AUTOMAATTISEN PUHUJANVARMENNUKSEN PÄÄTÖSLOGIIKKA

Teemu Kilpeläinen

17.6.2002

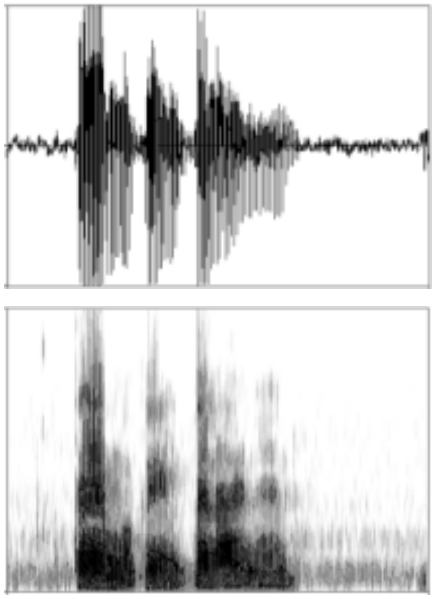
Joensuun yliopisto
Tietojenkäsittelytieteen laitos
Pro gradu –tutkielma

ESIPUHE JA KIITOKSET

Aloitin tämän pro gradu -tutkielman valmistelun ajatustasolla jo keväällä 2000, työskennellessäni ensimmäistä kertaa Joensuun yliopiston Tietojenkäsittelytieteen laitoksen kesätyöläisenä professori Pasi Fräntin johtamassa automaattista puhujantunnistusta tutkivassa projektissa. Kuluneiden kahden vuoden aikana olen osallistunut projektiin tekemällä maisterin tutkintooni kuuluvan erikoistyön ja kandidaatin tutkielman kyseisen projektin aihepiiriin liittyen. Varsinaisen pro gradun kirjoitustyön aloitin syyskuussa 2001 ja työ jatkui vaihtelevalla nopeudella yli kylmän talven ja aurinkoisen kevään 2002. Nyt kesällä 2002 kaikki on vihdoin valmista ja on kiitosten aika.

Tämän tutkielman valmistuminen ja filosofian maisterin tutkintoni saaminen olisi ollut pidempi prosessi ilman useiden henkilöiden myötävaikutusta. Erityisesti tahdon kiittää innostamisesta ja työnohjauksesta tutkija Tomi Kinnusta ja professori Pasi Fräntiä. Kiitokset assistentti Ismo Kärkkäiselle kärsivällisyydestä erikoistyöhöni ja graduun liittyneiden ohjelmakoodien ongelmien selvittämistyössä.

Lämpimät kiitokset myös kotiväelleni taustatuesta –ilman teitä tämä kaikki olisi ollut paljon vaikeampaa.



Joensuussa 17.6.2002 Teemu Kilpeläinen

TIIVISTELMÄ

Automaattinen puhujanvarmentaminen on yksi biometrisistä tunnistusmenetelmistä. Toimiva päätöslogiikka on kiinteä osa korkean luotettavuuden ja käytettävyyden omaavaa automaattista puhujanvarmennusjärjestelmää. Tässä tutkielmassa käydään ensin läpi automaattiseen puhujanvarmennukseen, ihmisen puheen tuottoon ja puheen havaitsemiseen liittyviä periaatteita. Lisäksi tutkielmassa keskitytään kuvaamaan päätöslogiikkaan liittyvä matemaattinen perusta, kynnysarvon teoreettinen määrittely ja asettaminen, sekä normalisoitujen joukkojen perusteet ja kyseisten joukkojen muodostusvaihtoehtoja.

Kynnysarvojen suunnittelussa paneudutaan kynnysarvon dynaamisen päivittämisen tarpeeseen ja päivittämiseen käytettäviin menetelmiin. Lisäksi tutkielma käsittelee puhujanvarmennusjärjestelmien arviointiin liittyviä työkaluja, joita ovat esimerkiksi järjestelmän keskimääräistä virhealttiutta kuvaavat mittarit: oikeiden puhujien hylkäysvirhe (FR), huijareiden hyväksymisvirhe (FA) ja järjestelmän keskimääräinen virhealttius (EER). Näitä samoja työkaluja käytetään myös kynnysarvojen suunnitteluun. Tutkielmaan kuuluu myös kokeellinen osuus, jossa esitellään näiden arviointiin käytettävien mittareiden testaukseen kehitetty testausjärjestelmä ja testiajojen tulokset.

Avainsanat: *Automaattinen puhujanvarmennus, automaattinen puhujantunnistus, päätöslogiikka, kynnysarvo.*

TUTKIELMASSA KÄYTETTYÄ SANASTOA

Automaattinen puhujantunnistus	Puhuja tunnistetaan tietokoneen avulla, puheen sisältämien puhujaa kuvaavien piirteiden perusteella.
Automaattinen puhujanvarmennus	Puhujan antama väite henkilöllisyydestään hyväksytään tai hylätään tietokoneen avulla, puheen sisältämien puhujaa kuvaavien piirteiden perusteella.
FA, $P(\hat{X} \mid \overline{X})$	(False Acceptance) Huijaripuhujien hyväksymisvirheen todennäköisyys.
FR, $P(\hat{\overline{X}} \mid X)$	(False Rejection) Oikeiden puhujien hylkäämisvirheen todennäköisyys.
EER	(Equal Error Rate) Keskimääräinen virhealttius, jonka arvo on sama kuin FA:n ja FR:n leikkauskohdan arvo. Käytetään varmennusjärjestelmien kynnysarvojen suunnittelussa.
Kynnysarvo	(Threshold) Päätöslogiikkaan liittyvä samankaltaisuusarvo (tai eroavaisuusarvo), jonka perusteella puhuja joko hyväksytään tai hylätään.
Normalisoidut puhujajoukot	Varmennettavasta puhujasta eroavat puhujat sijoitetaan usein normalisoituihin puhujajoukkoihin. Normalisoidut puhujajoukot voivat olla: 1. vastinjoukkoja, joihin sijoitetaan varmennettavaa puhujaa riittävästi muistuttavat puhujat, tai 2. eroavia joukkoja, joihin sijoitetaan vastaavasti varmennettavasta puhujasta riittävästi eroavat puhujat.
Piirreirrotus	Puheesta erotetaan puhujan henkilökohtaisia puheominaisuuksia kuvaavia piirteitä.
Päätöslogiikka	Päätöslogiikalla tarkoitetaan sitä puhujanvarmennusjärjestelmän tarvitsemaa älykästä päättelyosaa, jonka avulla järjestelmä kykenee tekemään varmennuspäätöksen puhujan hyväksymisestä tai hylkäämisestä.

SISÄLLYSLUETTELO

1 JOHDANTO

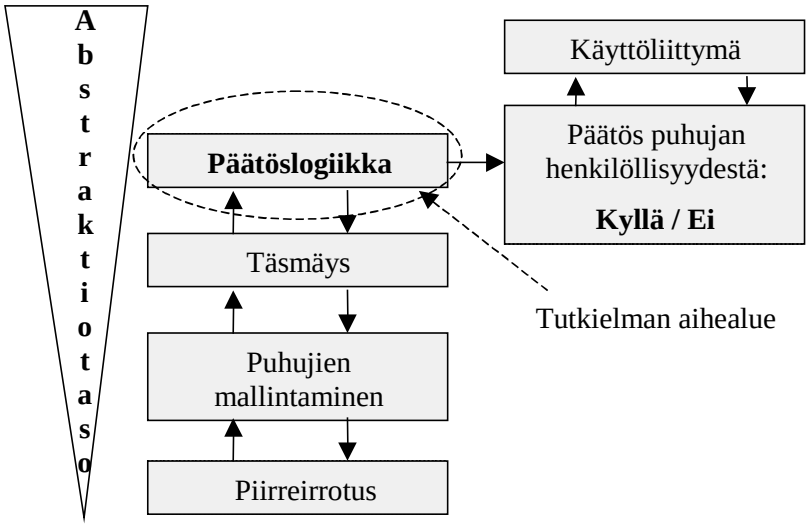
Ihmisen henkilöllisyyden varmentaminen on osa nykyihmisen arkipäivää. Henkilöllisyys voidaan varmentaa monella eri tavalla, joista yleisimpiä ovat erilaiset tunnussanat ja PIN-koodit. Näiden henkilöllisyydenvarmennusmenetelmien heikkoutena on niiden riippumattomuus tunnistettavan käyttäjän omista henkilökohtaisista ominaisuuksista, eli käyttäjän todellisesta henkilöllisyydestä [11, 23].

Tunnussanat ja PIN-koodit ovat itse asiassa ainoastaan merkkijonoja, jotka voivat helpostikin joutua ulkopuolisen käsiin ja antaa hänelle näin mahdollisuuden esiintyä tunnusluvun omistajana esimerkiksi pankkiautomaatilla tai teollisuuslaitoksen ovella. Tämä on vakava turvallisuusuhka, jonka torjumiseen tuovat apua ihmisen omiin henkilökohtaisiin ominaisuuksiin pohjautuvat *biometriset* henkilöllisyydenvarmennusmenetelmät [1, 3]. Tällaisia varmennusmenetelmiä ovat esimerkiksi sormenjälkien-, silmän rakenteen- ja puhujan henkilöllisyydenvarmentamiseen puheäänestä liittyvät menetelmät [3, 23]. Biometrisiä tunnistusmenetelmiä käytetään ovenavausjärjestelmien lisäksi apuna esimerkiksi rikostutkinnassa rikollisen tunnistamiseen [24].

Tässä tutkielmassa keskitytään puhujanvarmennusmenetelmien kuvaamiseen. Puhujan varmennukseen pohjautuvia henkilöllisyydenvarmentamismenetelmiä käytetään yleensä osana laajempaa turvallisuusjärjestelmää, jossa käytetään rinnakkain muitakin varmennusmenetelmiä, kuten tavallista metalliavainta tai sirukorttia. Tällaisilla useaan eri tunnistusmetodiin pohjautuvilla järjestelmillä saavutetaan *vahva tunnistus* [20]. Vahvassa tunnistuksessa varmennuspäätös perustuu sekä henkilöstä riippumattomaan varmennusmenetelmään että henkilöstä riippuvaan biometriseen varmennusmenetelmään.

Automaattinen puhujanvarmennus voidaan jakaa yleisellä tasolla seuraaviin osatehtäviin (kuva 1.1). Järjestelmän alimmalla abstraktiotasolla sijaitsevat puhenäytteen käsittelyyn tarvittavat signaalinkäsittelyoperaatiot, joilla puhesignaalista erotetaan puheen ominaispiirteitä. Seuraavalla tasolla tapahtuvassa puhujan mallinnuksessa järjestelmä luo piirteiden pohjalta puhujamallin, jolla mallinnetaan laskettujen piirteiden tilastollisia ominaisuuksia. Täsmäysvaiheessa järjestelmä vertailee puhujan antamaa ääninäytettä aiemmin tietokantaan tallennettuun kyseisen puhujan ääninäytteeseen. Seuraavalla tasolla ylöspäin mentäessä sijaitsee järjestelmän päätöslogiikka, jossa järjestelmä tekee päätöksen puhujan hyväksymisestä tai hylkäämi-

sestä. Ylimmällä tasolla on käyttöliittymä, jonka välityksellä käyttäjä on yhteydessä varmennusjärjestelmään.



Kuva 1.1: Puhujanvarmennusjärjestelmä yleisellä tasolla.

1.1 Tutkielman tavoitteet ja tutkimusongelman asettelu

Tässä tutkielmassa keskitytään selvittämään varmennusjärjestelmien sisältämää *päätöslogiikkaa* (desicion logic), joka on oleellinen osa toimivaa ja luotettavaa järjestelmää. Varmennusjärjestelmä tekee päätöslogiikkansa avulla päätöksen käyttäjän hyväksymisestä tai hylkäämisestä. Päätöslogiikka ja päätöslogiikan kehittäminen on yksi puhujan varmennukseen liittyvän tutkimuksen tämän päivän suurista haasteista [4]. Päätöslogiikkaan kuuluvan kynnysarvon (threshold) määrittämiseen ja päivittämiseen liittyvät asiat muodostavat suuren osan tämän tutkielman päämääristä.

Lukijalle selvitetään myös puhujanvarmennustehtävään liittyvän *piirreirrotuksen* (feature extraction) perusteet. Piirreirrotuksessa puheesta etsitään varmennustehtävän mahdollistavia piirteitä ja samalla puheesta karsitaan varmennuksessa turhaa informaatiota. Lisäksi tutkielmassa käydään läpi tarkemmin automaattiseen puhujanvarmennusjärjestelmien evaluointiin liittyviä osatekijöitä. Näitä ovat esimerkiksi seuraavat arviointimittarit kuten: järjestelmän keskimääräinen virhealttius EER (Equal Error Rate) ja ROC-käyrä (Receiver Operating Characteristics) [20].

Tutkielmassa etsitään vastausta seuraaviin kysymyksiin:

1. mitä etuja (tai haittoja) puhujakohtaisesti määritellystä kynnysarvosta on verrattuna koko puhujajoukolle asetettuun yleiseen kynnysarvoon,

2. kuinka koodikirjan koko, puhujien lukumäärä ja testidatan pituus vaikuttaa itse toteutetun puhujanvarmennusjärjestelmän varmennustarkkuuteen ja suoritusaikaan. Sekä vertaillaan saatujen tulosten järkevyyttä suhteessa muiden saamiin tutkimustuloksiin.

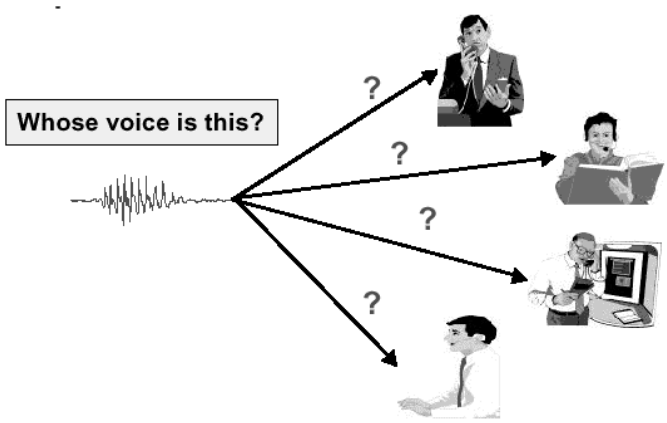
1.2 Tutkielman rakenne

Tutkielma on jäsennetty luvuittain seuraavasti. Luvussa 2 käydään läpi automaattisen puhujantunnistuksen periaatteet ja selvitetään kyseiseen tehtäväkenttään kuuluvaa yleistä termistöä. Luvussa 3 luodaan katsaus ihmisen puhe-elimiin ja puheen tuottamiseen sekä kuuloelimiin ja puheen havaitsemiseen. Luvussa 4 käydään läpi varmennustehtävään liittyvä piirreirrotus sekä puhujamallien luominen ja vertailu. Luvussa 5 käsitellään puhujanvarmennuksen arviointia käymällä läpi arviointiin tarvittavia menetelmiä. Lisäksi luvussa 5 verrataan puhujanvarmennusjärjestelmien vahvuuksia ja heikkouksia suhteessa muihin henkilönvarmennusmenetelmiin. Luvussa 6 perehdytään päätöslogiikkaan ja kynnysarvojen määrittelyyn liittyviin matemaattisiin menetelmiin sekä kynnysarvon dynaamiseen päivittämiseen. Luvussa 7 esitellään toteutetun puhujanvarmennusjärjestelmän prototyypin rakenne, testiajojen rakenne sekä testiajoissa saadut tulokset. Lisäksi luvussa 7 verrataan omissa testiajoissa saatuja tuloksia muiden tutkijoiden saamien tulosten kanssa. Lopuksi luvussa 8 luodaan yhteenveto tutkielman sisällöstä.

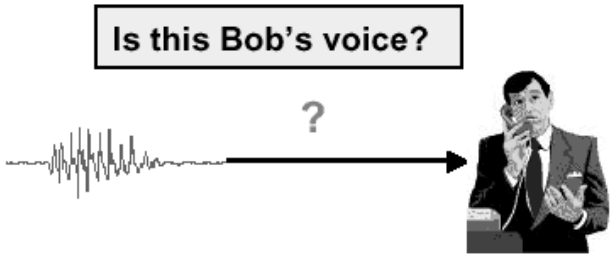
2 PUHUJANTUNNISTUKSEN PERIAATTEITA

Puhujantunnistustehtävä voidaan ajatella yksinkertaisesti prosessina, jossa tietokone tunnistaa puhujan henkilöllisyyden tämän äänen ominaispiirteiden perusteella [20]. Tässä luvussa selvitämme puhujantunnistustehtävän perusteita ja kyseiseen tehtävään liittyviä ongelmakenttiä yleisellä tasolla. Lisäksi tarkastelemme puhujanvarmennusjärjestelmien vahvuuksia ja heikkouksia suhteessa muihin henkilöntunnistusjärjestelmiin. Automaattinen puhujantunnistus voidaan jakaa tehtävänä muiden biometristen tunnistusmenetelmien tapaan kahteen osaan [3, 4, 11, 16, 19, 20, 26]:

Identifiointiin (speaker identification), jossa järjestelmän tavoitteena on erottaa puhuja rekisteröityneiden puhujien joukosta (kuva 2.1) ja *varmentamiseen* (speaker verification), jossa puhujantunnistusjärjestelmän tehtävänä on varmentaa käyttäjän antama väite henkilöllisyydestään tai hylätä väite (kuva 2.2).



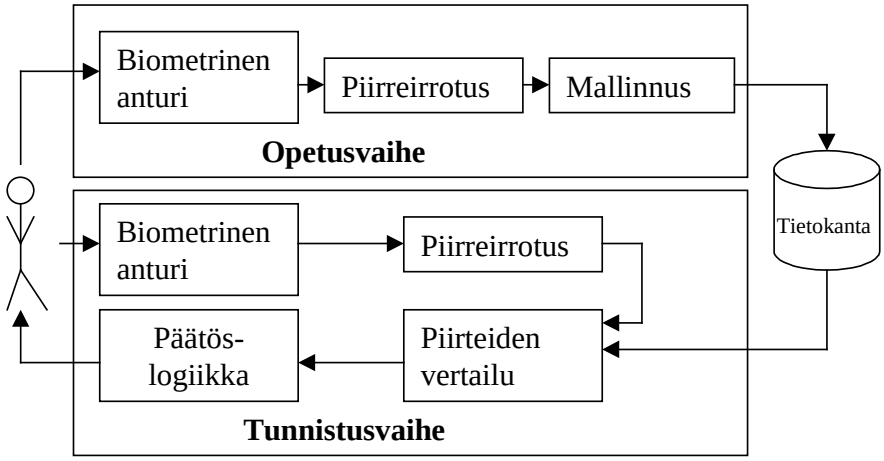
Kuva 2.1: Identifointitehtävän periaate [31].



Kuva 2.2: Puhujanvarmennuksen periaate [31].

Yleiskielessä käytetään usein termiä *puhujantunnistus* yleisnimenä automaattisille puhujanvarmennus- ja puhujantunnistamistehtäville. Tämä tutkielma keskittyy erityisesti varmennustehtävään liittyvän ongelmakentän selvittämiseen.

Kuvassa 2.3 on esitetty yleisellä tasolla biometristen tunnistusjärjestelmien rakennetta. Biometriset tunnistusmenetelmät voidaan jakaa kuvan mukaisesti kahteen loogisesti erilliseen osaan: järjestelmän käyttäjäksi rekisteröitymisen suorittavaan osaan (*opetusvaihe*) ja tunnistamisen suorittavaan osaan (*tunnistusvaihe*) [3, 16, 20]. Näitä käsitellään seuraavaksi.



Kuva 2.3: Biometrisen tunnistusjärjestelmän osat [20].

