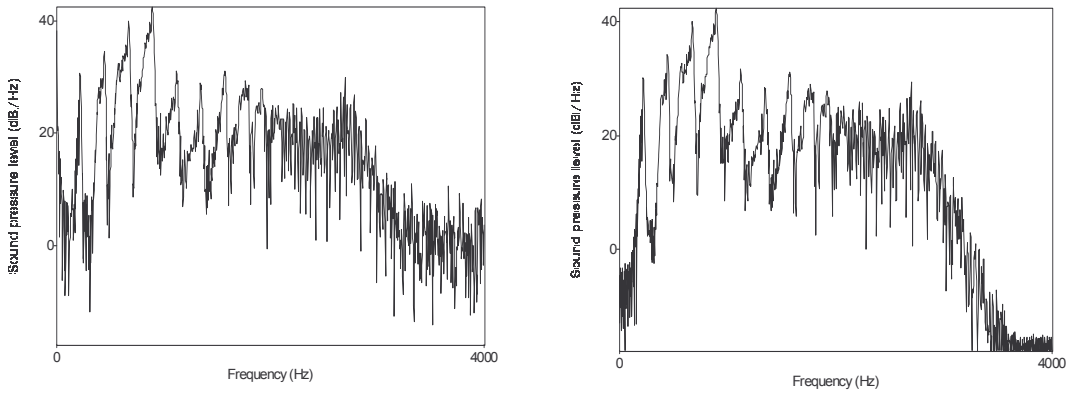


Kuva 12. Spkr300_10.au äänitiedosto PCM muodossa vasemmalla ja GSM-koodattuna oikealla.

Kuvissa 13 ja 14 on samasta äänitiedostosta rajattu pieni pätkä, jotta voidaan vertailla eroja yhden äänteen kohdalla. Kuten kuvaajista havaitaan, koodaus ei muuta äänitiedostoja kovin havaittavasti. Selkein havaittavissa oleva ero on siinä, että GSM-koodaus hävittää informaatiota niin matalista kuin korkeistakin taajuuksista. Tämä on tyypillistä GSM-koodekeille ja samalla järkevää, sillä matalat ja korkeat taajuudet eivät ole puheen ymmärtämisen kannalta oleellisia. Äänitiedostoja ja niiden ominaisuuksia vertailtiin Praat ohjelmalla (Boersma, 2004).



Kuva 13. Yksi äänne puheesta. PCM vasemmalla ja GSM-koodattu oikealla.



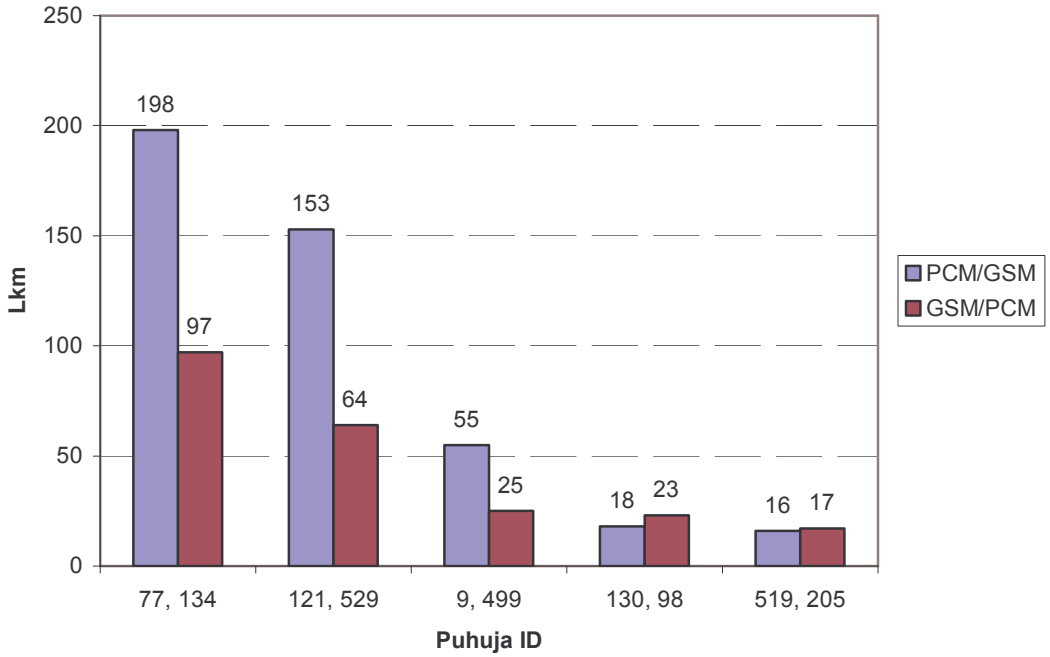
Kuva 14. Sama äänne puheesta (dB/Hz). Vasemmalla PCM, oikealla GSM-koodattu.