Opetusvaihe

Opetusvaiheen tarkoituksena on opettaa järjestelmälle kunkin puhujan *puhujamalli*, eli ”äänijälki”, jonka avulla kyseinen puhuja voidaan hakea myöhemmin tietokannasta [20]. Tätä puhujan tunnistamisen mahdollistavaa informaatiota kutsutaan usein puheen *piirteiksi*. Automaattisen puhujan varmennuksen osavaihetta, jossa puhenäytteestä etsitään piirteitä, kutsutaan *piirreirrotukseksi*. Piirreirrotusta varten tuotettu analoginen puhesignaali muutetaan ensin analogisesta digitaaliseksi eli tietokoneen ymmärtämään muotoon [16]. Piirreirrotuksen tuloksena saadaan joukko *piirrevektoreita*, josta muodostetaan jollakin menetelmällä *puhujamalli*. Puhujamalli tallennetaan tietokantaan.

Piirteiden tilastollisten ominaisuuksien tarkoituksena on kuvata puhujan äänisignaalin vaihteluita mahdollisimman hyvin, jotta malli olisi toimiva myös tilanteissa, joissa puhuja on vilustunut tai hänen puheäänensä on jotenkin muulla tavalla muuttunut. Puhujamallit voidaan jakaa yleisesti kahteen eri luokkaan [20]: *sapluunamalleihin* (template models) ja *stokastisiin*

malleihin (stochastic models). Sapluunamallien tarkoituksena on pyrkiä mallintamaan jotakin tiettyä lausahdusta useasta piirrevektorista muodostetun sarjan pohjalta rakennetun keskiarvosarjan perusteella [1]. Stokastisissa malleissa puheentuottamisprosessi oletetaan satunnaisprosessiksi, johon tarvittavat parametrit voidaan arvioida tarkasti jollain määritellyllä menetelmällä [3, 20]. Esimerkkejä sapluunamalleista ovat esimerkiksi DTW (Dynamic Time Warping) [4] ja VQ (Vector Quantization) [35]. Stokastisia malleja ovat esimerkiksi GMM (Gaussian Mixture Model) [30] ja HMM (Hidden Markov Model) [4].

Piirreirrotus parantaa tehokkuuden lisäksi myös puhujantunnistamistehtävän suorittavan järjestelmän tarkkuutta, koska piirreirrotus poistaa äänisignaalista tunnistuksen kannalta turhaa tietoa [16]. Piirreirrotuksessa voidaan poistaa esimerkiksi ääninäytteessä olevat tauot ja taustamelu. Opetusvaihe on usein samanlainen sekä puhujan identifiointi- että varmennustehtävissä.

Tunnistusvaihe

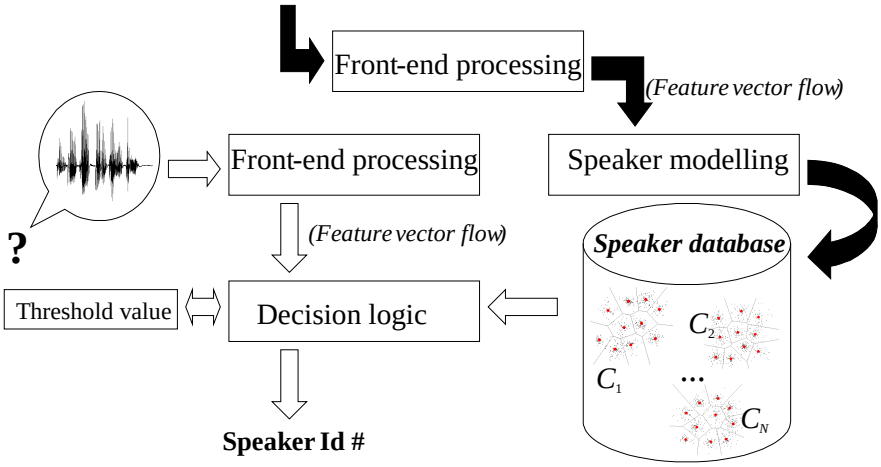
Tunnistusvaiheessa toteutetaan varsinainen puhujantunnistus. Puhuja antaa tunnistusjärjestelmälle ensin ääninäytteensä, jolle tehdään opetusvaiheen mukaisesti piirreirrotus. Piirreirrotuksen tuloksena syntynyttä puhujamallia verrataan tämän jälkeen järjestelmän tietokannassa oleviin puhujamalliin/puhujamalleihin. Tunnistustehtävä voi olla joko *avoimen joukon ongelma* (open set), jossa puhujan puhujamallia ei välttämättä ole olemassa järjestelmän tietokannassa, tai *suljetun joukon ongelma* (closed set), jolloin puhujan malli on olemassa [3, 16]. Tunnistusvaiheet eroavat toisistaan puhujan identifiointi- ja varmennustehtävässä.

2.1 Identifiointitehtävä

Jos halutaan tietää, ketä puhujatietokannan puhujaa tuntematon puhuja eniten muistuttaa, on kyseessä puhujan identifiointitehtävä. Identifiointi tehdään suljetun joukon tapauksessa laskemalla samankaltaisuus tai eroavaisuus puhujamallin sisältämien piirrevektoreiden ja kaikkien järjestelmän tietokannassa olevien puhujamallien suhteen [17]. Puhuja tunnistetaan sen mukaan, jonka puhujamallit täsmäävät parhaiten (kuva 2.4).

Avoimen joukon tapauksessa puhuja tunnistetaan, jos löydetään *kynnysarvon* (threshold) alittava eroavaisuus kyseisen puhujan ja jonkin tietokannassa olevan puhujamallin kanssa. Kyn-

nysarvo on puhujan varmennusjärjestelmän päätöslogiikkaan liittyvä samankaltaisuusarvo, jonka perusteella puhuja joko hyväksytään sisään järjestelmään tai hylätään [4, 20]. Puhuja hyväksytään, jos väitetyn puhujan puhujamalli vastaa vähintään tämän arvon verran väitettyä puhujaa. Kynnysarvon alittavista puhujamalleista valitaan kaikkein parhaiten täsmäävä.

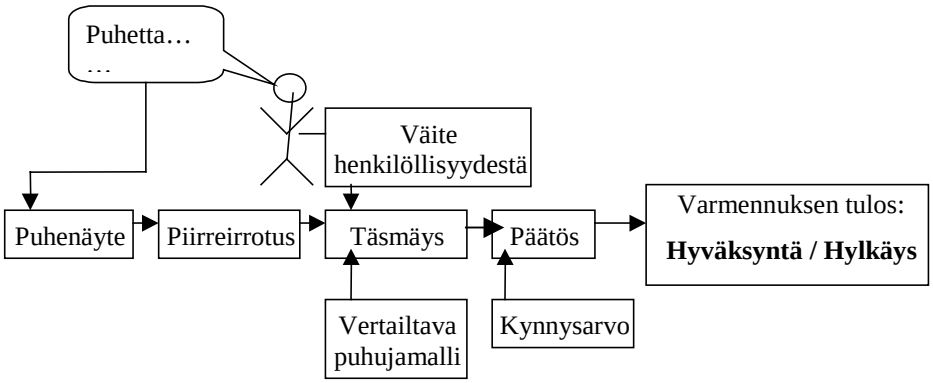


Kuva 2.4: Identifiointitehtävä kaaviokuvana.

2.2 Varmennustehtävä

Puhujanvarmennusjärjestelmän tarkoituksena on varmistaa, onko puhuja se joka hän väittää olevansa. Väite henkilöllisyydestä on joko tosi tai epätosi. Varmennustehtävä on *binäärinen luokitteluongelma*, käyttäjän antama väite henkilöllisyydestään joko hyväksytään tai hylätään (kuva 2.5) [4, 20]. Varmennustehtävässä verrataan puhujan piirrevektoreita ainoastaan väitetyn puhujan puhujamalliin, toisin kuin identifiointitehtävässä, jossa vertailu tehdään kaikkien puhujamallien välillä.

Jos vertailussa päästään kynnysarvoa (threshold) pienempään eroavuuteen, annetaan vastaus: ”*puhuja oli se, joka hän väittää olevansa*”. Muuten annetaan vastaus: ”*puhuja ei ole se, joka hän väittää olevansa*”. Jos puhuja ei ole kysytty henkilö, voi tunnistusalgoritmi näyttää kielteisen päätöksen lisäksi pisteytyksen jonka perusteella varmennus epäonnistui. Esimerkiksi: ”*löydettiin yli 2 prosentin eroavaisuus, puhuja ei ole kysytty henkilö*”. Turvallisuustehtävässä toimiva varmennusjärjestelmä voi varmennuksen hylkäämisen lisäksi ilmoittaa hylkäyksestä suoraan esimerkiksi vartiointiliikkeeseen.



Kuva 2.5: Varmennustehtävä kaaviokuvana.

2.3 Sisällöstä riippuvat ja sisällöstä riippumattomat tunnistustehtävät

Puhujanvarmennus- ja puhujan identifiointitehtävät voivat olla [3, 11, 16]:

1. *Tekstin sisällöstä riippuvia* (text-dependent) tehtäviä, joissa käyttäjä antaa kirjautumisvaiheessa puhenäytteenään joitakin opetusvaiheessa antamistaan sanoista. Tunnistukseen käytettävät lauseet ovat siis jo valmiiksi järjestelmän tiedossa.

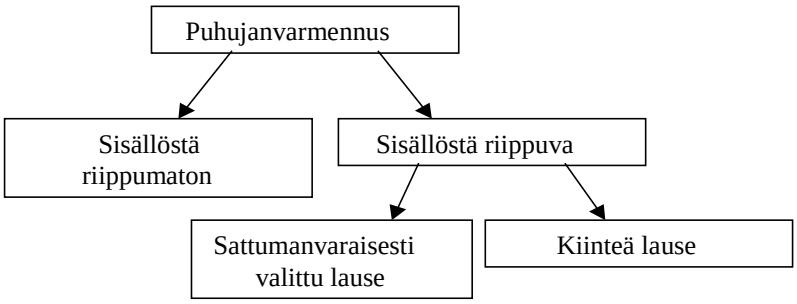
2. *Tekstin sisällöstä riippumattomia* (text-independent) tehtäviä, joissa käyttäjän kirjautumisvaiheessa antama puhenäyte ei riipu opetusvaiheessa annetusta näytteestä, vaan voi sisältää muitakin sanoja.

2.3.1 Sisällöstä riippuva tunnistustehtävä

Sisällöstä riippuvan tunnistustehtävän tapauksessa käyttäjä antaa varmennusjärjestelmälle järjestelmän opetusvaiheessa ääninäytteensä, jonka sisältö on ennalta määrätty ja koostuu esimerkiksi numeroista ja sanoista. Järjestelmä luo tästä ääninäytteestä puhujamallin kyseiselle puhujalle. Tämä puhujamalli toimii jatkossa kyseisen käyttäjän ”avaimena” järjestelmään. Kun käyttäjä yrittää myöhemmin kirjautua sisään järjestelmään, järjestelmä vaatii käyttäjää lukemaan opetusvaiheessa käytettyjä numeroita ja sanoja varmentamista varten. Näytteen saatuaan järjestelmä vertaa käyttäjän ääninäytettä ennalta tallennettuun puhujamalliin.

Sisällöstä riippuvan varmennusjärjestelmän vaatimat lauseet voivat olla *kiinteitä* (fixed phrase), tai *sattumanvaraisesti valittuja* (prompted phrase) [19] (kuva 2.6). Kiinteitä lauseita käytettävissä järjestelmissä käyttäjää vaaditaan toistamaan jokaisella sisäänkirjautumiskerralla sama lause, kun taas jälkimmäisellä tavalla toimiva järjestelmä arpoo jokaisella sisäänkirjautumiskerralla käyttäjälle luettavaksi uuden tekstin.

Sattumanvaraisella sisällöllä ja tiukalla aikamarginaalilla ääninäytteitä vaativat järjestelmät ovat kiinteitä puhenäytteitä vaativia järjestelmiä turvallisempia, koska näillä tekniikoilla voidaan estää, tai ainakin huomattavasti vaikeuttaa tallennetun puheäänen käyttömahdollisuutta järjestelmän sisäänkirjautumisessa [4, 20]. Toisaalta järjestelmän käytettävyyden kannalta toistettavan lauseen satunnaistaminen saattaa olla epäloogista ja käyttäjälle jopa epämiellyttävää: toistettavana lauseena voisi esimerkiksi olla ”viisi kaksi ankka yksi koira yhdeksän pää…”.



Kuva 2.6: Puhujanvarmennuksen luokittelu sisältöriippuvuuden mukaan.

Tekstin sisällöstä riippuva puhujanvarmennusjärjestelmä sopii hyvin esimerkiksi teollisuuslaitoksen ovenavausjärjestelmään, jossa käyttäjäryhmä on ennalta määrätty ja kaikilta käyttäjiltä on mahdollista saada opetusvaihetta varten ääninäytteet. Lisäksi puhenäytteen ei tarvitse sisäänkirjautumisvaiheessakaan olla kuin korkeintaan opetusvaiheessa käytetyn ääninäytteen mittainen [3, 23].

2.3.2 Sisällöstä riippumaton tunnistustehtävä

Tekstin sisällöstä riippumaton puhujanvarmennusjärjestelmä ei vaadi käyttäjältä tiettyä ennalta määrätyn sisällön omaavaa ääninäytettä. Varmennettavan henkilön ääninäyte voidaan

ottaa järjestelmän opetusvaihetta varten jo olemassa olevasta käyttäjän ääninäytteestä. Järjestelmän opetukseen tarkoitetun ääninäytteen sisällöllä ei ole varsinaista sisältövaatimusta, mutta näyte on sitä parempi mitä monipuolisemmin se sisältää käytettävän kielen yleisimpiä äänteitä [3]. Monipuolinen opetusääninäyte parantaa varmennustehtävän tarkkuutta, koska tunnistusvaiheessa käytettävä ääninäyte ei tällaisessa tilanteessa useinkaan sisällä kokonaisia samoja sanoja kuin järjestelmän opetusvaiheessa käytettiin.

Tekstin sisällöstä riippumaton puhujanvarmennusjärjestelmä sopii hyvin esimerkiksi poliisin käyttöön puhelinkuuntelutehtävän yhteydessä. Poliisin tutkija käyttää tällöin järjestelmän opettamiseen hallussaan olevaa epäillyn aiempaa ääninäytettä ja varmentaa puhujan henkilöllisyyden tuoreemmasta ääninäytteestä [24].

Tekstin sisällöstä riippumatonta puhujanvarmennusjärjestelmää ei kuitenkaan saada yhtä tarkaksi kuin tekstin sisällöstä riippuvaa järjestelmää [20, 23]. Tämä johtuu siitä, että huijarin on mahdollista nauhoittaa oikean käyttäjän puhetta ja näin päästä sisälle järjestelmään esiintymällä toisena henkilönä. Lisäksi sisällöstä riippumaton järjestelmä tarvitsee tunnistusvaihetta varten jopa 10-30 sekunnin mittaisen ääninäytteen [23]. Omissa testeissämme olemme kuitenkin todenneet jopa huomattavasti alle kymmenen sekunnin ääninäytteen tarkan tunnistustuloksen saamiseen. Tämä johtunee kuitenkin testeissämme käytössä olleista suhteellisen pienistä puhujatietokannoista.

3 PUHEEN TUOTTAMISESTA JA HAVAITSEMISESTA

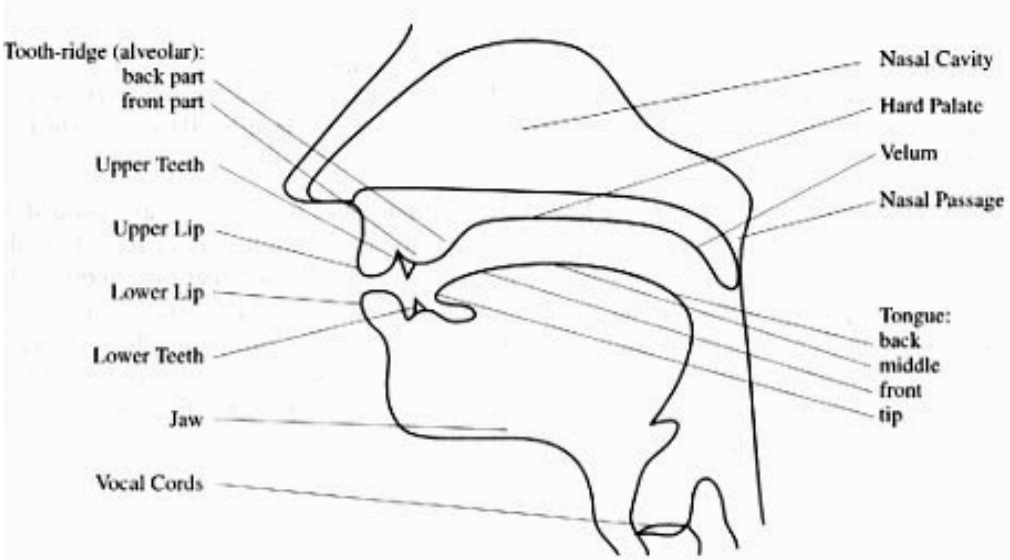
Tässä luvussa kuvataan ihmisen puheentuottoon ja havaitsemiseen liittyvät perusteet yleisellä tasolla. Puhuminen on ihmiselle luonnollinen tapa kommunikoida [14, 20]. Ihminen käyttää puhetta tiedon välittämiseen itseltään kuuntelijalle. *Puheen tuottaminen* ja *puheen havaitseminen* ovat molemmat tärkeitä komponentteja puheketjussa. Puheen tuottaminen alkaa aivoissa tapahtuvasta ajatuksesta ja päätöksestä puhua. Aivot aktivoivat puheentuottamiseen tarvittavat lihakset ja tuottavat näitä lihaksia liikuttamalla puheäänen. Puheen havainnoija eli kuuntelija ottaa puhesignaalin vastaan kuulojärjestelmäänsä ja prosessoi puheen aivojen ymmärtämään muotoon [14].

Puhe ei ole pelkästään toisiaan seuraavia, helposti erotettavia yksikköjä kuten äänteitä tai sanoja, vaan se koostuu useista toisiinsa hierarkisessa suhteessa olevista tekijöistä [36]. Puhe tuotetaan erilaisin puhe-elinten liikkein, jotka ovat ajallisesti sekä toisiaan seuraavia että samanaikaisia. Seuraavaksi käydään läpi puhe-elimet ja niiden tehtävät äänentuottoprosessissa.

Ihmisen äänen ominaisuudet määräytyvät puheen tuottamiseen osallistuvien elimien koon ja muodon sekä ympäristöstä opitun puhetavan perusteella. Keuhkot toimivat puheprosessissa ilmalähteenä ja puheen tuottaminen perustuu keuhkoissa olevan ilman liikkumiseen. Tätä keuhkoista lähtevää ilmavirtaa moduloidaan äänen tuottamiseen osallistuvien puhe-elinten eri asennoilla. Ilmavirta muuttuu kuultavaksi, kun siihen syntyy epätasaisuuksia [14].

3.1 Puhe-elimet

Ihmisen puhe-elimet voidaan jakaa keskushermoston kannalta kahteen osaan: *perifeerisiin* eli varsinaisiin ja *sentraalisiin* eli aivoissa sijaitseviin puhe-elimiin. Perifeerisiä puhe-elimiä ovat esimerkiksi kieli, kurkunpää ja huulet. Sentraalisiksi puhe-elimiksi kutsutaan aivojen puheen tuottamiseen osallistuvia rakenteita [14]. Kuvassa 3.1 on esitetty tärkeimmät perifeeriset puhe-elimet. Kuvasta nähtävien elinten lisäksi perifeerisiä puhe-elimiä ovat myös esimerkiksi keuhkot, pallea ja henkitorvi.



Kuva 3.1: Ihmisen perifeeriset äänentuottoelimet [14].

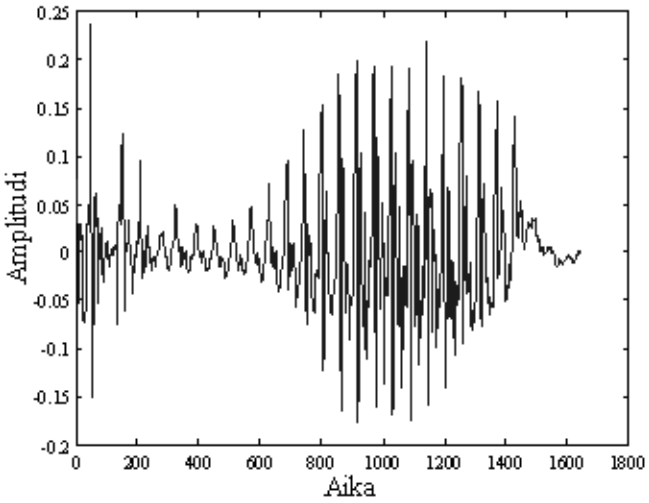
3.1.1 Äänentuotto

Kun ihminen hengittää sisään, pallea ja kylkivälilihakset jännittyvät. Tällöin rintakehä laajenee ja keuhkoihin syntyy samalla alipainetta. Tämä aiheuttaa ilman virtaamisen keuhkoihin. Ulos hengittäessä pallea ja kylkivälilihakset puolestaan rentoutuvat, rintakehä supistuu ja ilma virtaa ulos keuhkoihin syntyneen ylipaineen vaikutuksesta [14]. Puhe tapahtuu yleensä uloshengityksen aikana, mutta esimerkiksi suomenkielessä puhetta tuotetaan joskus myös sisään hengittäessä [20].

Kun äänihuulet (vocal cords) ovat lähellä toisiaan ja värähtelevät toisiaan vastaan hengityksen aikana, tuloksena syntyy puheääni [14]. Kitapurje (velum) toimii eräänlaisen venttiilin tavoin, avaten äänelle kulkuväylän nenäonteloon. Kitapurjeen ja nenäontelon avulla muodostetaan äänteet /m/ ja /n/. Kieli (tongue) mahdollistaa kovan kitalaen (hard palate) ja hampaitten kanssa konsonanttien ääntämisen. Vokaalien muodostamiseen ei käytetä äänihuulten lisäksi aktiivisesti juurikaan muita äänentuottoelimiä kuin kieltä ja huulia. Vokaaleita äännettäessä huulet ovat koko ajan avoinna, mutta huulet mahdollistavat täysin kiinni ollessaan myös joidenkin konsonanttien (/p/, /b/, /m/) muodostamisen [14].

Puhesignaalin muuntaminen ihmisen kuultavissa olevasta muodosta tietokoneen ymmärtämään muotoon edellyttää puheen (ilmanpaineen muutosten) mittaamista mikrofonin avulla

jännitteen vaihteluna [14, 20]. Kuvassa 3.2 on tyypillinen esimerkki puhesignaalin *aaltomuodosta*. Aaltomuodoksi kutsutaan esitystapaa, jossa signaali esitetään piirtämällä ilmanpaineen (jännitteen) vaihtelut aikatasossa.



Kuva 3.2: Aaltomuodossa esitettävä puhesignaali. Naishenkilö lausuu sanan (”ovat”). Kuvassa on vaaka-akselilla näytearvojen indeksit (aika), pystyakselilla näytteiden arvot (ilmanpaine/jännite). Kuvan signaali on näytteistetty 8000 Hz taajuudella (puhelinlaatu).

Puhesignaalia tietokoneella analysoitaessa signaali joudutaan muuntamaan piirreirrotusta varten aaltomuotoesityksestä taajuustasoesitykseksi. Seuraavassa luvussa kerrotaan kuinka tämä muunnos tehdään ja kuinka puheesta voidaan muunnoksen jälkeen irrottaa puhujan henkilökohtaisia puheominaisuuksia kuvaavia piirteitä.

3.2 Kuuloelimet ja puheen havaitseminen

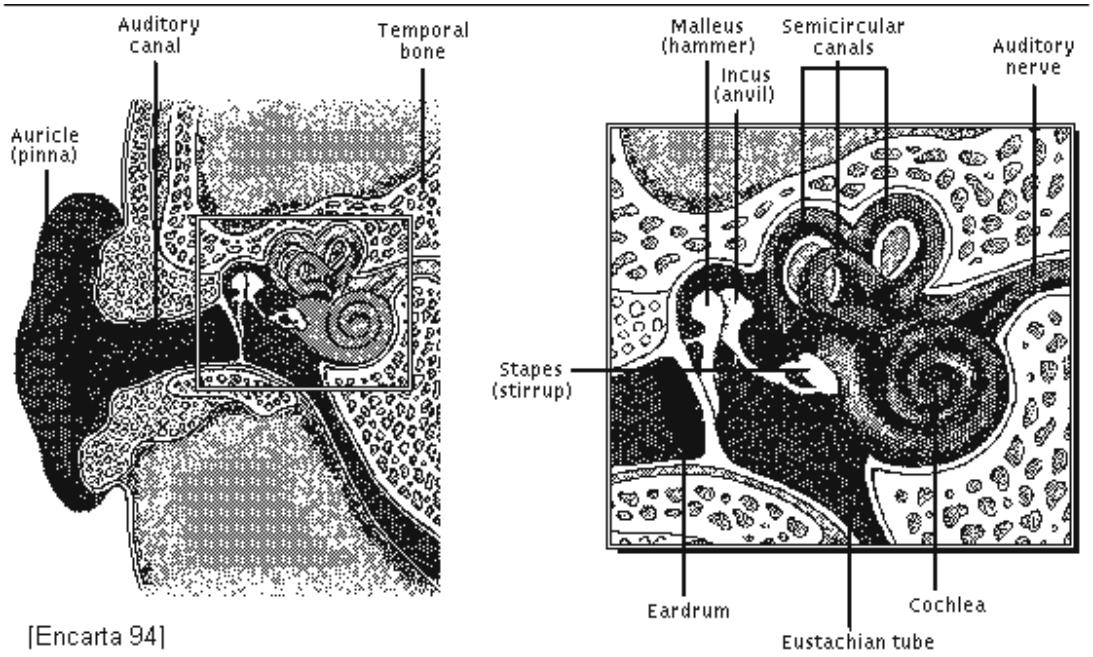
Piirreirrotuksessa käytetään usein ns. psykoakustisia malleja, eli mallinnetaan ihmisen kuulohavaintoprosessia. Tässä kohdassa esitellään ihmisen kuulohavaintoprosessi. Ihminen käyttää puheen havaitsemiseen kuuloelimiään ja puheen ymmärtämiseen aivoissa sijaitsevaa puheen ymmärtämisen mahdollistavaa osaa.

3.2.1 Kuuloelimet

Ihmisen kuuloelimet muodostuvat viidestä pääosasta [8], jotka ovat:

1. korvalehti (auricle),
2. korvakäytävä (auditory canal),
3. tärykalvo (eardrum),
4. kuuloluut (hammer, anvil, stapes),
5. korvasimpukka (cochlea).

Ainoa selkeästi ulospäin näkyvä kuuloelin ihmisellä on korvalehti. Korvalehti kerää ääniaaltoja, jonka jälkeen ääni kulkee korvakäytävää pitkin tärykalvolle. Tärykalvo värähtelee ilmanpaineen vaihteluiden mukaisesti, ja kuuloluiden muodostama vipujärjestelmä johtaa värähtelyt korvasimpukan eteisikkunaan (oval window) (kuva 3.3) vahvistaen ne sellaisiksi, että ne voivat edetä nesteessä [14]. Korvatorvi (eustachian tube) on luun, ruston ja limakalvon muodostama kanava, joka yhdistää välikorvan nenänieluun ja toimii välikorvan ilmastoijana ja eritteiden poistajana.



Kuva 3.3: Ihmisen kuuloelimet [8].

Kuuloelinten monimutkaisin osa on sisäkorvassa sijaitseva korvasimpukka. Korvasimpukassa sijaitsevat varsinaiset reseptorit, eli tuntoaistimet. Korvasimpukan käytävässä on ns. tyvilevy,

jonka eri kohdilla on eri ominaistaajuus [8]. Tietynkorkuinen ääni saa tyvilevyn tietyn kohdan värähtelemään tietyllä voimakkuudella (riippuen kyseisen äänen amplitudista) ja tyvilevyssä olevat karvalliset kuulosolut johtavat impulssiin yläpuolella olevaan katekalvoon koskettamalla sitä ja aiheuttamalla siinä olevien hermosolujen ärtymisen. Näin ollen äänien eri taajuudet ärsyttävät vain tiettyjä soluja ja signaali voidaan kokonaisuudessaan johtaa aivoihin tulkittavaksi [14].

Korvasimpukan rakenne on yksinkertaisesti seuraavanlainen: kuuloluut rummuttavat eteisikkunaa aiheuttaen paineenvaihteluja simpukan sisällä olevassa nesteessä; simpukan toisessa päässä oleva pyöreä ikkuna tasoittaa paineen (nesteen kokonaistilavuus ei voi muuttua).

3.2.2 Puheen havaitseminen

Kuuloaistimus kulkee sähköisinä hermoimpulsseina sisäkorvasta lähtevää kuulohermoa (auditory nerve) pitkin aivojen ohimolohkon kuuloalueelle [8]. Aivojen kuuloalueella ihminen tekee varsinaisen kuulohavainnon ja hermoimpulssit saavat ymmärrettävän muodon. Ääniaallot saavuttavat korvat yleensä hieman eri aikaan, mikä tekee mahdolliseksi äänen tulosuunnan arvioimisen. Ihmiskorvalle riittää tähän noin 0.0001 sekunnin ero (joillain eläinlajeilla tämä aika on vielä merkittävästi lyhyempi) [8].

Puheen havaitsemiseen liittyy niin sanottu *invarianssiongelma*: puhe on akustisesti täynnä vaihtelua ja kielelliset yksiköt ovat usein foneettisesti hyvinkin puutteellisesti tuotettuja [36]. Tästä huolimatta kuulijat yleensä tunnistavat erilaiset variantit samaksi yksiköksi. Kuulija pystyy ilmeisesti poistamaan äänen havaitsemisprosessista vain puhujasta aiheutuvat ominaisuudet [36].

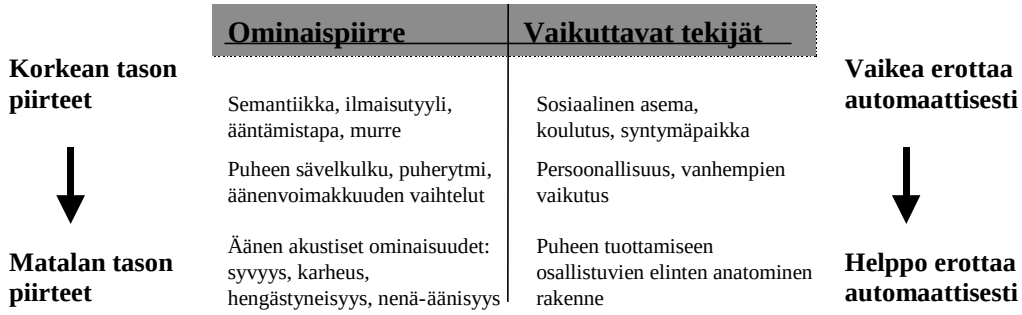
4 PIIRREIRROTUS JA PUHUJAMALLIT

Tässä luvussa käymme lyhyesti läpi automaattisen piirreirrotusprosessin. Luku etenee kohdittain seuraavasti. Kohdassa 4.1 kerromme puheen ominaispiirteistä ja niiden jaottelusta. Kohdassa 4.2 perustelemme puhesignaalin esikäsittelyn merkitystä piirreirrotuksessa. Kohdassa 4.3 käymme läpi piirreirrotuksen perusteet. Kohdassa 4.4 ja 4.5 kerromme piirteiden tilastollisen mallintamiseen sekä puhujamallien vertailuun liittyvät perusteet. Tässä luvussa käsittelemämme peruskäsitteet mahdollistavat myöhempien lukujen sisällön syvällisen ymmärtämisen.

4.1 Johdanto

Puhe sisältää tiettyjä ominaispiirteitä, joiden perusteella puhujan henkilöllisyys voidaan varmistaa. Kuvan 4.1 mukaisesti nämä piirteet voidaan jaotella korkean tason ja matalan tason piirteisiin. Korkean tason piirteet ovat yleensä ihmisen elinympäristöstä opittuja ominaisuuksia, eivätkä riipu ihmisen puhe-elimien rakenteesta. Matalan tason piirteet puolestaan perustuvat ihmisen fysiologisiin puhe-elinten ominaisuuksiin.

Ihminen tunnistaa puhujan yleensä käyttämällä osaa tai kaikkia näistä piirteistä. Toisaalta ei ole täysin selvää mitä kaikkia vihjeitä ihminen todella käyttää varmentamiseen. Tietokoneella tehtävässä puhujantunnistuksessa käytetään matalan tason piirteitä, koska ne voidaan erottaa puheesta helpommin automaattisesti [14, 20].



Kuva 4.1 Puheen ominaispiirteiden jaottelua [20].

4.2 Puhesignaalin esikäsittely

Signaalille tehdään ennen piirreirrotusta yleensä *esikäsittelyoperaatioita*, joilla signaalia muokataan mahdollisimman hyvin analysoitavaan muotoon. Usein puhesignaali joudutaan esimerkiksi *esikorostamaan*, tai siitä joudutaan poistamaan kohinaa tai kaikua. Signaalin esikorostuksen tarkoituksena on korostaa signaalin korkeita taajuuksia, jotka kirjallisuuden mukaan sisältävät erityisen paljon puhujakohtaista informaatiota [17, 25]. Esikorostus tehdään yleensä seuraavanlaisella suodattimella [6]:

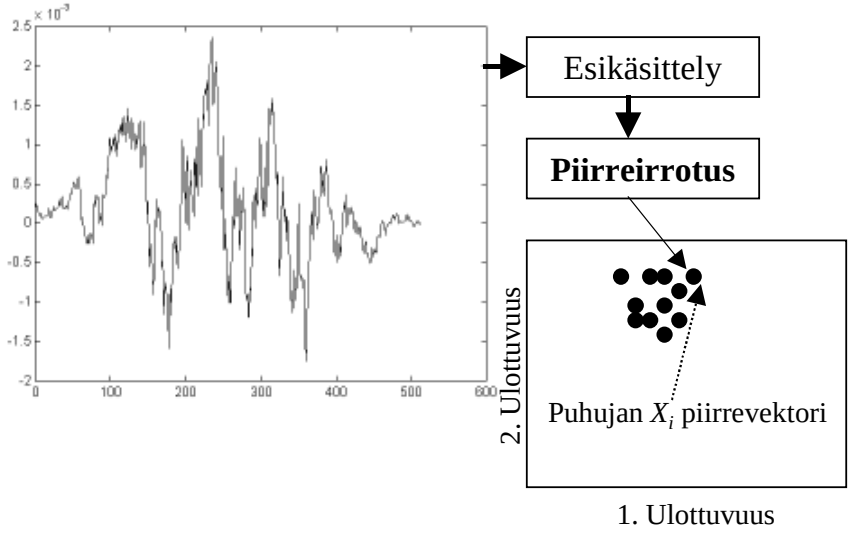
$$output(n) = input(n) - a * input(n-1). \tag{3.1}$$

Suodattimen perusteella ulostulosignaalin *output(n)* näytearvo saadaan vähentämällä vastaavasta sisäänmenosignaalista *input(n)* sitä edeltävän näytteen arvo *input(n-1)* kerrottuna vakiolla *a*. Oletusarvona suodattimen (3.1) kertoimella *a* on jokin lukua 1 lähellä oleva arvo, esimerkiksi 0.95. Kerroin voidaan laskea myös adaptiivisesti analysoitavan kehyksen *soinnillisuusasteen* (voicing degree) perusteella [6]. Tällöin soinnittomia kehyksiä ei korosteta niin paljon kuin soinnillisia.

4.3 Piirreirrotus

Piirreirrotus tarkoittaa puhesignaalissa olevien puhujaa kuvaavien muuttujien irrottamista muista puheen piirteistä [3]. Piirreirrotuksen tuloksena puhemateriaalin piirteet muunnetaan puhenäytettä kuvaaviksi piirrevektoreiksi (kuva 4.2). Tarkoituksena on tehdä muunnos siten, että piirteet siirretään verrattain pieniulotteiseen piirreavaruuteen, jossa vektorit kuitenkin erottelevat puhujat toisistaan mahdollisimman tehokkaasti.

Kaikkien puhujien piirrevektorit sijoitetaan piirreavaruuteen ja ryhmitellään tämän jälkeen luokittelualgoritmilla. Samanlaisia äänteitä vastaavat piirrevektorit sijaitsevat piirreavaruudessa lähellä toisiaan, kun taas erilaiset äänteet sijaitsevat kaukana toisistaan [17]. Ryhmittelyn tuloksena piirrevektoreista muodostuvat puhujien puhujamallit. Löydettyjen piirteiden tulee olla samalla mahdollisimman hyvin järjestelmän toimintaa palvelevia, eli piirteiden väliset erot puhujien välillä täytyy pystyä mittaamaan yksinkertaisilla samankaltaisuusmittareilla [3].



Kuva 4.2: Puhesignaalin muuntaminen piirrevektoreiksi.

4.3.1 Syötesignaalin jakaminen kehyksiin

Signaalia ei analysoida piirreirrotusvaiheessa kokonaisena, vaan se on jaettava ennen analysointia lyhyisiin osiin, joita sanotaan *kehyksiksi* (frame). Oleellista on jakaa signaali oikean mittaisiin kehyksiin. Optimi kehyksen pituus riippuu varmennusjärjestelmän toteutuksesta, mutta voi olla esimerkiksi 10-30 millisekuntia [4]. Tavallisesti signaali jaetaan noin 20 millisekunnin mittaisiin kehyksiin. Kehykset muodostetaan tavallisesti siten, että seuraava kehys menee osittain edellisen kehyksen päälle. Kehyksen peittoaste on tavallisesti hieman alle puolet kehyksen pituudesta [17, 30].

Jokaista kehystä käsitellään piirreirrotusvaiheessa omana erillisenä signaalinaan ja jokaiselle kehykselle tehdään samat piirreirrotusvaiheen operaatiot riippumatta toisten kehysten sisällöstä. Jotta signaali voidaan jakaa halutun mittaisiin kehyksiin, on kehyksen pituuden lisäksi tiedettävä signaalin *näytteenottotaajuus*. Näytteenottotaajuus tarkoittaa puheesta yhdessä sekunnissa kerättävien näytteiden lukumäärää. Yhden kehyksen pituus l olisi näin laskettuna:

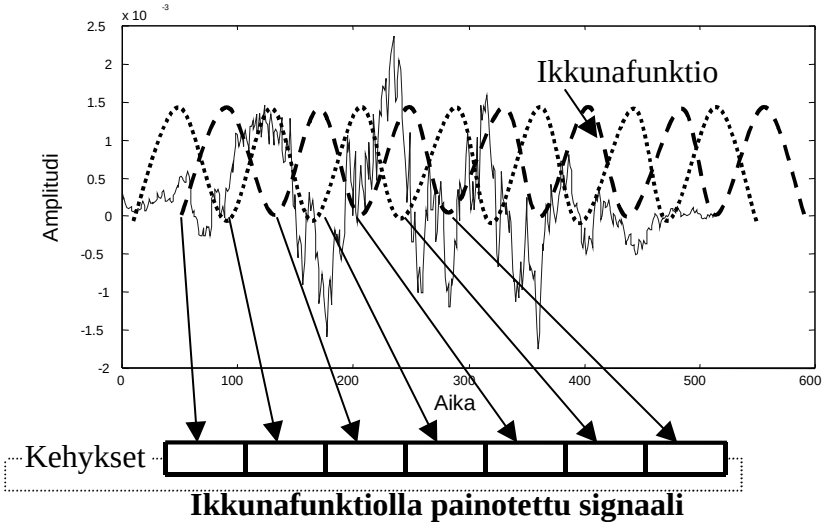
$$l = s * Fs \tag{3.2}$$

Kaavassa (3.2) s tarkoittaa sekunteja ja Fs on puolestaan signaalin näytteenottotaajuus. Koska yksi millisekunti (*ms*) = 1/1000 sekuntia (*s*), on esimerkiksi 20 millisekuntia sekunteina:

20*ms* * (1/1000)*s* = 0.020 sekuntia. Jos signaalin näytteenottotaajuus olisi 8000 Hz, tulisi yhden kehyksen pituudeksi kaavan (3.2) mukaisesti laskettuna 0.020 * 8000 = 160 näytettä. Kehysten muodostamisen jälkeen kehykset kerrotaan ns. ikkunafunktiolla, joita käsittelemme seuraavaksi.