5.4 MAGNEETTIPUHUJAEFEKTI

Johtuen tunnistustarkkuuden rajusta pudotuksista ei-täsmäävissä tapauksissa tutkittiin hieman tarkemmin kuinka puhujantunnistus GSM-koodekin tapauksessa käyttäytyi. Tutkittaessa virheellisesti tunnistettuja puhujia havaittiin varsinkin tapauksessa, jossa opetustilassa käytettiin koodaamatonta dataa ja tunnistusvaiheessa GSM-koodattua dataa, että muutama puhuja nousi niin sanotuksi *magneettipuhujaksi*, joksi tunnistusohjelma muitakin puhujia virheellisesti tunnisti. Puhujaa kutsutaan magneettipuhujaksi, jos tunnistusohjelma veikkaa useita tuntemattomia puhujia kyseiseksi puhujaksi selvästi muita puhujia useammin.

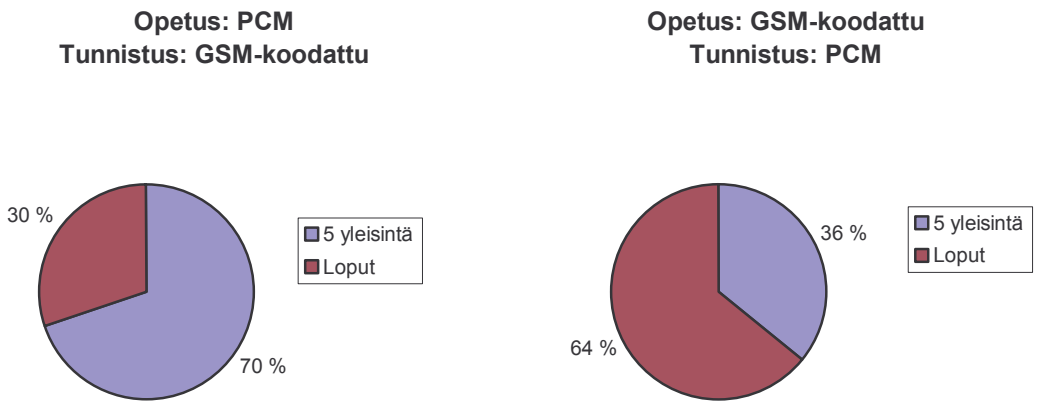
Kuvassa 15 on esimerkkitapaus testistä, jossa opetus tapahtui koodaamattomalla PCM-datalla ja tunnistus GSM-koodatulla datalla tai päinvastoin. Kuvassa lukumäärä ilmaisee kuinka useasti kyseinen puhuja valittiin tunnistetuksi. PCM/GSM tarkoittaa, että opetus tapahtui PCM-koodatulle ja tunnistus GSM-koodatulle materiaalille. GSM/PCM tarkoittaa päinvastaista tapausta. Yleisimpien magneettipuhujien ja heidän puheensa kuuntelemalla tehty vertailu ei paljastanut yhtäläisyyksiä, jotka voisivat selittää miksi juuri kyseiset puhujat olivat magneettipuhujia.