4.3.2 Kehyksen ikkunointi

Ikkunointi on syötesignaalin kertomista painofunktiolla ns. *valevärähtelyjen* vähentämiseksi signaalin spektrissä, joiden vaikutus olisi suurempi ilman ikkunointia [6]. Ikkunoinnissa alkuperäistä kehystä painotetaan siten että se muuttuu paremmin analysoitavaksi. Ikkunointia varten lasketaan *ikkunafunktio*, joka on saman mittainen kuin signaalista tehtävät kehykset (kuva 4.3). Kehyksen sisällön on tarkoituksena pehmentyä ilman tärkeiden piirteiden häviämistä. Ikkunoinnin jälkeen painotetut kehykset muunnetaan Fourier-muunnoksella signaalin *spektriksi* (spectrum). Spektriesitysmuodossa signaalia analysoidaan aikatason sijasta taajuustasossa [6].

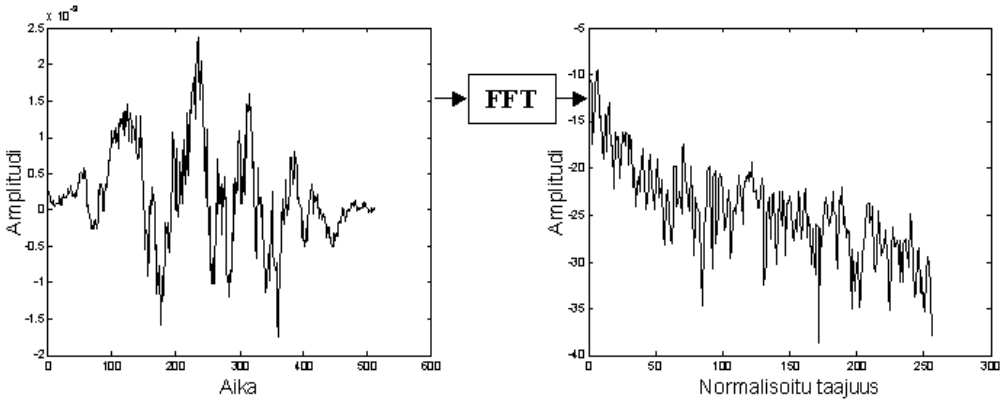


Kuva 4.3: Puhesignaalin ikkunointi

4.3.3 Fourier-muunnos

Fourier-muunnos on menetelmä, jolla päästään käsiksi puhesignaalin taajuusinformaatioon [6]. Fourier-muunnos muuttaa aikatason signaalin taajuustasoon, jossa yksittäiset taajuusfunktion kompleksit pisteet kuvaavat vastaavan taajuuskomponentin amplitudia ja vaihetta [27]. *Nopea Fourier-muunnos* (FFT, Fast Fourier Transform) on tietty, optimoitu tapa järjestää termit diskreetin Fourier-muunnoksen summauksessa. Tavallisen diskreetin Fourier-muunnoksen aikakompleksisuus on N:n näytteen pituiselle signaalille $O(N^2)$, kun taas FFT:n kompleksisuus on $O(N \log_2 N)$ [6].

FFT vaatii syötteenään kehyksen, jonka pituus on 2^n näytettä. Tämä täytyy ottaa huomioon jaettaessa signaalia ikkunoihin. Tästä vaatimuksesta johtuen ikkunaa täytyy tarvittaessa jatkaa (tai lyhentää) siten, että sen pituus on jokin kahden potenssi. FFT:n tuloksena saadaan siis spektri, jossa signaali siirretty on aikatasosta taajuustasoon (kuva 4.4).



a) Ikkunoitu kehys b) Amplitudispektri

Kuva 4.4: Puhesignaalista FFT-suodatettu spektri.

4.4 Piirteiden tilastollinen mallinnus vektorikvantisoinnilla

Piirreirrotusvaiheessa syntyneistä piirrevektoreista luodaan seuraavaksi vektoreiden tilastollista jakaumaa kuvaava puhujamalli, joka tallennetaan järjestelmän tietokantaan. Signaalista irrotetut piirteet täytyy *mallintaa* kuvaamaan kyseisen puhujan yksilöllisiä piirteitä. Mallinnus kuuluu olennaisena osana hahmontunnistusprosesseihin sekä opetus- että tunnistusvaiheessa. Opetusvaiheessa luodaan piirrevektoreiden avulla kaikille *luokille* matemaattiset mallit, jotka kuvaavat näiden luokkien yksilöllisiä ja erottelevia piirteitä [7].

Puhujantunnistusongelmassa luokat ovat puhujia [16]. Opetuksen tuloksena syntyy puhujamalli. Tunnistusvaiheessa varmennettavan puhujan puhujamallia verrataan tietokannassa olevaan väitetyn henkilön puhujamalliin. Eräs luokittelumenetelmä on *vektorikvantisointiin* (VQ, Vector Quantization) perustuva luokittelu, joka on puheen- ja puhujantunnistuksessa sekä puheen koodauksessa hyväksi todettu ja helposti toteutettava mallinnusalgoritmi [16, 35].

VQ on periaatteeltaan tiedonpakkausalgoritmi, jonka avulla suuresta vektorijoukosta voidaan tiivistää alkuperäisiä vektoreita yhdistämällä alkuperäistä huomattavasti pienempi vektorijoukko. Yhdistettävien vektoreiden valinta perustuu johonkin *optimaalisuuskriteeriin* [9]. Yleensä kaksi yhdistettävää vektoria valitaan siten että ne sijaitsevat lähellä toisiaan. Näitä VQ:n avulla valittuja vektoreita kutsutaan *koodivektoreiksi* ja koodivektoreista muodostettua joukkoa kutsutaan usein *koodikirjaksi* [35].

Eräs koodikirjan muodostusalgoritmi on *PNN* (Pairwise Nearest Neighbor), joka yleisellä tasolla esitettynä etenee seuraavalla tavalla [9]:

1. Valitaan haluttu koodikirjan koko, eli kuinka monta koodivektoria koodikirjaan valitaan.
2. Erotetaan jokainen vektori erilliseksi koodivektoriksi. Algoritmi etenee hierarkisesti siten, että jokaisella algoritmin kierroksella yhdistetään kaksi uutta koodivektoria ja algoritmin suoritusta jatketaan kunnes haluttu koodikirjan koko saavutetaan.
3. Koodivektorit valitaan siten, että kyseisten vektoreiden yhdistämisestä aiheutuva vektoriavaruuden väärentymä on mahdollisimman pieni.

Koodikirjaa tulee myös päivittää ajan mittaan. Päivitystarve johtuu esimerkiksi siitä että puhujan ääni ei pysy samanlaisena ajan kuluessa, vaan ikä muokkaa äänen piirteitä [4], myös

opetus- ja tunnistusvaiheen äänitysolosuhteet saattavat poiketa toisistaan. Toisessa saattaa olla mukana taustamelua tai muita häiriötekijöitä, jotka toisesta puuttuvat [16].

4.5 Puhujamallien vertailu

Kun varmennettava puhuja antaa tunnistusvaiheessa ääninäytteensä, näytteestä voidaan muodostaa ensin puhujamalli samalla tavalla kuin opetusvaiheessakin (jos puhujamalli on luotu VQ-algoritmia [35] käyttämällä, puhujamallia kutsutaan koodikirjaksi). Toisaalta varmennusvaiheessa kerätyn ääninäytteen piirteitä voidaan vertailla tietokannassa olevaan koodikirjaan suoraankin, ilman uuden koodikirjan muodostamista. Syynä varmennusmateriaalin koodikirjan muodostamiseen voi olla mahdollinen ajansäästö ja parantunut tarkkuus. Tätä ei kuitenkaan ole vielä järjestelmällisesti tutkittu.

Seuraavaksi tätä puhujamallia verrataan tietokannassa olevaan varmennettavan puhujan malliin. Puhujamallien vertailualgoritmi on varmennustehtävässä yksinkertainen, tunnistustehtävässä algoritmi on monimutkaisempi. Seuraavassa on esitetty yksinkertainen puhujamallien vertailualgoritmi varmennustehtävässä [4]:

ALGORITMI 4.1: Puhujamallien vertailualgoritmi varmennustehtävässä.

Syötteet : väite puhujasta (i), puhujan malli Ci.
Tuloste: HYVÄKSY/HYLKÄÄ -päätös.

1. Lasketaan piirrevektorit $X=\{\mathbf{x}_1, \mathbf{x}_2,\ldots, \mathbf{x}_T\}$ varmennettavalle puhujalle X.
2. Lasketaan etäisyys $D = d(X, C_i)$ piirrevektoreiden X ja varmennettavan puhujan mallin C_i välillä.
3. Oletetaan, että etukäteen on laskettu puhujakohtainen kynnysarvo T_i. Jos etäisyys D on pienempi kuin kynnysarvo T_i, puhuja hyväksytään. Muuten puhuja hylätään.

Algoritmissa 4.1 ei ole huomiotu etäisyyksiä muihin puhujiin eikä algoritmissa käytetä hyväksi vastinjoukkoja.

5 Varmennusjärjestelmien arviointi

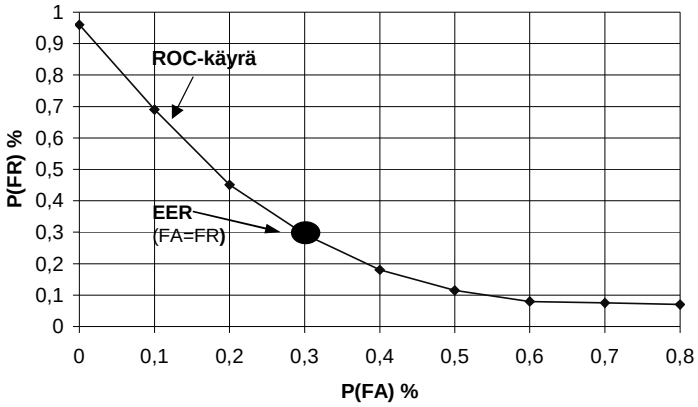
Tässä luvussa tutustutaan puhujanvarmennusjärjestelmien arviointiin eli evaluointiin käytettäviin menetelmiin. Arvioinnin perusteena voi olla joko parhaimman järjestelmän etsiminen, jolloin arvioijana toimii esimerkiksi järjestelmää valitseva henkilö, tai toisaalta oman varmennusjärjestelmän kehittäminen, jolloin arvioijana toimii järjestelmää kehittäjät. Tässä tutkielmassa keskitytään jälkimmäisen arviointiperusteen tarkasteluun.

Varmennusjärjestelmien arvioinnin päämääränä ovat entistä varmemmat ja luotettavammat varmennusjärjestelmät. Varmennusjärjestelmien luotettavuutta arvioidaan erilaisilla tavoilla, joista yleisimmin käytettyjä ovat järjestelmän virheherkkyyttä mittaavat virhetulosparametrit [4]:

1. *Oikean puhujan hylkäämisvirhe* (FR, false rejection rate), joka kertoo kuinka monta prosenttia oikeista puhujista järjestelmä keskimäärin hylkää.

2. *Huijarin hyväksymisvirhe* (FA, false acceptance rate), joka kertoo kuinka monta prosenttia huijareista pääsee keskimäärin sisään järjestelmään.

Virhetulosparametrien kuvaajat voidaan yhdistää samaan koordinaatistoon niin sanottuun ROC (Receiver Operating Characteristics) –kuvaajaan, jossa FR- ja FA-arvojen leikkauspiste kertoo järjestelmän *keskimääräisen virhealttiuden* (EER, Equal Error Rate) (kuva 5.1).



Kuva 5.1: ROC-käyrä ja keskimääräinen virhealttius (EER).

FR- ja FA-virhearvoista voidaan myös laskea *normalisoidut keskiarvot* (HTER, Half Total Error Rate). Edellä mainittuja virhemittareita käsittelemme kohdissa 5.1 - 5.5. Lisäksi varmennusjärjestelmiä voidaan vertailla järjestelmän *kokonaissuoritusaikojen* (VT, Verification Throughput) perusteella, jota käsittelemme kohdassa 5.6. Kohdissa 5.7 ja 5.8 pohdimme puhujanvarmennusjärjestelmän vahvuuksia ja heikkouksia verrattaessa niitä muihin tunnistusmenetelmiin.

5.1 Oikean puhujan hylkäämisvirhe (FR)

Oikean puhujan hylkäämisvirhe (FR, False Rejection) tapahtuu, kun järjestelmä hylkää *aidon puhujan*. Aidolla puhujalla tarkoitetaan puhujaa, joka on järjestelmän hyväksytty käyttäjä ja yrittää kirjautua sisään järjestelmään esiintymällä itsenään. Monissa tunnistusjärjestelmissä tällaisille virheille annetaan noin 5 prosentin marginaali kaikista varmennetuista ääninäytteistä [23]. Esimerkiksi erittäin laajan käyttäjäkunnan omaavalla puhelinvälitteisellä varmennusjärjestelmällä, jossa käyttäjiä on jopa tuhansia, jo yli 10 prosentin virheosuus voisi olla joustavan käytön kannalta liian suuri [23].

Virhe johtuu usein järjestelmän lisäksi myös ääninäytteen huonosta teknisestä laadusta. Puhelinvälitteisessä järjestelmässä puhelinlinjasta johtuvat rasahdukset ja äänen katkokset hankaloittavat puhujan varmennusjärjestelmän toimintaa. Virheestä ei koidu vakavaa tietoturvavahinkoa ja virhe voidaan usein poistaa käsittelemällä käyttäjän uusi ääninäyte.

5.2 Huijarin hyväksymisvirhe (FA)

Huijarin hyväksymisvirhe (FA, False Acceptance) tapahtuu, kun järjestelmä hyväksyy huijarin. Huijarilla tarkoitetaan tässä puhujaa, joka yrittää sisään järjestelmään imitoimalla jotain toista, aitoa puhujaa. FA-tyyppiset virheet ovat huomattavasti FR–tyyppisiä virheitä vakavampia. Virhe käytännössä romahduttaa käytettävän järjestelmän luotettavuuden esimerkiksi ovenaukaisupalvelimena muuten tarkasti varjellussa teollisuuskohteessa.

Hyväksytty virhemarginaali tämän tyyppisten virheiden esiintymisille tarkasti suojelluissa järjestelmissä voi olla esimerkiksi 0.1 prosenttia kaikista varmennetuista ääninäytteistä [23]. Erittäin tarkasti suojelluissa kohteissa, joissa on vähän käyttäjiä (n. 50 – 100 kpl.), kuten armeijan tietokonekeskuksessa, varmennusjärjestelmän virhemarginaalin FA-tyyppisten virheiden osalta tulee olla jopa tätäkin pienempi [23].

5.3 Keskimääräinen virhealttius (EER)

Keskimääräinen virhealttius saadaan etsimällä virhearvojen FA ja FR kuvaajien leikkauskohta. Toisin sanoen EER on se arvo, jossa FA- ja FR-arvot ovat samat. Esimerkiksi järjestelmässä jossa FR-tyyppisten virheiden määräksi on saatu 2.5 % ja FA-tyyppisten virheiden määräksi samoin 2.5 % kaikista puhujista, tulee EER:n arvoksi myös 2.5 %.

EER:n käyttöä järjestelmän yleisen virhealttiuden määrittelyssä voidaan kuitenkin mielestäni hieman kyseenalaistaa. Onhan nimittäin tosiasia, että FA- ja FR-tyyppisten virheiden tulee olla erilaisia eri järjestelmissä. Esimerkiksi turvallisuuspalvelussa toimivalla järjestelmällä on aivan oleellista saavuttaa FA:n arvoksi FR:a reilusti pienempi suhteellinen arvo, koska tällaisissa järjestelmissä on tärkeintä etteivät huijarit pääse sisälle. FA-tyyppisten virheiden totennäköisyys kyseisessä järjestelmässä olla jopa lähellä 10 prosenttia, kun taas FA tulee saada ainakin lähelle 0.1 prosentin arvoa, ehkä jopa alle sen [23]. Toisaalta paljon käyttäjiä omaavissa järjestelmissä, kuten esimerkiksi hotellihuoneiden varauspalvelu, käytön sujuvuus on yleensä pääasia ja näin FR-tyyppisten virheiden todennäköisyys täytyy saada pieneksi, jolloin FA-tyyppisten virheiden määrä kasvaa [23].

Periaatteessa EER kertoo järjestelmän virhealttiuden erittäin hyvin ja suoraviivaisesti, mutta käytännössä se ei kerro mitään järjestelmän sisäisestä toteutuksesta FA- ja FR–tyyppisten virheiden välillä. Laadukkaissa puhujanvarmennusjärjestelmissä vaaditaan todellisuudessa erilaiset kynnysarvot kaikille puhujille, koska näin päästään parempaan tarkkuuteen kuin yleistä kynnysarvoa käyttämällä [3, 20]. Tämä vie pohjaa varmennusjärjestelmän keskimääräisen virhealttiuden (EER) antaman järjestelmän luotettavuuskuvan järkevyydeltä. Näin EER ei mielestäni sovellu niinkään kynnysarvon määrittelyyn varsinkaan järjestelmissä joissa käytetään kaikille puhujille yhteistä kynnysaroa, vaan ainoastaan varmennusjärjestelmän arviointiin.

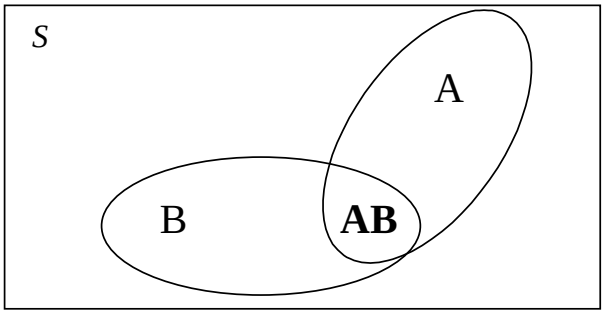
Seuraavaksi tarkastellaan esimerkin vuoksi taulukoita, joihin on laskettu puhujanvarmennus-järjestelmän läpipäästö- ja hylkäysprosentteja mahdollisessa todellisessa järjestelmässä.

ESIMERKKI 5.1: Puhujanvarmennusjärjestelmien EER-tulokset.

Esimerkin taulukoissa käytetään todennäköisyyslaskennan tulosääntöä [33]:

$$P(A \wedge B) = P(A)P(B) \tag{5.1}$$

Tulosääntö (5.1) tarkoittaa, että tapahtumien *A* ja *B* yhtäaikaisen toteutumisen todennäköisyys on kyseisten tapahtumien todennäköisyyksien tulo. Kuva 5.2 esittää tulojoukkoja todennäköisyysavaruudessa.



Kuva 5.2: Tulojoukot A ja B todennäköisyysavaruudessa [14].

Järjestelmä 1 tunnistaa aidon puhujan 95 prosentin todennäköisyydellä ja väittää aitoa käyttäjää huijariksi 5 prosentin todennäköisyydellä. Lisäksi järjestelmä 1 päästää huijarin sisälle 5 prosentin todennäköisyydellä ja tunnistaa huijarin huijariksi 95 prosentin todennäköisyydellä. Järjestelmässä jossa on vain yksi yrityskerta puhenäytteen antamiseen ja EER = 5% (taulukko 5.1).

Järjestelmä 2 tunnistaa aidon puhujan ensimmäisellä yrityskerralla 95 prosentin todennäköisyydellä. Toisella kerralla tunnistetuksi tuleminen vaati ensin yhden hylkäyksen, joka tapahtuu 5 prosentin todennäköisyydellä. Näin todennäköisyys tälle tapahtumalle on 0.5 * 0.95 = 0.047, eli 4.7 prosenttia. Vastaavasti kolmen peräkkäisen hylkäyksen todennäköisyys aidolle puhujalle on 0.5 * 0.5 * 0.5 = 0.0001, eli 0.01 prosenttia. Järjestelmässä on kolme yrityskertaa puhenäytteen antamiseen, puhujat aitoja puhujia ja EER = 5% (taulukko 5.2).