Ensimmäisessä testitapauksessa (PCM/GSM) puhuja numero 77 on nainen, muut neljä miehiä. Toisessa testitapauksessa (GSM/PCM) puhujat 499 ja 98 ovat naisia, muut kolme miehiä. Niin ikään puheen muissa kuultavissa olevissa ominaisuuksissa, kuten puhenopeudessa, tauoissa ja hiljaisuuden osuuksissa ei ole samankaltaisuuksia havaittavissa näiden puhujien osalta. Osa puhujista puhuu nopeasti, osa normaalilla nopeudella ja osa hyvinkin hitaasti. Koko puhujamateriaaliin suhteutettuna testitapauksessa, jossa opetus tapahtui koodaamattomalla materiaalilla ja tunnistus GSM-koodatulla materiaalilla viiden yleisimmän magneettipuhujan osuus nousi 70%:iin, kun loppujen 18 osuudeksi jäi vain 30%. Lukumääräisesti laskettuna nämä viisi yleisintä puhujaa toteutui 440 tunnistustapauksessa.



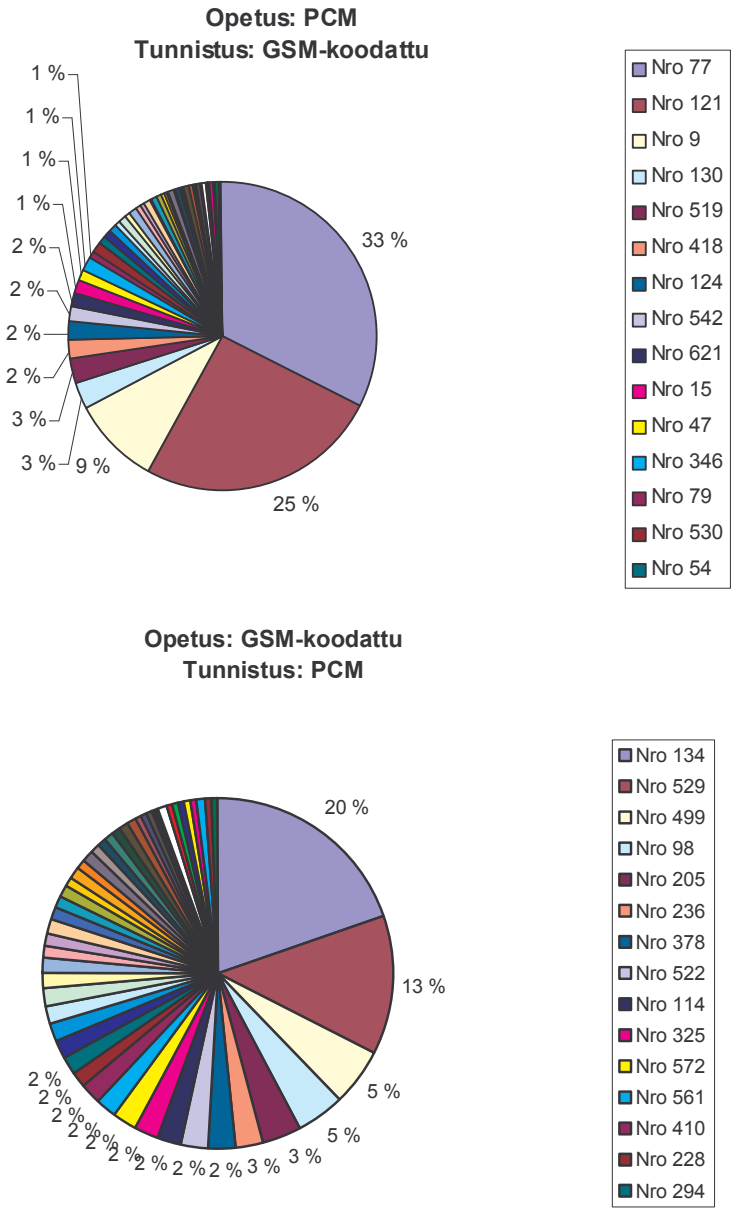
Kuva 15. Yleisimmät niin sanotut magneettipuhujat, joiksi virheellisesti tunnistettu.

Kuvassa 15 on esitetty myös testitapaus jossa opetus tapahtui GSM-koodatulla materiaalilla ja tunnistus koodaamattomalla materiaalilla. Magneettipuhujaefekti esiintyi myös tässä tapauksessa, muttei niin selvästi kuin edellä esitetyssä tilanteessa. Nyt viiden yleisimmän magneettipuhujan puhujan osuus oli 36% ja loppujen 149 osuudeksi tuli 64% (kuva 16). Lukumääräisesti laskettuna tässä tapauksessa viisi yleisintä puhujaa toteutui 226 tunnistustapauksessa.



Kuva 16. Viiden yleisimmän magneettipuhujan osuus kaikista puhujista.

Näistä testitapauksista huomataan, että mikäli opetus tapahtui GSM-koodatulla materiaalilla ja tunnistus koodaamattomalla materiaalilla, jakautuu tunnistusohjelman ehdottamat puhujat selvästi useamman puhujan kohdalle. Lukumääräisesti laskettuna tunnistusohjelma ehdotti testitapauksessa, jossa opetus tapahtui koodaamattomalla datalla ja tunnistus GSM-koodatulla datalla, 74 eri puhujaa 630:sta mahdollisesta. Testitapauksessa, jossa opetus tapahtui GSM-koodatulla materiaalilla ja tunnistus koodaamattomalla datalla, jakautui tunnistusohjelman ehdottamat puhujat 153 eri puhujan kesken. Kuvassa 17 on esitetty yleisimpien magneettipuhujien jakaumat ei-täsmäävissä tilanteissa.



Kuva 17. Yleisimmät magneettipuhujat ei-täsmäävissä olosuhteissa.

Normaalin testiajon lisäksi tunnistustarkkuutta tutkittiin tapauksilla, joissa materiaalia sekoitettiin niin, että muutamien puhujien äänitiedostot korvattiin opetustilassa GSM-koodatulla materiaalilla kaikkien muiden puhujien äänitiedostojen ollessa PCM-dataa. Taulukosta 9 nähdään, että yksikin GSM-koodattu puhuja toimi magneetin tavoin tunnistustilanteessa niin, että jokaisen puhujan kohdalla tunnistusohjelma ehdotti puhujaksi tätä opetustilassa GSM-

koodattua puhujaa. Tällaisesta puhujasta, joka pystyy ”imitoimaan” muita puhujia, käytetään myös termiä *susipuhuja* (Doddington, 1998). Tässä tapauksessa näyttää siltä, että GSM-koodaus tekee puhujasta susipuhujan tai ainakin on selkeä tekijä, joka vaikuttaa susipuhujan syntymiseen.

Tilanne käännettiin myös toisinpäin ja opetustilanteessa muutamien puhujien äänitiedostot korvattiin PCM-datalla kaikkien muiden puhujien äänitiedostojen ollessa GSM-koodattua dataa. Myös näissä tapauksissa näyttää siltä, että koodaus on se tekijä, joka vaikuttaa selkeästi susipuhujan syntymiseen.

Taulukko 9. Testitapaukset, joissa aineistoa sekoitettu opetustilassa.