Järjestelmä 3 päästää huijarin ensimmäisellä yrityskerralla sisään 5 prosentin todennäköisyydellä. Toisella kerralla tunnistetuksi tuleminen vaatii ensin yhden hylkäyksen, joka tapahtuu 95 prosentin todennäköisyydellä ja virheellisen tunnistuksen toisella yrityskerralla. Näin todennäköisyys tälle tapahtumalle on 0.5 * 0.95 = 0.047, eli 4.7 prosenttia. Vastaavasti kolmen peräkkäisen hylkäyksen todennäköisyys huijarille on 0.95 * 0.95 * 0.95 = 0.857, eli 85.7 prosenttia. Järjestelmässä on kolme yrityskertaa puhenäytteen antamiseen, puhujat huijareita ja EER = 5% (taulukko 5.3).

Taulukko 5.1: Järjestelmä 1.

		Käyttäjä on todellisuudessa:	
		Aito	Huijari
Järjestelmä vastaa:	Aito	.95	.05
	Huijari	.05	.95

Taulukko 5.2: Järjestelmä 2.

Yrityskerrat:	Todennäköisyydet:	Tulokset:
Läpäisy:	.95	.95
Hylkäys, läpäisy	.05 * .95 =	.047
Hylkäys, hylkäys, läpäisy	.05 * .05 * .95 =	.002
Hylkäys, hylkäys, hylkäys	.05 * .05 * .05 =	.0001

Taulukko 5.3: Järjestelmä 3.

Yrityskerrat:	Todennäköisyydet:	Tulokset:
Läpäisy	.05	.05
Hylkäys, läpäisy	.95 * .05 =	.047
Hylkäys, hylkäys, läpäisy	.95 * .95 * .05 =	.045
Hylkäys, hylkäys, hylkäys	.95 * .95 * .95 =	.857

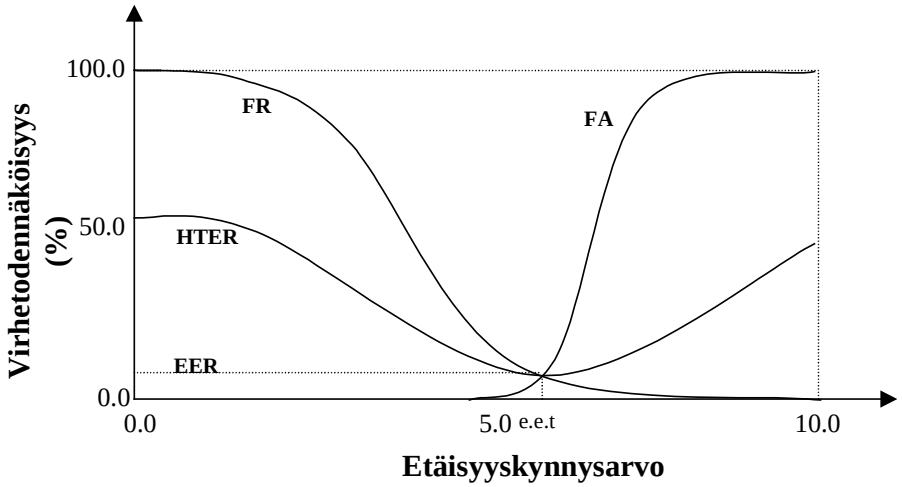
5.4 FA- ja FR-tyyppisten virheiden normalisoitu keskiarvo (HTER)

Jos todellisten puhujien ja huijareiden *a priori* todennäköisyys oletetaan yhtä suureksi, vastaavaksi päätöskynnysarvoksi tulee EER. Tällöin voidaan laskea *HTER*:n (Half Total Error Rate) minimiarvo. HTER määritellään seuraavasti [4, 28]:

$$HTER = \frac{P(FR) + P(FA)}{2}. \tag{5.2}$$

Kaavassa (5.2) *P(FR)* tarkoittaa FR-tyyppisen virheen todennäköisyyttä ja *P(FA)* puolestaan FA-tyyppisen virheen todennäköisyyttä. HTER:n voidaan ajatella intuitiivisesti tarkoittavan FA:n ja FR:n *normalisoituja* keskiarvoja. Normalisoinnilla tarkoitetaan tässä sitä että HTER kuvaa FA:n ja FR:n välistä *eroavaisuutta* niin, että se saa pienimmän arvon kohdassa, jossa FA ja FR leikkaavat. Toisin sanoen HTER-arvo on yhtenevä EER:n kanssa ainoastaan, jos järjestelmän puhujakohtaiseksi tai yleiseksi kynnysarvoksi on asetettu EER:ää vastaava arvo. Tätä EER:n mukaan asetettua kynnysarvoa voidaan kutsua *e.e.t*:ksi (equal error threshold) [4].

Koska FR:n ja FA:n arvojen muutokset eivät ole todellisuudessa jatkuvia samalla tavalla kuin kynnysarvoon tehtävät muutokset, on hyvin mahdollista että ei koskaan tule tilannetta, jossa FA olisi arvoltaan sama kuin FR [4]. Tällöin tarkkaa EER:ää ei voida laskea. HTER voidaan puolestaan laskea kaikilla FA:n ja FR:n arvoilla. Näin ajateltuna HTER tuo käytännöllisen mittarin tosielämän puhujanvarmennusjärjestelmien vertailuun EER:n tueksi. Kuvassa 5.3 on havainnollistettu FA:n, FR:n, HTER:n ja EER:n välistä suhdetta.



Kuva 5.3: FR, FA, EER ja HTER samassa kuvaajassa [4].

Kuten aiemmin todettiin, kun EER voidaan laskea, kaikki mainitut virhekuvaajat (FA, FR, EER ja HTER) leikkaavat kyseisessä pisteessä. Seuraavassa kohdassa tarkastellaan ROC-käyrää, jonka avulla voidaan havainnollistaa FA- ja FR-tyyppisten virheiden sekä EER:n välistä riippuvuutta.

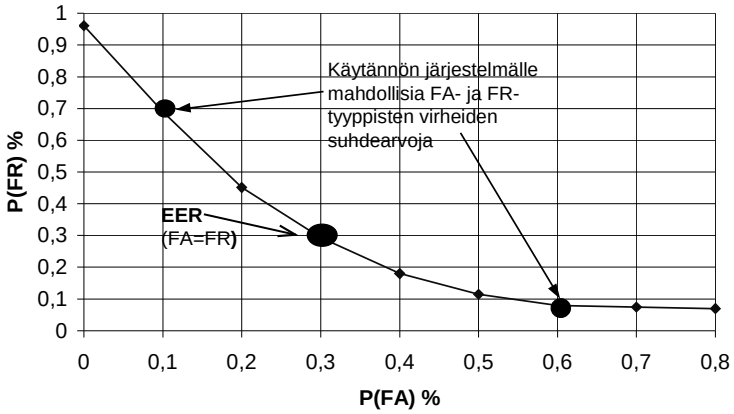
5.5 ROC-käyrä

Yritettäessä vähentää järjestelmästä joko FA- tai FR-tyyppisiä virheitä, aiheuttaa toisen väheneminen samalla toisen tyypin virheen kasvua [3]. Tästä voidaan päätellä että kyseiset virhetyypit ovat riippuvuussuhteessa toisiinsa. Tässä kohdassa esitetään tämän riippuvuussuhteen esittämiseen soveltuva menetelmä, *ROC-käyrä* (Receiver Operating Characteristic).

Kuten edellä todettiin, yhdistämällä virhekäyrien FA- ja FR-kuvaajat samaan koordinaatistotasoon ja etsimällä arvojen leikkauskohta, saadaan selville kyseisen puhujanvarmennusjärjestelmän keskimääräisen virhealttius, EER [11, 20, 21]. Tällaista virhearvojen suhdetta ilmaisevaa kuvaajaa sanotaan ROC-käyräksi. ROC-käyrä kuvaa FA-tyyppisten virheiden todennäköisyyttä suhteessa FR-tyyppisten virheiden todennäköisyyteen (kuva 5.4) [3]. Kuvaa luetaan seuraavasti: koordinaatiston *y*-akselilla on järjestelmän FR-tyyppisten virheiden todennäköisyydet ja *x*-akselilla FA-tyyppisten virheiden todennäköisyydet.

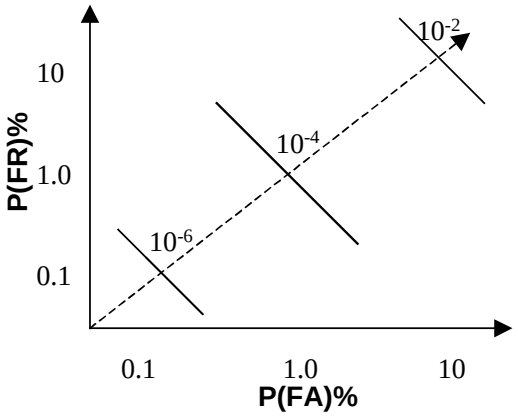
Kuvasta voidaan huomata esimerkiksi seuraavia järjestelmän luotettavuutta kuvaavia todennäköisyysarvoja: jos FA:ksi on asetettu 0.1 prosenttia, tulee järjestelmän FR-tyyppisten virheiden määräksi tällöin 0.7 prosenttia. Vastaavasti jos FR:ksi on asetettu 0.2 prosenttia, tulee järjestelmän FA-tyyppisten virheiden määräksi tällöin 0.37 prosenttia. EER, eli piste jossa FA:n ja FR:n kuvaajat leikkaavat, on 0.3 prosenttia.

Kuitenkin, kuten edellä jo pohdittiin, käytännön puhujanvarmennusjärjestelmissä käytetään harvoin suoraan EER:ää vastaavaa kynnysarvoa. Kuvaan onkin esimerkin vuoksi merkitty EER:n lisäksi kaksi käytännön järjestelmälle todennäköisempää FA- ja FR-tyyppisten virheiden keskiarvoa.



Kuva 5.4: ROC–käyrä, EER ja käytännön järjestelmälle mahdollisia FA- ja FR-tyyppisten virheiden suhdearvoja.

Kuvassa 5.5 on puolestaan piirretty hypoteettisia eli oletettuja ROC-käyriä [4]. EER:n todennäköisyys on piirretty kuvaan katkoviivalla asteikon diagonaalille. Poikittaiset -45° asteen kulmassa olevat viivat ilmaisevat EER:ltään poikkeavia järjestelmiä. Tarkemmat järjestelmät sijaitsevat lähempänä origoa kuin epätarkat järjestelmät. Kuvan tarkoituksena on osoittaa, että ROC-käyrän sijainti FR- ja FA-tyyppisten virheiden todennäköisyyksiä kuvaavalla asteikolla muuttuu suhteessa järjestelmän keskimääräiseen virhealttiuden, EER:n, kanssa.



Kuva 5.5: Hypoteettiset ROC-käyrät [4].

ROC-käyrät yleistyivät signaalinkäsittelytehtävissä 1950-luvulta lähtien. Aluksi niitä käytettiin apuna signaalinhavaitsemisjärjestelmien, kuten tutkien ja signaalinkäsittelysovelluksien kehittämisessä. 1970-luvulta lähtien ROC-käyriä on käytetty testimateriaalin diagnosointiin esimerkiksi lääketieteen ja psykologian alojen tutkimuksessa [21]. Myöhemmin ROC-käyrät ovat yleistyneet myös tietojenkäsittelytieteessä signaalinkäsittelytehtävien yhteydessä [21].

ROC-käyrä sopii hyvin analysointivälineeksi puhujanvarmennuksen lisäksi muihinkin binäärisiin luokitteluongelmiin, erityisesti tilanteisiin, joissa halutaan löytää syötedatan joukosta jokin harvoin esiintyvä ilmiö kuten signaalissa olevat yllättävät vääristymät. Esimerkiksi lääketieteessä tämän avulla voidaan todeta potilaan sydänkäyrässä olevia poikkeamia [21].

5.6 Varmennuksen suoritusaika (VT)

Puhujanvarmennusjärjestelmän käyttökelpoisuutta voidaan arvioida myös mittaamalla järjestelmän kokonaissuoritusaikoja (VT, Verification Throughput). Järjestelmän kokonaissuoritusaika koostuu seuraavista suorituksen osa-ajoista [23]:

- Ajasta, joka järjestelmällä kuluu hyväksyttävän ääninäytteen saamiseen. Ts. kuinka useasti käyttäjä joutuu uusimaan ääninäytteensä, eli kuinka monta *FR* -tyyppistä virhettä järjestelmä tekee ennen puhujan onnistunutta varmennusta.

- Ajasta, joka järjestelmällä kuluu hyväksyttävän ääninäytteen varmentamiseen. Tähän osavaiheen kuluva aika riippuu järjestelmän toteutusratkaisusta, järjestelmän käyttöympäristöstä, sekä siitä kuinka pitkälle järjestelmän toteutus on optimoitu.

- Ajasta, joka järjestelmällä kuluu vastauksen antamiseen käyttäjälle. Vastausaika saattaa olla pitkä etenkin järjestelmissä, jotka vastaavat käyttäjälle puhesyntetisaattorin välityksellä. Aikaa kuluu tällöin varsinaisen puhujan varmentamisen lisäksi puheen syntetisointiin.

5.7 Varmennusjärjestelmien vahvuudet

Biometriset tunnistusmenetelmät tarjoavat useita etuja verrattuna perinteisiin menetelmiin. Käyttäjän ei tarvitse muistaa salasanoja tai PIN-koodeja, eikä käyttää aikaa niiden syöttämiseen. Biometrinen tunnistaminen ehkäisee myös ulkopuolisten henkilöiden pääsemisen järjestelmään esimerkiksi avaimen varastamisen tai salasanan paljastumisen takia. Edellä mainittujen etujen lisäksi järjestelmää käyttävän organisaation turvallisuudesta vastaavan henkilöstön rutiininomaisen työn määrää voidaan vähentää otettaessa käyttöön biometrinen tunnistusjärjestelmä perinteisen tilalle/rinnalle [20]. Puhujanvarmennusjärjestelmien vahvuuksia ovat *edullisuus, turvallisuus* ja *helppokäyttöisyys* [23].

Edullisuus: Useimmat puhujanvarmennusjärjestelmät toimivat pääpiirteissään ohjelmistotasolla. Tästä syystä tunnistusjärjestelmän käyttöä varten ei tarvitse välttämättä hankkia uutta kallista teknologiaa. Automaattiset puhujanvarmennusjärjestelmät ovatkin edullisia verrattuna useimpiin muihin käyttäjän henkilökohtaisia piirteitä identifiointitehtävään käyttäviin (biometrisiin) järjestelmiin. Mahdollisia pakollisia, mutta suhteellisen edullisia investointeja voivat olla hyvälaatuinen mikrofoni ja digitaaliseen signaalinkäsittelyyn (DSP, Digital Signal Processing) tarvittava piiritason toteutus.

Helppokäyttöisyys: Varmennusjärjestelmän opetusvaiheen jälkeen käyttäjän tarvitsee usein ainoastaan toistaa järjestelmän vaatima lause ilman tarkempaa tietämystä varmennusjärjestelmän syvemmän tason toteutuksesta. Lisäksi käyttäjien voi ajatella kokevan puhujanvarmennusjärjestelmät esimerkiksi sormenjälkien- tai silmänrakenteen tunnistamiseen pohjautuvia järjestelmiä vähemmän henkilön yksityisyyttä loukkaaviksi. Tämä johtuu siitä, että toisin

kuin sormenjälkien tai silmänrakenteen tunnistamiseen pohjautuvissa järjestelmissä, puhujanvarmennuksessa ei tarvita fyysistä kontaktia käyttäjän ja varmennuksen tekevän laitteen välillä.

Turvallisuus: Toisen puhujan ääntä on erittäin vaikea matkia riittävän tarkasti mahdollista varmennusjärjestelmän väärinkäyttöä ajatellen [3, 23]. Tämä koskee niin puheäänen yleispiirteitä (murre, puhetyyli, puheen tunneperäiset yksityiskohdat) kuin matalammankin tason piirteitä (äänenkorkeus, äänen spektrin suuruusluokka, formanttien taajuudet). Nykyisen tietämyksen valossa ihmiset käyttävät eri piirteitä puhujan tunnistamiseen kuin koneet. Ihmiset käyttävät tunnistamiseen lähinnä yleisen tason piirteitä, kun koneet taas käyttävät lähes poikkeuksetta matalan tason piirteitä [34]. Toisin sanoen ihminen ja kone eivät tunnista puhujaa samojen asioiden perusteella. Vaikka imitoijan ääni kuulostaa ihmisen mielestä aivan samalta kuin imitoitavan henkilön oma ääni, kone osaa mitä todennäköisimmin nähdä eron näiden puheäänien välillä. Tätä asiaa ei ole vielä kuitenkaan tutkittu riittävästi.

5.8 Varmennusjärjestelmien heikkoudet

Vertailtaessa puhujanvarmennusjärjestelmiä muihin tunnistusjärjestelmiin täytyy arviointeihin ottaa odotettavissa olevien hyötyjen lisäksi mukaan kuitenkin myös järjestelmien mahdolliset heikkoudet. Puhujanvarmennusjärjestelmien heikkouksiin voidaan laskea *aikavaativuus*, *häiriöalttius* ja *epäluotettavuus* [23].

Aikavaativuus: Puhujanvarmennusjärjestelmän aikavaativuutta voidaan mitata sekä käyttäjältä järjestelmän käyttöön kuluvana aikana että järjestelmän suoritusnopeutena. Puhujanvarmennus vaatii usein enemmän käyttäjän aikaa kuin esimerkiksi tietokoneen näppäimistöltä kirjoitettavat PIN-koodit ja salasanat. Toisaalta jatkuva teknologian kehitys mahdollistaa suoritusnopeudeltaan entistä parempien puhujanvarmennusjärjestelmien kehittämisen.

Häiriöalttius: Puhujanvarmennusjärjestelmien tehokkuuteen vaikuttavat tekijät ovat erittäin herkkiä ulkopuolisille häiriöille, joita ovat esimerkiksi ympäristön melu, käytettävän tiedonsiirtokanavan häiriöt ja puhujanvarmennusjärjestelmään liitetyn mikrofonin laatu [23]. Käyttäjän muuttuvat ominaisuudet tuottavat myös ongelmia varmennusjärjestelmille; käyttäjän

ääni muuttuu iän myötä pitkällä aikavälillä, mutta vaikeampi tilanne on esimerkiksi flunssasta johtuva äänen muuttuminen.

Epäluotettavuus: Toisin kuin salasanoihin tai PIN-koodiin pohjautuvilla perinteisillä henkilönvarmennusjärjestelmillä, biometriseen tunnistamiseen pohjautuvilla varmennusjärjestelmillä ei ehkä koskaan päästä tilanteeseen, jossa huijari ei saisi huijattua järjestelmää [3]. Puhujanvarmennusjärjestelmiä voidaan edelleen huijata nauhoitetun puheen avulla, mutta toisaalta sisällöstä riippuvan järjestelmän käyttäminen vaikeuttaa tällaista huijaamista erittäin paljon [3].

Mainitut puhujanvarmennusjärjestelmien vahvuudet ja heikkoudet pätevät myös muillekin biometrisille tunnistusmenetelmille. Tosiasia on se, että biometrisillä tunnistusmenetelmillä ei ainakaan vielä nykypäivänä saavuteta aivan 100% tunnistusvarmuutta [1]. Taulukkoon 5.1 on listattu yhteenvetona biometrisiin tunnistusmenetelmiin liittyviä vahvuuksia ja heikkouksia.

Taulukko 5.1: Puhujantunnistuksen vahvuuksia ja heikkouksia [1].