Testitapaus	Koodaus opetustilassa	Koodaus tunnistus-tilassa	Tunnistustulos
Lähtötilanne, 16-bit, linear	PCM	GSM	Tunnistuivat 74 eri puhujaksi
1	PCM, paitsi puhuja nro 77 GSM-koodattu	GSM	Kaikki tunnistuivat puhujaksi nro 77
2	PCM, paitsi puhuja nro 121 GSM-koodattu	GSM	Kaikki tunnistuivat puhujaksi nro 121
3	PCM, paitsi puhuja nro 4 GSM-koodattu	GSM	Kaikki tunnistuivat puhujaksi nro 4
4	PCM, paitsi puhujat nro 258 ja 547 GSM-koodattu	GSM	Kaikki tunnistuivat joko puhujaksi nro 258 (70,32%) tai nro 547 (29,68%)
Lähtötilanne 16-bit, linear	GSM	PCM	Tunnistuivat 154 eri puhujaksi
5	GSM, paitsi puhuja nro 77 PCM	PCM	96,51% tunnistui puhujaksi nro 77
6	GSM, paitsi puhuja nro 121 PCM	PCM	80,32% tunnistui puhujaksi nro 121
7	GSM, paitsi puhuja nro 4 PCM	PCM	Kaikki tunnistuivat puhujaksi nro 4
8	GSM, paitsi puhujat nro 258 ja 547 GSM-koodattu	PCM	Kaikki tunnistuivat joko puhujaksi nro 258 (70,48%) tai nro 547 (29,52%)
Lähtötilanne, 8-bit, μ –law	PCM	GSM	Tunnistuivat 269 eri puhujaksi
9	PCM, paitsi puhuja nro 77 GSM-koodattu	GSM	24,13% tunnistui puhujaksi nro 77
10	PCM, paitsi puhuja nro 121 GSM-koodattu	GSM	18,73% tunnistui puhujaksi nro 121
11	PCM, paitsi puhuja nro 4 GSM-koodattu	GSM	35,40% tunnistui puhujaksi nro 4
12	PCM, paitsi puhujat nro 258 ja 547 GSM-koodattu	GSM	55,08% tunnistui joko puhujaksi nro 258 (33,17%) tai nro 547 (21,91%)
Lähtötilanne, 8-bit, μ –law	GSM	PCM	Tunnistuivat 326 eri puhujaksi
13	GSM, paitsi puhuja nro 77 PCM	PCM	20,00% tunnistui puhujaksi nro 77
14	GSM, paitsi puhuja nro 121 PCM	PCM	3,33% tunnistui puhujaksi nro 121
15	GSM, patisi puhuja nro 4 PCM	PCM	22,54% tunnistui puhujaksi nro 4
16	GSM, paitsi puhujat nro 258 ja 547 GSM-koodattu	PCM	45,87% tunnistui joko puhujaksi nro 258 (25,08%) tai nro 547 (20,79%)

6 YHTEENVETO

Äänenkoodaukseen on kehitetty erilaisia koodausmenetelmiä, joita käytetään myös puheenkoodaukseen. Kaikkien koodausmenetelmien tavoite on vähentää tarvittavien resurssien määrää ilman, että puheen laatu kärsisi liikaa. Koodaus ei saa hävittää puheesta liikaa informaatiota, jotta puhuja voidaan tunnistaa luotettavasti ääninäytteen perusteella. Informaatiota täytyy kuitenkin hävittää esimerkiksi sen vuoksi, että puhe siirretään verkon yli, jolloin resursseja ei välttämättä ole käytössä kovin runsaasti. Tapauskohtaisesti onkin selvitettävä mikä koodekki tilanteeseen kulloinkin parhaiten sopii ja kuinka käytettävissä olevat resurssi- ja laatuvaatimukset saadaan täytettyä.

Tutkimuksessa pyrittiin selvittämään koodauksen vaikutusta puhujantunnistukseen. Tutkimuksen puhujatietokannaksi valittiin TIMIT. Testeissä käytettiin PCM-, GSM- ja ADPCM-koodattua dataa. PCM- ja ADPCM-datalla tunnistustarkkuus oli noin 95% ja tunnistustulokset olivat keskenään kutakuinkin yhteneväiset niin täsmäävissä kuin ei-täsmäävissäkin olosuhteissa. Tämä olikin odotettavaa, sillä molemmat menetelmät perustuvat PCM-koodaukseen.

Sen sijaan GSM-koodatulla datalla tunnistustarkkuus putosi edellisistä. Täsmäävissä olosuhteissa tunnistustarkkuus laski noin 8 prosenttiyksikköä 87%:iin, mutta ei-täsmäävissä olosuhteissa tarkkuus romahti verrattuna PCM- ja ADPCM-koodattuun dataan ollen 40%:n molemmin puolin. GSM-koodaus tehtiin FR-koodekilla, joka on ensimmäinen standardoitu GSM-koodekki. GSM-koodekki edustaa hyvin yleistä, matkapuhelimissa käytettävää koodausmenetelmää, jonka perustana on LPC. Mielenkiintoista olisi tietää millainen tunnistustarkkuus saavutettaisiin, jos koodaus tehtäisiin käyttäen GSM-standardin uusimpia AMR tai AMR-WB –koodekkeja.

Testimateriaalia sekoitettaessa tutkimuksessa havaittiin koodauksen vaikutus magneettipuhujaefektiin. Saatujen tulosten perusteella vaikuttaa siltä, että GSM-koodaus tekee joistain puhujista magneetti- eli susipuhujia, jotka vetävät

tunnistustilanteessa muita tunnistettavia puhujia itseensä ja heikentävät näin tunnistustarkkuutta.

Matkapuhelinten määrän yhä kasvaessa ja GSM-koodauksen yleistyessä paranee epäilemättä myös koodauksen taso. Jo nyt puhe ja henkilön tunnistaminen puheen perusteella on yksi biotunnistuksen osa. Jatkossa myös puhelimen välityksellä siirretty puhe ja sen käyttäminen yhtenä biotunnistuksen menetelmänä on varteenotettava vaihtoehto.

VIITELUETTELO

Besacier, L., Grassi, S., Dufaux, A., Ansorge, M., Pellandini, F. (2000) GSM Speech Coding And Speaker Recognition. *IEEE International Conference on* Acoustics, Speech, and Signal Processing *ICASSP '00, volume* **2**, (5-9 June), 1085 -1088.

Boersma, P., Weenink, D. (2004) *Praat:doing phonetics by computer*. WWW-sivusto, http://www.fon.hum.uva.nl/praat/ (15.11.2004).

Burstein, A., Stölze, A., Brodersen, R.,W. (1992) *Using Speech Recognition in a Personal Communications System*. EECS Department, University of California at Berkeley, CA, USA.

Campbell, J., P. (1997) Speaker Recognition: A Tutorial*. Proceedings of the IEEE*, *volume* **2**, No. 9. September 1997.

Chen, J-H., Cox, R., V., Lin, Y-C., Jayant, N., Melchner, M., J. (1992) A Low-Delay CELP Coder for the CCITT 16kb/s Speech Coding Standard. *IEEE Journal on Selected Areas in Communications* **10 (**5).

Doddington, G., Liggen, W., Martin, A., Przybocki, M., Reynolds, D. (1998) *SHEEP, GOATS, LAMBS and WOLVES.* A Statistical Analysis of Speaker Performance in the NIST 1998 Speaker Recognition Evaluation.

ETSI (2003) *The European Telecommunications Standards Institute*. WWW-sivusto, http://www.etsi.org/ (15.12.2003).

Gold, B., Morgan, N. (2000) *Speech and audio signal processing : processing and perception of speech and music*. John Wiley & Sons, Inc., New York.

Hanhijärvi, J. (1998) *Tutkielma DAB:stä… eli radiosta josta voi nähdä.* Multimediatekniikka, Tietoliikenneohjelmistojen ja multimedian laboratorio, Teknillinen korkeakoulu.

Hillebrand, F. (toim.) (2002) *GSM and UMTS : the creation of global mobile communication*. John Wiley & Sons, West Sussex.

Hirsch, H-G. (2002) The Influence of Speech Coding on Recognition Performance in Telecommunication Networks. *7th International Conference on Spoken Language Processing - ICSLP'02*, Denver, USA, syyskuu 2002.

Holm, J-M (1998) *Audioformaatit*. Ohjelmistotekniikan seminaari, Tietotekniikan Cum Laude-harjoitustyö, Jyväskylän yliopisto. WWW-sivusto, http://www.cc.jyu.fi/~hojakr/audio/audioformats.html (13.11.2003).

ITU (2003) *International Telecommunication Institution*. WWW-sivusto, http://www.itu.int/home/index.html (15.12.2003).

Karpov, E. (2003) *Sprofiler manual*. Tietojenkäsittelytieteen laitos, Joensuun yliopisto.

Kettunen, M. (2003) *Puheenkoodaus*. Seminaarityö, Digitaalisen signaalinkäsittelyn erikoiskurssi, Sähkötekniikan osasto, Lappeenrannan teknillinen yliopisto.