VAHVUUDET:	HEIKKOUDET:
• Edullisuus • Helppokäyttöisyys • Turvallisuus	• Aikavaativuus • Häiriöalttius • Epäluotettavuus

6 PÄÄTÖSLOGIIKAN SUUNNITTELU JA TOTEUTUS

Puhujanvarmennusjärjestelmien sisältämä *päätöslogiikka* on järjestelmien toiminnan kannalta avainasia. Päätöslogiikalla tarkoitetaan sitä puhujanvarmennusjärjestelmän tarvitsemaa älykästä päättelyosaa, jonka avulla järjestelmä kykenee tekemään varmennuspäätöksen puhujan hyväksymisestä tai hylkäämisestä [4]. Tässä luvussa paneudutaan varmennuspäätöksen ja kynnysarvon merkitykseen ja niiden kehittämiseen ja määrittämiseen liittyviin matemaattisiin työkaluihin.

Tämä luku etenee kohdittain seuraavasti: kohdassa 6.1 pohdimme varmennuspäätöksen ja kynnysarvon merkitystä puhujanvarmennustehtävässä. Kohdassa 6.2 esittelemme kynnysarvon määrittelyyn liittyvät matemaattiset työkalut sekä määrittelemme *a priori* ja *a posteriori* kynnysarvot. Kohdassa 6.3 esittelemme uskottavuussuhteeseen pohjautuvan päätössäännön. Kohdassa 6.4 estimoimme päätössäännön parametrit ja käymme läpi normalisoitujen puhujajoukkojen merkityksen varmennustehtävässä. Lopuksi kohdassa 6.5 käymme läpi syitä kynnysarvon dynaamisen päivittämisen tarpeellisuuteen ja esittelemme erilaisia mahdollisuuksia kynnysarvon päivittämiseen.

6.1 Varmennuspäätös ja kynnysarvo

On ymmärretty, että varmennuspäätösprosessin sisältö kuuluu tiiviisti kaikkiin varteenotettaviin puhujantunnistuksen perusteisiin. Kysymyksen *kuinka varmennusjärjestelmämme päätöslogiikkaa voidaan parantaa* täytyy olla kiinteä osa järjestelmän arviointi- ja kehitysprosessia [4].

Pienetkin puutteet päätöslogiikassa vaikuttavat suoraan järjestelmän luotettavuuteen ja voivat romuttaa sitä kautta koko varmennusjärjestelmän käytettävyyden esimerkiksi turvallisuuspalvelutehtävissä. Puhujanvarmennusjärjestelmien päätöslogiikan suunnitteluun ja kehittämiseen käytetään samoja työkaluja kuin järjestelmien evaluointiinkin. Onnistuneen varmennuspäätöksen tekemiseen liittyy oleellisesti oikein valittu kynnysarvo. Tässä kohdassa perustellaan kynnysarvon merkitystä onnistuneessa päätöslogiikassa.

Puhujan hyväksymis/hylkäyspäätös voidaan ajatella päätöksen lisäksi myös todennäköisyyksien pohjalta tehtynä valintana [12]. Niinpä onkin paikallaan valottaa ensin hieman terminologian välisiä eroja.

6.1.1 Päätös vai valinta

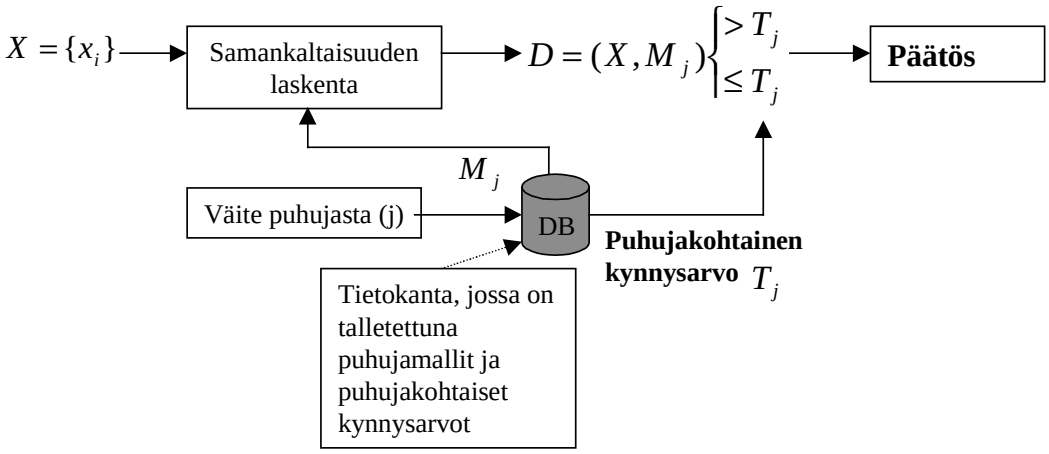
Usein termit *valinta* ja *päätös* ajatellaan kahtena eri asiana. Valinta tehdään yleensä intuitiivisesti ajattelematta ongelmaa välttämättä läpikotaisin. Päätöksen tekemisen ajatellaan puolestaan usein pohjautuvan vakavaan ja perusteltuun harkintaan. Kuitenkaan jyrkän eron tekeminen näiden termien välille ei ole välttämättä järkevää. Useissa ongelmatapauksissa on mahdotonta sanoa, kumpaan kategoriaan tehtävä selvästi kuuluu: valinnan vai päätöksen kategoriaan. Ihmiset tekevät valintansa usein jonkin asteinen miettimisen ja asiaan tutustumisen pohjalta [12].

Yhtenä tutkielman aihepiiristä eroavana käytännön esimerkkinä tehtävän sijoittamisen vaikeudesta mainittuihin kategorioihin tosielämässä voidaan mainita ihmisen henkilökohtaisen uskonnon suuntautumisen. Jollekin ihmiselle uskonnon valinta on vakaasti harkittu ja ainakin omassa mielessä perusteltu päätös. Toiselle taas pikemminkin valinta, joka ei pohjaudu vakaaseen ja pitkään harkintaan.

Toisena esimerkkinä voidaan ajatella tupakoinnin lopettamista [12]. Jotkut ihmiset tekevät tupakoinnin lopettamispäätöksen vakaan harkinnan ja hyöty/haitta-analyysin pohjalta, toiset taas nopeasti valintana perustelematta asiaa itselle sen tarkemmin. Kun ihminen tunnistaa puhujaa puhujan äänen perusteella, hän tekee päätöksen jonkin asteisen harkinnan ja mietinnän perusteella, verraten kuulemaansa puhetta muistissaan oleviin puhujamalleihin. Myös automaattisessa puhujanvarmennustehtävässä on kyse päätöksestä. Kone tekee varmennuspäätöksen matemaattisesti perusteltujen menetelmien avulla. Näin on perusteltua käyttää puhujanvarmennustehtävän tuloksesta puhuttaessa mieluummin termiä *päätös* termin *valinta* sijasta.

6.1.2 Kynnysarvon merkityksestä

Varmennuspäätöksen tekeminen perustuu varmennusta haluavan puhujan piirrevektoreiden ja väitetyn puhujan mallin väliseen vertailuun ja varmennuspäätös tehdään vertailussa todetun samankaltaisuuden pohjalta [20]. Järjestelmän päätöslogiikkaan asetettu *kynnysarvo* kertoo järjestelmälle, kuinka samankaltaisia puhujamallien tulee olla hyväksytyn varmennustuloksen saamiseen, tai toisaalta kuinka erilaisia puhujamallien tulee olla varmennuksen hylkäämiseen (kuva 6.1).



Kuva 6.1: Kynnysarvon merkitys varmennuspäätöksessä.

Oikean kynnysarvon määrittely on tärkeä osa järjestelmän päätöslogiikkaa ja varmennuspäätöstä. Varmennuspäätöksenteko on luokiteltu puhujanvarmennusjärjestelmien historiassa yksinkertaiseksi ongelmaksi, mutta viimeaikoina kyseinen ongelma on huomattu erittäin haastavaksi osaksi tosielämän henkilönvarmennusjärjestelmän suunnittelua [4].

Kynnysarvon määrittäminen vaatii ohjelmoijalta tietämystä järjestelmän käytön sovelluskohdealueesta [20]. Kynnysarvon sallittuun mittapoikkeamaan eli *toleranssin* vaikuttaa nimittäin käytettävän varmennusjärjestelmän yleisen tarkkuuden lisäksi tehtävä, johon järjestelmä on liitetty. Kynnysarvo on yleensä tiukempi esimerkiksi teollisuuslaitoksen ovenavausjärjestelmässä kuin kotitietokoneeseen liitetyssä sisäänkirjautumisjärjestelmässä [23].

Jos puhujamalli on toteutettu stokastisiin malleihin pohjautuvan päätöslogiikan avulla, varmennustehtävä on itse asiassa todennäköisyyksien vertailua [2]. Puhuja hyväksytään, mikäli aidon puhujan todennäköisyys on suurempi kuin huijarin todennäköisyys. Sapluunamalleissa,

erityisesti VQ:ssa [35] lasketaan sen sijaan usein *etäisyysarvoja* todennäköisyyksien sijasta. Päätössäännöt kääntyvät ikään kuin "nurinniskoin" käytettävästä täsmäysmenetelmästä riippuen. Seuraavaksi käsittelemme varmennuspäätöksen tekemiseen ja kynnysarvon asettamiseen tarvittavia matemaattisia työkaluja.

6.2 Matemaattisia työkaluja

Kynnysarvon määrittäminen, samoin kuin koko automaattinen puhujanvarmennusjärjestelmä, perustuu malleja ja todennäköisyysvertailuja sisältävien algoritmien käyttöön. Tässä kohdassa käymme läpi kynnysarvon määrittämiseen liittyvät matemaattiset työkalut. Kohdassa 6.2.1 määrittelemme *a priori* ja *a posteriori* kynnysarvot. Kohdassa 6.2.2 käsittelemme Bayesin päätössääntöä. Kohdassa 6.2.3 puolestaan käymme läpi normaalijakauman periaatteet ja sen merkityksen varmennusjärjestelmän kynnysarvon määrityksessä.

6.2.1 *A priori*- ja *a posteriori* kynnysarvo

Kynnysarvon määrittely voidaan määrittää karkeasti kahdella eri tavalla: joko opetusvaiheessa kerätyn opetusdatan pohjalta tai estimoimalla järjestelmän FA- ja FR-tyyppisten virheiden todennäköisyyksiä [2, 4]:

1. *a priori* kynnysarvo määritellään opetusvaiheessa kerätyn opetusdatan pohjalta. Tämä kynnysarvon määrityskeino mahdollistaa helposti toteutettavan keinon todellisen puhujanvarmennusjärjestelmän toteuttamiseen.

2. *a posteriori* kynnysarvo määritellään usein etsimällä EER. Toisin sanoen *a posteriori* –kynnysarvoa etsittäessä joudutaan estimoimaan ensin FA- ja FR-tyyppisten virheiden todennäköisyydet luokittelemalla osa opetusdatasta. *A priori* –tapauksessa tätä luokittelua ei tarvitse tehdä, vaan kynnysarvo määrätään suoraan opetusdatan tilastollisten ominaisuuksien perusteella.

A posteriori-kynnysarvo on yleensä järjestelmäkohtaisuuden lisäksi myös käyttäjäkohtainen [20]. Tämä tarkoittaa että jokaiselle käyttäjälle joudutaan määrittelemään kynnysarvo järjestelmän opetusvaiheessa.

A posteriori kynnysarvon käyttö mahdollistaa puhujanvarmennusjärjestelmien välisen vertailun tilanteessa, jossa verrataan järjestelmien kykyä tietyn puhujamallin toteuttamiseen tietystä samasta opetusdatasta. Vaikka tällainen vertailujärjestelmä mahdollistaisi objektiivisen tarkastelun varmennusjärjestelmän puhujanmallinnuskyvyistä määrätyn opetusdatan pohjalta, sen toteuttaminen siten, että se toimisi järkevässä suoritusajassa, olisi kuitenkin kaiken kaikkiaan erittäin haastava tehtävä. Tämä johtuu algoritmin vaatimasta suuresta laskentatehosta [4]. Seuraavaksi käydään läpi Bayesin päätössäännön ja normaalijakauman merkitys varmennusjärjestelmän kynnysarvon suunnittelussa.

6.2.2 Bayesin päätössääntö

Puhujanvarmennusjärjestelmän kynnysarvon määrittelyssä käytetään usein apuna *Bayesin päätössääntöä* (BDR, Bayesian Decision Rule) [4, 7]. Oletetaan seuraavat merkinnät: X ilmaisee puhujaa ja $\overline{X}$ edustaa huijaria. *x*:lla merkitään kyseisen puhujan todennäköisyysmallia ja $\overline{x}$:lla puolestaan merkitään *vastinjoukon* (cohort set) edustajan puhujamallia. Vastinjoukko tarkoittaa tässä sitä varmennettavasta puhujasta eroavien puhujien joukkoa, johon kuuluvat varmennettavaa puhujaa ”tarpeeksi paljon muistuttavat puhujat” [4, 20]. Puhujajoukko voi olla myös *eroava joukko*, johon kuuluvat varmennettavasta puhujasta ”tarpeeksi paljon eroavat” puhujat.

Sekä puhuja X että vastinjoukon edustaja $\overline{x}$ voidaan kuvata todennäköisyysmallien avulla. Puhujan hyväksymispäätöstä kuvataan tästä lähtien merkinnällä $\hat{X}$ ja puhujan hylkäämistä merkinnällä $\hat{\overline{X}}$. Bayesin päätöslogiikan mukaan optimaalinen päätös tehdään minimoimalla alla oleva kustannusfunktio [7, 28]:

$$C = C_{(\hat{X}|\overline{X})} * p_{\overline{X}} * P(\hat{X} \mid \overline{X}) + C_{(\hat{\overline{X}}|\overline{X})} * p_X * P(\hat{\overline{X}} \mid X). \tag{6.1}$$

Kaavassa (6.1) p_X tarkoittaa *a priori*-todennäköisyyttä sille, että puhuja on väitetty puhuja. Vastaavasti $p_{\overline{X}}$ tarkoittaa *a priori*-todennäköisyyttä sille, että puhuja X on vastinjoukon edustaja. *A priori*-todennäköisyydet määritetään opetusdatan tilastollisen analyysin perusteella. $P(\hat{X} \mid \overline{X})$ ja $P(\hat{\overline{X}} \mid X)$ ovat FA:n ja FR:n *a posteriori* todennäköisyydet. Jos merkinnät muutetaan sanallisiksi, $P(\hat{X} \mid \overline{X})$ tarkoittaa todennäköisyyttä sille että puhuja varmennetaan mutta puhuja onkin huijari. Vastaavasti $P(\hat{\overline{X}} \mid X)$ tarkoittaa todennäköisyyttä sille että puhujaa ei hyväksytä vaikka puhuja oli väitetty puhuja. FA ja FR käytiin läpi tarkemmin tutkielman viidennessä luvussa.

Kaavassa (6.1) $C_{(\hat{X}\mid\overline{X})}$ ja $C_{(\hat{\overline{X}}\mid\overline{X})}$ ilmaisevat kustannuksia todellisen puhujan hylkäämiselle (FR) ja huijarin hyväksymiselle (FA). Kustannukset ovat käyttäjän antamia valinnaisia parametrejä, joiden avulla käyttäjä voi vaikuttaa väärän varmennustuloksen vakavuuden painotukseen kustannusfunktion tuloksessa [10]. Käyttäjä voi asettaa kyseisille kustannuksille arvot 1, jolloin niitä ei huomioida kustannusfunktiota laskettaessa.

Funktion (6.1) minimointi kustannuksen suhteen johtaa tunnettuun *Bayesin päätössääntöön*, joka on luokitteluvirheen todennäköisyyden mielessä optimaalinen [7]. Kyseisen päätössäännön avulla voidaan havainnollistaa myös kynnysarvon merkitystä puhujanvarmennustehtävässä seuraavasti.

$$\begin{cases} P(X \mid x) = \dfrac{P(X \mid x)}{P(X \mid \overline{x})} \geq R, & \textit{puhuja hyväksytään} \\ P(X \mid x) = \dfrac{P(X \mid x)}{P(X \mid \overline{x})} < R, & \textit{puhuja hylätään} \end{cases} \tag{6.2}$$

Päätössäännössä (6.2) $P(X \mid x)$ tarkoittaa puhujan *todennäköisyyden tiheysfunktiota* (*PDF*, Probability Density Function). Tästä lähtien tutkielmassa käytetään tästä arvosta lyhennettä *PR* (PDF Ratio). $P(X \mid \overline{x})$ tarkoittaa PDF:n arvoa ei-puhujien (huijarien) jakaumassa. R tarkoittaa puolestaan Bayesin todennäköisyyskynnysarvoa [32].

Puhuja hyväksytään päätössäännön (6.2) mukaan silloin, jos todennäköisyys tapahtumien $P(X \mid x)$ ja $P(X \mid \overline{x})$ välillä on suurempi tai yhtä suuri kuin vaadittu todennäköisyyskyn-

nysarvo *R*. Puhuja vastaavasti hylätään jos todennäköisyys tapahtumien välillä on pienempi kuin todennäköisyyskynnysarvo *R*. Kynnysarvo *R* määritellään seuraavasti [32]:

$$R = \frac{C_{(\hat{X}|\bar{X})}}{C_{(\hat{\bar{X}}|X)}} \cdot \frac{P(X \mid x)}{P(X \mid \bar{x})} \tag{6.3}$$

Kynnysarvon määritelmästä (6.3) voidaan havaita, että optimaalinen kynnysarvo riippuu ainoastaan suhdeluvusta FA:n ja FR:n välillä, samoin kuin *a priori* todennäköisyys huijareiden ja todellisiin puhujien välillä. Tosielämän puhujanvarmennusjärjestelmissä kustannusfunktiot, $C_{(\hat{X}|X)}$ ja $C_{(\hat{\bar{X}}|X)}$, voidaan laskea järjestelmän kohdealueesta olevan etukäteistiedon perusteella [4].

Etukäteistiedolla tarkoitetaan esimerkiksi järjestelmän toiminta-alueella vallitsevaa kaikua tai jotain muuta varmennusta haittaavaa tekijää [4]. Tilanteessa, jossa etukäteistietoa ei ole, edellä mainitut kustannukset asetetaan yhtä suuriksi, jolloin niiden vaikutus on yhtä suuri. Siten optimaalinen kynnysarvo riippuu ainoastaan *a priori* todennäköisyyssuhteesta [4].

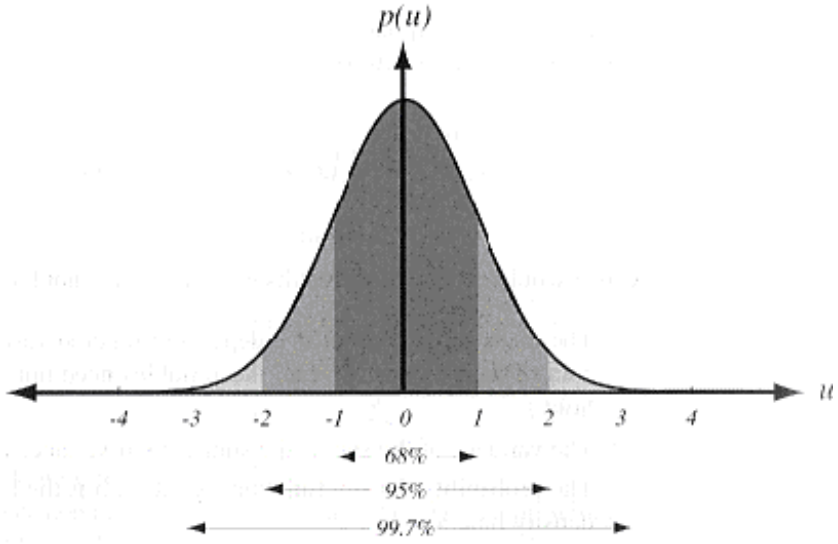
6.2.3 Normaalijakauma

Jos puhenäytteen kokonaispituus on tarpeeksi suuri, puhenäytteen aiemmin määritellyn uskottavuussuhteen voidaan otaksua noudattavan keskeisen raja-arvolauseen (CLT, Central Limit Theorem) mukaan *Gaussin jakaumaa* eli *normaalijakaumaa* [15, 28]. Gaussin jakauma määritellään yksiulotteisessa tapauksessa seuraavasti [7]:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2((x-\mu)^2/\sigma^2)}. \tag{6.4}$$

Gaussin jakauma voidaan määritellä täydellisesti siihen kuuluvan kahden parametrin avulla: keskiarvoparametrillä μ ja varianssilla σ^2. Usein merkitään: $p(x) \sim N(\mu, \sigma^2)$. Ehtolause luetaan: ”satunnaismuuttuja *x* noudattaa normaalijakaumaa, jolla on keskiarvo μ ja varianssi σ^2”. Varianssin neliöjuurta σ sanotaan jakauman *keskihajonnaksi*.

Gaussin jakauman kuvaajan huippukohta sijaitsee pisteessä $x = \mu$ ja kuvaajan "leveys" on sitä suurempi, mitä suurempi jakauman hajonta on [7]. Kaksi Gaussin jakauman kuvaajaa ovat siis aina toisiinsa nähden samanmuotoisia, vain kuvaajan korkeus ja leveys vaihtelevat. Kuvassa 6.2. oleva tummin sektori kuvaa jakaumaa $p(u) \sim N(0,1)$, eli: ”tapahtuman *u* toteutumisen todennäköisyys sijoittuu 68 prosentin todennäköisyydellä Gaussin jakauman alueelle -1..1”. Vastaavasti $p(u) \sim N(0,2)$ = 95 prosenttia ja $p(u) \sim N(0,3)$ = 99.7 prosenttia.



Kuva 6.2: Yksiulotteisen Gaussin jakauman kuvaaja [7].

6.3 Hypoteesitestaus

Tässä kohdassa esittelemme uskottavuussuhteeseen (likelihood ratio) pohjautuvan päätössäännön ja perustelemme kyseisessä päätössäännössä olevien parametrien tarkoituksen.

6.3.1 Uskottavuussuhde

Kuten aiemmin todettiin, PR (PDF Ratio) –puhujan todennäköisyyden tiheysfunktiot lasketaan PDF:ien estimaateista. Koska jakaumia ei kuitenkaan voida tarkasti tietää, vaan ne on estimoitava äärellisestä määrästä opetusdataa, estimointi ei vastaa tarkasti todellista

puhujien jakaumaa. Tämän takia on usein tarpeen asettaa kynnysarvo PR-testejä mukaillen, mutta ottaen kuitenkin huomioon myös mahdollinen (ja todennäköinenkin) epätasapaino varmennettavan puhujan mallin ja muun kerätyn opetusdatan välillä [28]. Epätasapainolla tarkoitetaan tässä sitä, että eri puhujille voidaan kerätä opetusdataa erilainen määrä. Käytännössä on huomattu että varmennustulos on tarkempi kun opetusdataa on käytössä enemmän [3, 4, 19, 23]. Tästä johtuen kaavan (6.2) PR-testit muutetaan *uskottavuussuhdetesteiksi* (*LR*, Likelihood Ratio):

$$\begin{cases} LR_{\chi}(X) = \dfrac{\hat{P}_{\chi}(X)}{\hat{P}_{\bar{\chi}}(X)} > \Theta_{x}(R), & \textit{puhuja hyväksytään} \\ LR_{\chi}(X) = \dfrac{\hat{P}_{\chi}(X)}{\hat{P}_{\bar{\chi}}(X)} < \Theta_{x}(R), & \textit{puhuja hylätään} \end{cases} \tag{6.5}$$

Yllä olevassa uskottavuussuhteeseen pohjautuvassa päätösvertailussa $\hat{P}_{\chi}$ ja $\hat{P}_{\bar{\chi}}$ merkitsevät puhujamallien uskottavuusfunktioita varmennettavalle puhujalle ja huijarille. $\Theta x(R)$ on puhujan ja asetettujen kustannusten suhteen riippuvainen kynnysarvo [28].

6.4 Päätöskaavan parametrien estimointi

Tässä kohdassa käymme läpi tarkemmin aiemmin mainitun uskottavuustesteihin pohjautuvan päätöskaavan (6.5) ja estimoimme sen parametrit. Vastinpuhujan todennäköisyyden estimointi on ongelmallista. Tässä kohdassa tähän ongelmaan etsitään ratkaisua normalisoitujen puhujajoukkojen avulla. Väitetyn puhujan itsensä todennäköisyyden estimointi on suoraviivainen tehtävä ja se voidaan laskea suoraan vertaamalla testidataa väitetyn puhujan malliin.

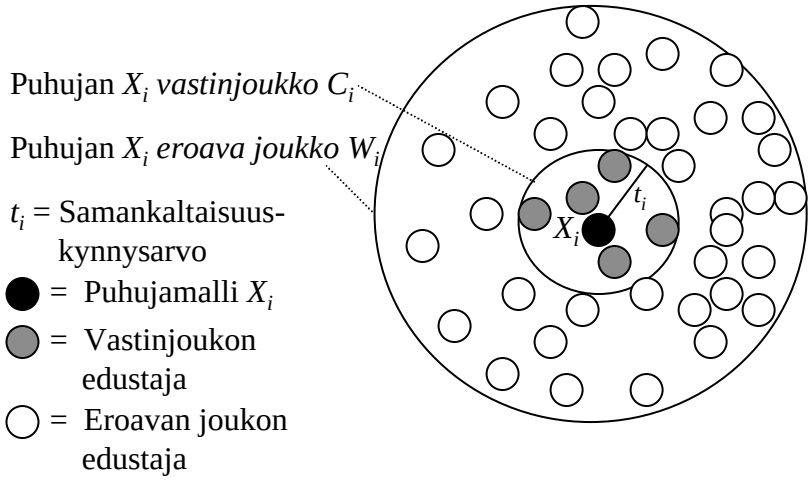
6.4.1 Normalisoidut puhujajoukot

Kuten jo aiemmin mainittiin perinteisessä puhujanvarmennustehtävässä tunnistusvaiheessa kerättävää puhenäytettä verrataan ainoastaan varmennettavan puhujan puhujamalliin ja varmennuspäätös tehdään vertaamalla samankaltaisuusarvoa kynnysarvoon. Tällaista varmennustehtävää kutsutaan *normalisoimattomaksi* puhujanvarmennustehtäväksi.

Normalisoitujen puhujajoukkojen (normalized speaker set) idea perustuu tulosarvojen normalisointiin (score normalization). Tulosarvojen normalisoinnin tarkoituksena on tässä parantaa puhujanvarmennuksen luotettavuutta tekemällä varmennuspäätös muidenkin kuin varmennettavan puhujan mallin perusteella. Normalisoidut puhujajoukot jaetaan *eroaviin joukkoihin* ja *vastinjoukkoihin* [4, 28].

Eroavat joukot (world model, speaker backround model) sisältävät varmennettavan puhujan äänimallista riittävän paljon eroavat puhujamallit, kun taas vastinjoukot (cohort model) sisältävät puolestaan varmennettavan puhujan äänimallista vähän eroavien puhujien puhujamallit [11]. Formuloidaan nämä käsitteet seuraavasti. Olkoon puhujatietokannan mallit $\{X_1, X_2, .., X_n\}$. Puhujan i eroava joukko on $W_i = \{X_j \mid D(X_i, X_j) \geq t_i\}$ ja vastinjoukko $C_i = \{X_j \mid D(X_i, X_j) < t_i\}$.

Puhujamalli X_j siis on kelvollinen joukkoon *Wi*, jos kyseinen puhujamalli eroaa varmennettavan puhujan puhujamallista X_i vähintään kynnysarvon t_i verran. Vastaavasti puhujamalli X_j on kelvollinen joukkoon *Ci*, jos kyseinen puhujamalli eroaa varmennettavan puhujan puhujamallista X_i vähemmän kuin kynnysarvon t_i verran. Kuvassa 6.3 on havainnollistettu puhujan X_i eroavien puhujien joukkoa W_i ja vastinjoukkoa C_i.



Kuva 6.3: Puhujan X_i eroavien puhujien joukko W_i ja vastinjoukko C_i.

Eroavia joukkoja on mainittu käytettävän usein järjestelmissä, joissa puhujamallit on luotu stokastisilla menetelmillä kuten HMM (Hidden Markov Model) [4, 22], GMM (Gaussian Mixture Model) ja neuroverkkojärjestelmät [29]. Vastinjoukkoja on puolestaan käytetty yleisesti sapluunamalleihin perustuvissa varmennusjärjestelmissä kuten vektorikvantisointijärjestelmät (VQ) ja Dynamic Time Warping (DTW) -järjestelmät [5, 11]. Tässä tutkielmassa keskityn erityisesti vastinjoukkoihin, sillä niitä käytetään kirjallisuudesta saatuun käsitykseeni pohjautuen eroavia joukkoja yleisemmin varmennustehtävässä.

6.2.4 Vastinjoukkojen muodostaminen

Vastinjoukon muodostamisen (ICN, Impostor Cohort Normalisation) teorian esittelivät ensimmäisenä Li ja Porter vuonna 1988 [18]. Sen jälkeen sitä on käytetty menestyksekkäästi useissa puhujantunnistusjärjestelmissä [5]. Perinteisessä normalisoimattomassa puhujanvarmennuksessa tunnistusvaiheen puhenäytettä verrataan ainoastaan väitetyn puhujan malliin. Vastinjoukkoihin perustuva varmennustehtävä muistuttaa tunnistustehtävää siinä mielessä, että kummassakin tunnistusvaiheen puhenäytettä verrataan sekä varmennettavan puhujan, että muiden puhujien malleihin [28].

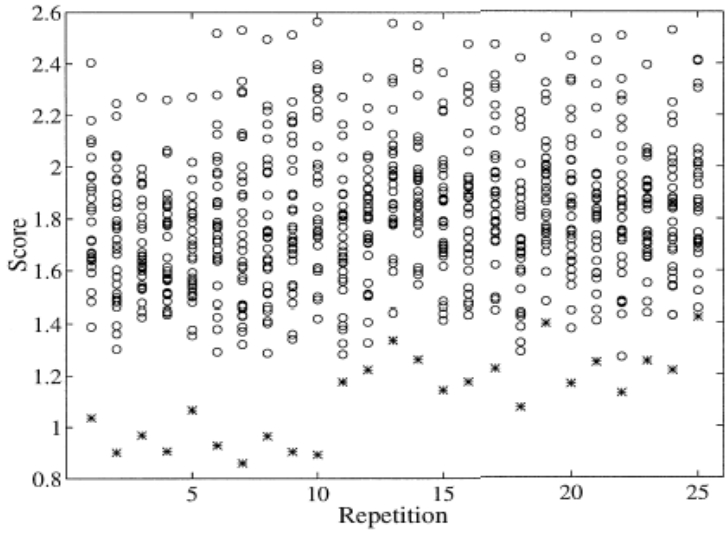
Puhenäytteen ja huijaripuhujien mallien välisten vertailujen tulosarvoja käytetään varmennettavan puhujan mallin normalisoinnissa. Varmennuspäätös tehdään tällöin vertaamalla kynnysarvoa normalisoinnista saatuihin tulosarvoihin [18]. Kynnysarvo voi olla joko puhujakohtainen tai yleinen, eli kaikille puhujille sama. Varmennuksen luotettavuus on yleensä parempi puhujakohtaisia kynnysarvoja käyttävissä järjestelmissä [11]. Tässä tutkielmassa keskitytään lähinnä puhujakohtaisten vastinjoukkojen muodostamiseen.

Vastinjoukon muodostamisen helpottamiseksi vastinpuhujat on lajiteltava sen mukaan, kuinka paljon he muistuttavat varmennettavaa puhujaa (impostor ranking), eli kuinka samanlaisia puhujamallit ovat [11]. Tämän lajittelun pystyy tekemään helpommin jälkikäteen (*a posteriori*), kun kaikkien puhujien opetusdata on kerätty ja vertailu koko huijarijoukon välillä on mahdollista [11]. Vaikka vastinpuhujien lajittelu on tärkeä osa vastinjoukon muodostamista, se on usein vaikea tehtävä käytännön puhujanvarmennusjärjestelmissä. Dynaamisen vastinjoukon päivittämisen sisältävissä järjestelmissä uuden puhujan mukaan ottaminen tietokantaan johtaisi raskaisiin ja aikaa vieviin vastinjoukkojen päivityksiin [5, 11].

Varmennettavan puhujan vastinjoukkoista saataisiin mahdollisimman tarkkoja ottamalla vastinpuhujien lajitteluun mukaan kaikki vastinpuhujien tunnistusvaiheessa antamat ääninäytteet ja vertaamalla varmennettavan puhujan mallia niihin kaikkiin.

Toisaalta vastinpuhujien lajittelu voidaan tehdä myös *a priori* opetusdatan keräämisen yhteydessä. Tällöin lajittelu tehdään testaamalla todellisen puhujan mallia vastinpuhujan puhujamallin kanssa. Tämäkin saattaa kestää kauan, riippuen varmennettavan puhujan puhujamallia vastaavien huijaripuhujamallien määrästä [11].

Kuvassa 6.4 on esimerkki tulosarvojen tyypillisestä jakaumasta testissä, jossa verrataan puhenäytettä puhujan omaan ja huijaripuhujan puhujamalliin [11]. Kuvassa on näytetty 25 kertaa toistetun sanan tulosarvojen jakaumat verratessa puhenäytettä varmennettavan puhujan malliin ja huijareiden puhujamalleihin. Jokaisella toistokerralla on kyseessä eri puhuja ja kaikilla puhujilla on sama määrä vastinpuhujia. Kuvassa "o" tarkoittaa huijaritulosarvoa ja "*" tarkoittaa oikean puhujan tulosarvoa. Kuvasta voidaan helposti huomata ero varmennettavan puhujan ja huijarin välisten vertailujen jakaumassa. Merkkien ”o” vaihteluväli on kuvasta päätellen noin välillä 1.3 - 2.6, kun taas merkeillä ”*” jakauma sijoittuu noin välille 0.9 - 1.4. Mitä pienempi arvo *:lla on, sitä tarkemmin puhuja varmennetaan ja mitä kauempana ”*”-merkki on ensimmäisestä ”o”-merkistä, sitä enemmän puhuja eroaa eniten häntä vastaavasta huijarista.



Kuva 6.4: Esimerkki tulosarvojen jakaumasta varmennustehtävässä [11].

6.5 Kynnysarvon ja mallin päivittäminen

Optimaalisessa varmennusjärjestelmässä puhujan mallia kehitetään jatkuvasti järjestelmän normaalin käytön yhteydessä [19]. Nykypäivän puhujanvarmennusjärjestelmiltä odotetaankin kynnysarvojen automaattista päivittämistä järjestelmän normaalin päivittäisen käytön yhteydessä [4]. Automaattisen puhujantunnistuksen tutkimus koostui alkuaikoina suurelta osin oleellisten puheen piirteiden selvittämisestä [4]. Viime aikoina tutkimus on siirtynyt selvittämään myös ongelmaa, kuinka puhujanvarmennusjärjestelmän luotettavuus ja käytettävyys taataan puhujan äänen muutoksista huolimatta.

Kynnysarvon asettaminen on ongelmallista, koska arvon määrittäminen tehdään usein pelkästään opetusvaiheessa kerätyn puhedatan perusteella [19]. Kynnysarvon määrittäminen etukäteen (*a priori*) määrää samalla pitkälle järjestelmän tehokkuuden. *A priori* kynnysarvon määritys sopii hyvin sisällöstä riippuvaan varmennusjärjestelmään, koska se varmistaa optimaalisen varmennustehokkuuden opetusvaiheessa käytetylle opetusdatalle [19].

Sitä vastoin sisällöstä riippumattomia varmennusjärjestelmiä ajatellen *a priori* kynnysarvon määritys ei usein tuota optimaalista varmennustulosta. Tämä johtuu siitä, että *a priori*-kynnysarvoon pohjautuva puhujanvarmennusjärjestelmä vaatii tehokkaasti toimiakseen saman puhenäytteen sekä opetusvaiheessa että tunnistusvaiheessa [19]. Ongelmia kynnysarvon määrityksessä aiheuttavat myös puhujan nopeat ja hitaat äänen muutokset (ikääntyminen, flunssa) [4].

Perusideana useimmissa olemassa olevissa puhujamallin päivittämiskeinoissa on uudelleen estimoida ja yhdistää olemassa oleva puhujan opetusdata ja uusi opetusdata. Tämän jälkeen luodaan uusi puhujamalli ja kynnysarvo tämän uuden yhdistetyn datan perusteella [4]. Kun otetaan huomioon, että puhujatietokanta voi sisältää tuhansia puhujia, on tällaiseen koko puhujamallin korvaukseen pohjautuva kynnysarvon päivittäminen erittäin raskas operaatio [4].

Tähän ongelmaan pohjautuen Chen [4] esittelee oman, reaaliaikaiseen kynnysarvon päivittämiseen kykenevän päivitysmenetelmän. Esittelen tässä menetelmän menemättä kovin tarkasti sen yksityiskohtiin. Menetelmän idea perustuu siihen, että se tarvitsee vanhan opetusdatan tilastollisen estimaatin. Nämä estimaatit määritellään menetelmässä reaaliajassa, ennen uuden kynnysarvon määritystä ja uuden opetusdatan vastaanottamista.

$$\overline{\mu}_{n+1} = \frac{n\tilde{\mu}_n + \tilde{x}_{n+1}}{n+1} \tag{6.6}$$

ja

$$\tilde{\sigma}_{n+1} = \frac{n(n+1)\tilde{\sigma}^2 + n(\tilde{x}_{n+1} - \tilde{\mu}_n)^2}{(n+1)^2} \tag{6.7}$$

Kaavoissa (6.6 ja 6.7) $\overline{\mu}_n$ ja $\tilde{\sigma}_n$ ovat estimoituja toisen kertaluvun momentteja, joita on käytetty vanhan kynnysarvon asettamiseen. $\overline{\mu}_{n+1}$ ja $\tilde{\sigma}_{n+1}$ ovat puolestaan estimaatteja uuden opetusdatan lisäämisen jälkeen. Perinteisempiin kynnysarvon määritysmenetelmiin verrattuna menetelmä tarjoaa kehittäjänsä mukaan [4] seuraavia etuja: 1. aiemman opetusdatan sijaan tarvitaan ainoastaan kyseisen datan tilastolliset estimaatit (tilan säästö), 2. kynnysarvo voidaan määrittää reaaliajassa (tässä täytyy ottaa kuitenkin huomioon, että uuden kynnysarvon laskenta vaatii tälläkin menetelmällä paljon konetehoa), 3. tämä menetelmä tuo uuden keinon puhujan äänen muutosten huomioimiseen varmennuspäätöksenteossa [4].

7 KOKEELLISET TULOKSET

Tutkielman tekemiseen kuului yhtenä osana kokeellinen osuus. Kokeellisessa osassa testasin TIMIT-tietokannassa olevien puhujamallien ja testidatan avulla itse tehdyn puhujanvarmennusjärjestelmän FA-, FR-, HTER ja EER-arvojen todennäköisyyksiä sekä järjestelmän suoritusaikoja. Lisäksi tutkittiin järjestelmän eri parametrien vaikutusta näiden virhemittareiden arvoihin. Tämä luku etenee kohdittain seuraavasti. Kohdassa 7.1 esittelen testiajoissa käyttämäni järjestelmän. Kohdassa 7.2 käyn läpi testiajoissa käytetyn puhedatan. Kohdassa 7.3 esittelen testiajojen tulokset. Lopuksi kohdassa 7.4 pohdin saatujen tulosten järkevyyttä ja keinoja tulosten parantamiseksi.

7.1 Testauksessa käytetty järjestelmä

Tässä kohdassa käyn läpi tutkielman kokeellisessa osuudessa käytetyn järjestelmän `verify_err` toiminnan yleisellä tasolla. Järjestelmä on kokeellinen puhujanvarmennusjärjestelmä, joka laskee puhujamallien ja testidatan perusteella järjestelmän FA- ja FR-arvot. EER-arvot laskin etsimällä FR- ja FA-arvojen kuvaajien leikkauskohdat. Ajoja suoritettiin sekä Windows NT4.0- että UNIX-käyttöjärjestelmässä SunOS 5.7 (Solaris 7). UNIX-kone on malliltaan Sun Enterprise 450. Koneessa on 4 kpl 400 MHz UltraSPARC2 prosessoreita ja 2GB keskusmuistia.