Kinnunen, T. (2003) *Spectral Features for Automatic Text-Independent Speaker Recognition.* Lisensiaattityö, Department of Computer Science, University of Joensuu.

Koskenniemi, K. (2001) *Johdatus kieliteknologiaan*. Kieliteknologian opetusmonisteita, nro 1, Helsingin yliopiston yleisen kielitieteen laitos. WWW-sivusto, http://www.ling.helsinki.fi/kit/2001s/ctl190/kt-johd-2001s/ (15.12.2003).

Kuitert, M., Boves, L. (1997) Speaker verification with GSM Coded Telephone Speech. *Proc. Eurospeech'97* **2**, 975-978.

Lamel, L., Gauvain, J (1993) Cross-Lingual Experiments with Phone Recognition. *Acoustics, Speech, and Signal Processing* **2**, 507-510.

LDC, Linguistic Data Consortium (2004) *Linguistic Data Consortium*. WWW-sivusto, http://www.ldc.upenn.edu/. (22.3.2004).

MPEG Home Page (2003) *MPEG Home Page*. WWW-sivusto, http://www.chiariglione.org/mpeg/index.htm. (30.10.2003).

Neumeyer, L., Weintraub, M. (1995) *Robust Speech Recognition in Noise Using Adaptation and Mapping Techniques*. SRI International, Speech Technology and Research Laboratory, CA, USA.

Painter, T., Spanias, A. (2000) Perceptual Coding of Digital Audio. *Proceedings of the IEEE* **88** (4).

Phythian, M., Ingram, J., Sridharan, S. (1997) Effects of speech coding on text-dependent speaker recognition. *IEEE Region 10 Annual Conference. Speech and Image. Technologies for Computing and Telecommunications* **1**, 137 –140.

Riederer, C., A., J. (2004) Mullistava MPEG-4. *Hifi 1/2004* s.56-57.

Spanias, A.,S., Wu, F., H. (1991) Speech coding and speech recognition technologies: a review. *IEEE International Symposium* **1**, 572 – 577.

Sunila, H., (1997) *Mobiiliin tiedonsiirtoon liittyvät standardit*. WWW-sivusto, http://www.tml.hut.fi/Studies/Tik-110.300/1997/Mobile/mobile_std_2.html (15.12.2003).

Vaarala, S. (1998) *MP3 Audiopakkaus*. Ti 3, Teknillinen korkeakoulu (30.10.2003).

Watkinson, J. (2000) *MPEG-2*. Focal Press, Oxford.

Woodland, P. (1998) *Speech Recognition*. The Institution of Electrical Engineers. Printed and published by the IEE, London, UK.

Ylitalo, J. (1997) *Langaton audio- ja visuaalitekniikka*. WWW-sivusto, http://www.tml.hut.fi/Studies/Tik-110.300/1997/Mobile/audiovisual_2.html (15.12.2003).

LIITE 1: SPEAKER PROFILER CONFIGURATION FILE

```
; Speaker profiler configuration file
; by Evgeny Karpov, 2003

; General parameters
[General]

action         = FE

input file     = timit_feature_vectors.ts
input type     = LIST

database       = timit.db

; Logging parameters
[Logging]
log events     = Yes
type           = FILE
file name      = timit_feature_vectors.log
file mode      = NEW

; Feature extraction parameters
[Feature extraction]
preprocess     = Yes
feature type   = MFCC
window size    = 30
window shift   = 20

; Preprocessing parameters
[Preprocess]
silence removal = No
dc removal     = Yes
high emphasis filtering = Yes
; high emphasis coefficient = 0.97

; Silence removal parameters
[Silence removal]
algorithm      = KINN
threshold      = 0.2

; Preprocessing before matching
[Matching preprocess]
prequantization  = No
online clustering = No
speaker pruning  = No
pruning type     = ADAPTIVE
```

```
; MFCC parameters
[MFCC]
order          = 12
mel filters    = auto
window function = Hamming
spectrum type  = Magnitude

; Modeling parameters
[Modeling]
model type     = VQ
index model set = No

; VQ params
[VQ]
size           = 64
index codebook  = Yes
matching func  = MSE
```