Järjestelmä laskee FR- ja FR-tyyppiset virhetodennäköisyydet seuraavalla tavalla.

ALGORITMI 7.1: Varmennusjärjestelmän FR- ja FA-tyyppisten virhetodennäköisyyksien laskeminen.

```
FUNCTION verify_err(INPUTS: DATA_SETS M, DATA_SETS X, USER PARAMETERS)
{

//Let M be set of all speaker models and let X be all set of all test sets
//Let E be distance table and let T be result table
(1) FOR EACH M AND X DO
          DIVIDE Mi IN TO BLOCKS; //block length has defined by user
(2) INITIALIZE E AND T; SET Threshold = MinThreshold;

(3) FOR EACH M AND X DO
          Ei = ComputeDistance (X, M);
```

```
(4) FOR EACH Ei AND Threshold DO{
          IF Ei > Threshold //FR-type error observed{
                    WRITE TO (Ti, FR_column, 1.0)}
          IF Ei <= Threshold  //FA-type error observed{
                    WRITE TO (Ti, FA_column, 1.0)}
          WRITE TO (Ti, Threshold_column, Threshold)
          }
(5) FOR EACH Ti DO{
          MODIFY (T, FA- AND FR-column's values) TO percents;

(6) COPY T TO output file;
}

END FOR;
```

Ohjelmassa ovat käytössä taulukossa 7.1 esitetyt käyttäjän vaihdettavissa olevat parametrit:

Taulukko 7.1: Verify_err -testausjärjestelmän käyttäjän vaihdettavissa olevat parametrit.

--test	Ajossa käytettävän testidatan tiedostojen nimet.
--models	Ajossa käytettävien puhujamallien nimet.
--out	Tulostiedoston nimi.
--minT	Minimikynnysarvo, josta järjestelmä aloittaa virheparametrien laskemisen (oletus: 0.01).
--maxT	Maksimikynnysarvo, jonne asti järjestelmä jatkaa virheparametrien laskemista (oletus: 1.0).
--inc	Parametri jolla käyttäjä määrää kuinka suurin askelin järjestelmä kasvattaa vertailtavaa kynnysarvoa (oletus: 0.01).
--per	Vertailussa käytettävien testivektoreiden määrä prosentteina (oletus: 100).
--seg	Parametri jolla määrätään kuinka monta vektoria testidatasta järjestelmä käyttää etäisyysarvojen laskemiseen testidatan ja puhujamallin välillä.

Järjestelmä perustuu VQ:hun [17]. Puhujien välinen etäisyysfunktio on keskisuhdeneliövirhe (MSE) testidatan ja puhujamallin välillä. Kynnysarvona käytin yleistä, kaikille puhujille samaa kynnysarvoa. Kynnysarvon laskin *a posteriori* periaatteella.

Ohjelma laskee etäisyysarvot suoraan eikä normalisoi niitä esimerkiksi välille [0..1]. Tästä johtuen "sopiva" minimi- ja maksimikynnysarvo sekä askeleen valitseminen riippuu tietokannasta, mallien koosta yms. Kynnysarvon asettaminen on näin hankalahkoa ja etäisyysarvot kannattaisikin normalisoida jotenkin laajempien testiajojen yhteydessä.

7.2 Testauksessa käytetty puhujatietokanta

Kokeellisessa osuudessa käytetty puhemateriaali kuuluu TIMIT-tietokantaan. Tietokanta sisältää amerikanenglanniksi puhuttua puhetta. Käytetty puhujajoukko koostui sadan puhujan puhemateriaalista. Tietokannassa olevat wave-tiedostot näytteistettiin uudelleen 8000:lle hertzille ja 16 bitin tarkkuuteen. Testiajot olivat tekstiriippumattomia. Puhujamallit ja testimateriaalin tein noudattamalla seuraavassa määriteltyjä tehtäväketjuja.

Puhujamallit:

1. Opetusvaiheessa käytetyn puhenäytteen keskipituus oli noin viisitoista sekuntia.

2. Signaalien esikorostus tehtiin seuraavalla kaavalla
   $output(n) = input(n) - 0.97 * input(n-1).$

3. Seuraavaksi opetusnäytteille tehtiin Fourier-analyysi (ja mel-kepstrianalyysi), jonka tuloksena saatiin 12-ulotteisia piirrevektoreita. Ikkunan pituudeksi valittiin 10 millisekuntia ja ikkunan siirron pituudeksi 15 millisekuntia. Ikkunointiin käytettiin Hamming-ikkunointia. Piirrevektoreihin lisättiin delta- ja delta-delta kepstrit, jolloin piirrevektorien lopullinen ulotteisuus oli 3 * 12 = 36. Delta ja delta-delta -kertoimet kuvaavat kepstrin dynaamisia muutoksia.

4. Piirrevektorit ryhmiteltiin koodikirjoiksi *RLS* (Random Local Search) algoritmilla [13].

Testimateriaalit:

1. Testausta varten teimme testimateriaalin, jonka pituus oli maksimissaan 15 sekuntia (testiajoissa käytettiin myös lyhyempiä testimateriaaleja).

2. Testivektoreillekin tehtiin sama käsittely kuin mallien muodostamisessa (kohdat 2. ja 3.).

7.3 Tulokset

Tässä kohdassa käyn läpi kokeellisen osuuden tulokset. Alakohdat on jaoteltu ja nimetty testattavan asian mukaisesti. Testiajoissa keskityin testaamaan

- koodikirjan koon,
- puhujien lukumäärän,
- testidatan pituuden ja
- järjestelmän suoritusajan

vaikutusta järjestelmän FA-, FR-, EER- ja HTER-tyyppisten virheiden todennäköisyyksiin.

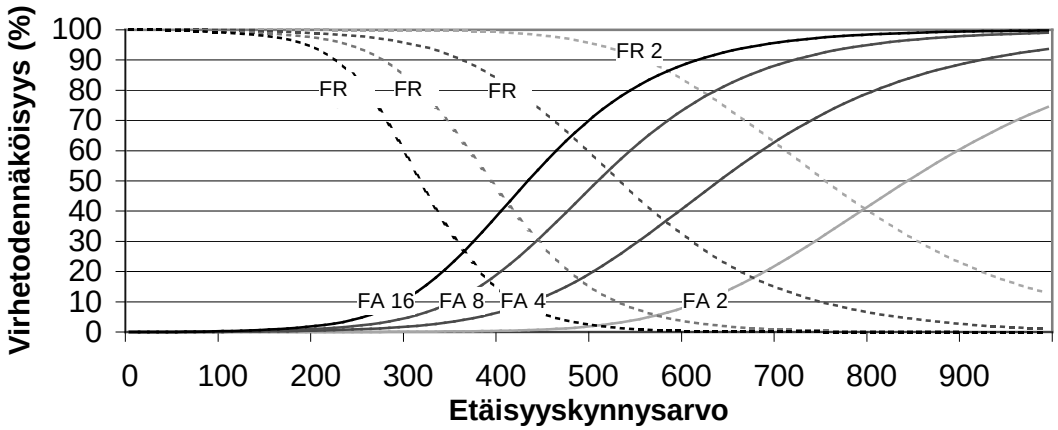
Kiinnitin testiajoissa käytetyn järjestelmän parametrit testauksen aluksi tehtyjen harjoitusajojen perusteella.

7.3.1 Koodikirjan koon vaikutus

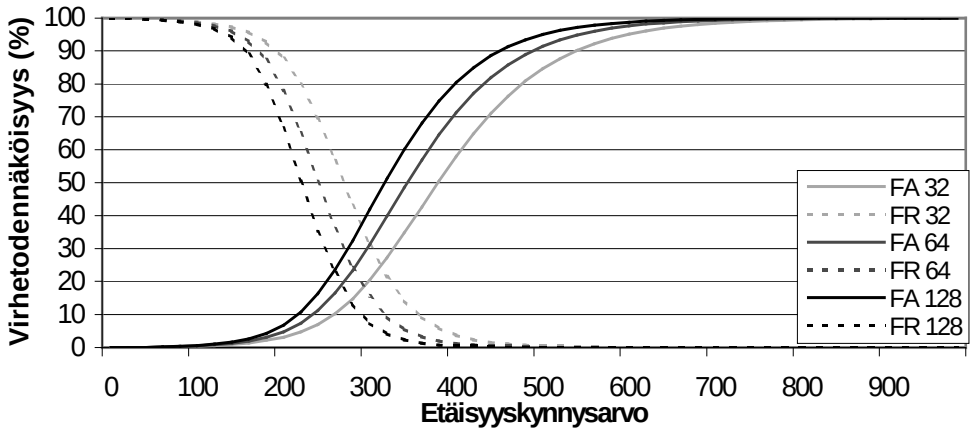
Kokeelliset testiajojen aluksi testattiin koodikirjojen koon vaikutusta järjestelmän FA-, FR-, EER- ja HTER-arvoihin. Testiajoissa 2, 4, 8 ja 16 kokoisilla koodikirjoilla oli kiinnitettynä seuraavat syöteparametrit.

```
verify_err --minT 0.01 --maxT 1000 --inc 10 --per 100 --seg 30
```

Toisin sanoen kynnysarvoa kasvatettiin kymmenen askeleen hyppäyksinä, testidata otetaan mukaan kokonaisuudessaan ja segmentin kokona oli 30 vektoria. Suuremmille koodikirjoille vaihdoin testiajon nopeuttamiseksi `--inc` –parametrille arvon 20, varmennusjärjestelmän tarkkuuteen tämän parametrin muuttaminen ei vaikuttanut. Kuvassa 7.1 on piirrettynä testiajojen FA- ja FR-virheiden tuloskuvaajat 2, 4, 8 ja 16:n kokoisille koodikirjoille. Kuvassa 7.2 on puolestaan 32, 64 ja 128:n kokoisten koodikirjojen kuvaajat.

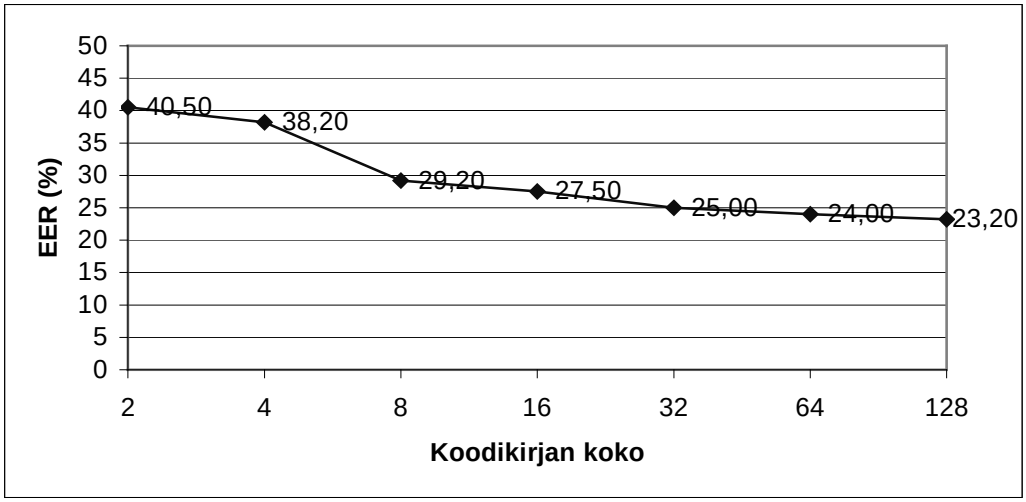


Kuva 7.1: Koodikirjan koon vaikutus FA- ja FR-arvoihin käytettäessä pieniä koodikirjoja.

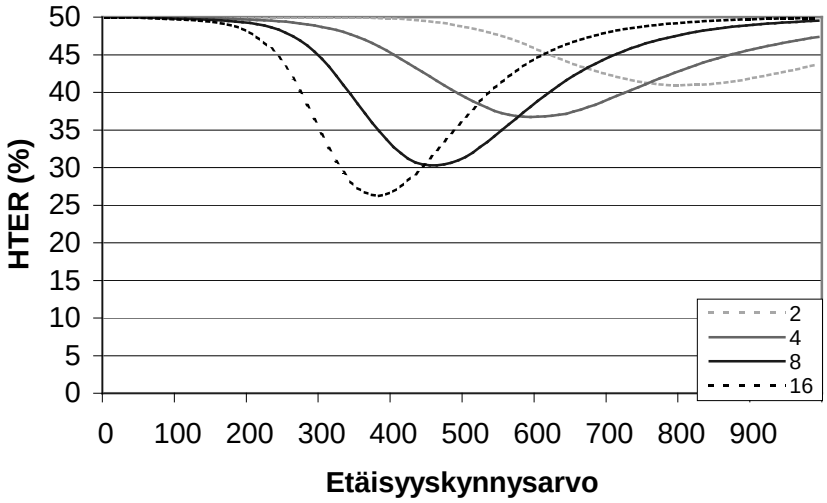


Kuva 7.2: Koodikirjan koon vaikutus FA- ja FR-arvoihin käytettäessä suuria koodikirjoja.

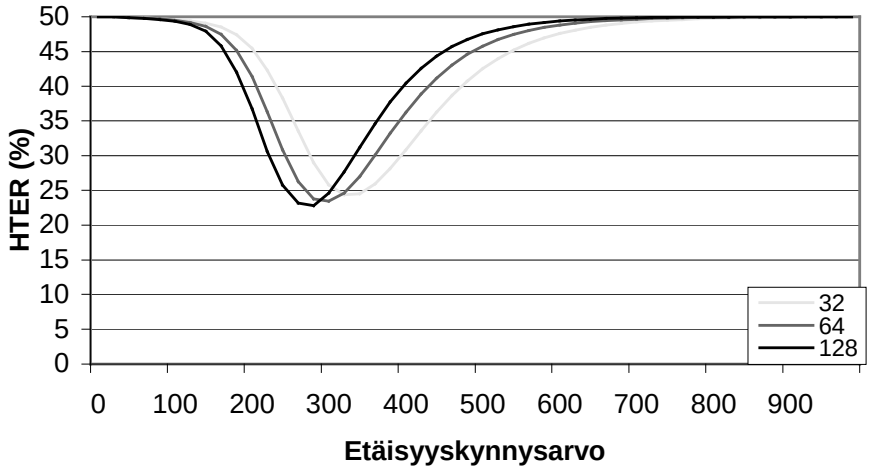
Kuvassa 7.3 on testiajon tuloksena saatujen EER-arvojen kuvaaja. Kuvissa 7.4 ja 7.5 on FA- ja FR-arvojen pohjalta laskettujen HTER-arvojen kuvaajat.



Kuva 7.3: Koodikirjan koon vaikutus järjestelmän EER-arvoihin.



Kuva 7.4: Koodikirjan koon vaikutus järjestelmän HTER-arvoihin käytettäessä pieniä koodikirjoja.



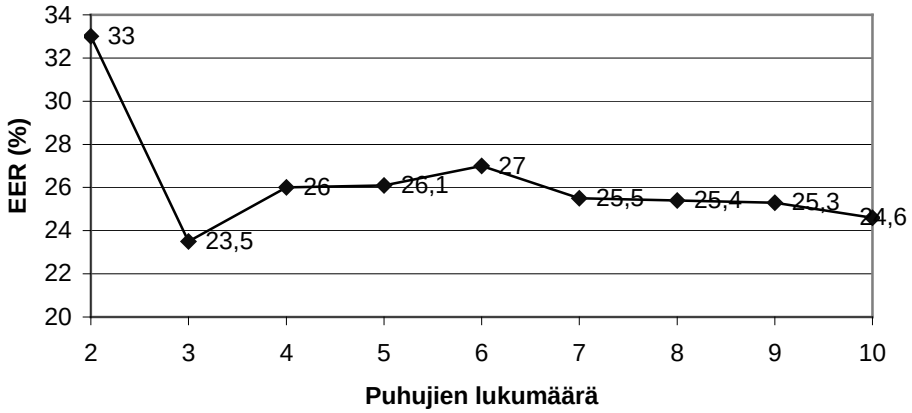
Kuva 7.5: Koodikirjan koon vaikutus järjestelmän HTER-arvoihin käytettäessä suuria koodikirjoja.

7.3.2 Puhujien lukumäärän vaikutus

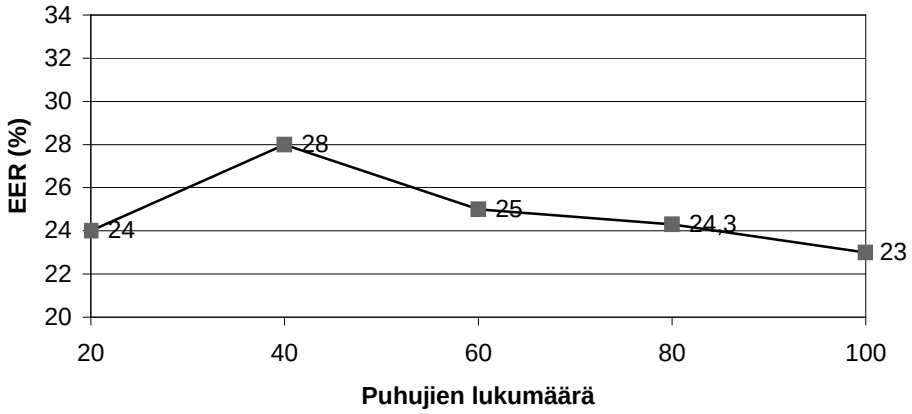
Puhujien lukumäärän vaikutusta järjestelmän FA-, FR-, EER- ja HTER-arvoihin vertailin ajamalla testiajoja erilaisille puhujamäärille. Testiajoissa oli kiinnitettynä seuraavat syöteparametrit.

```
verify_err --minT 0.01 --maxT 1000  --inc 20 --per 100 --seg 30
```

Toisin sanoen kynnysarvoa kasvatettiin kahdenkymmenen askeleen hyppäyksinä, testidata otettiin mukaan kokonaisuudessaan ja segmentin kokona oli 30 vektoria. Lisäksi kiinnitin koodikirjan kooksi 64, jonka havaitsin aiemman testiajon perusteella olevan riittävän suuren järjestelmän tarkkuus huomioiden, mutta samalla myös nopeasti toteutettava. Puhujien lukumäärää varioin niin, että suoritin ajon 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 40, 60, 80 ja 100:lle puhujalle. Kuvissa 7.6 ja 7.7 on testiajon tuloksena saatujen EER-arvojen kuvaajat.

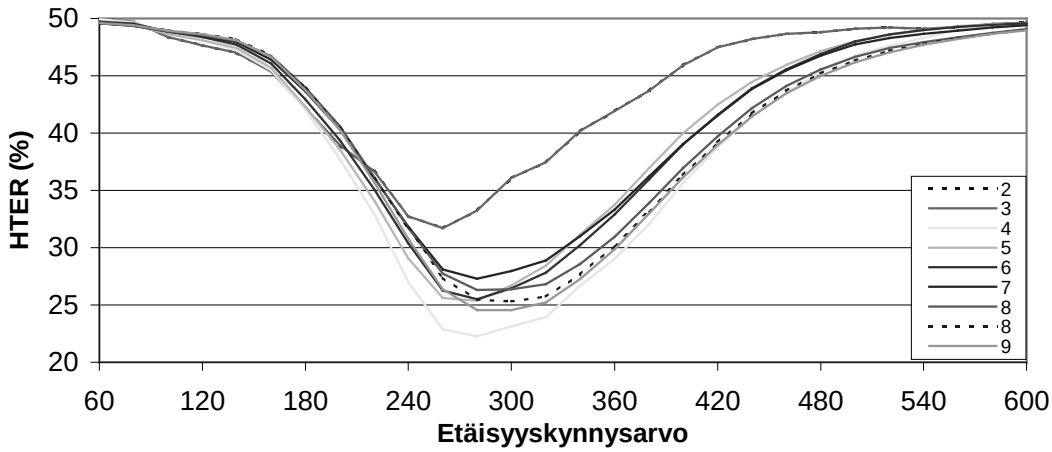


Kuva 7.6: Puhujien lukumäärän vaikutus järjestelmän EER-arvoihin käytettäessä pieniä puhujien lukumääriä.

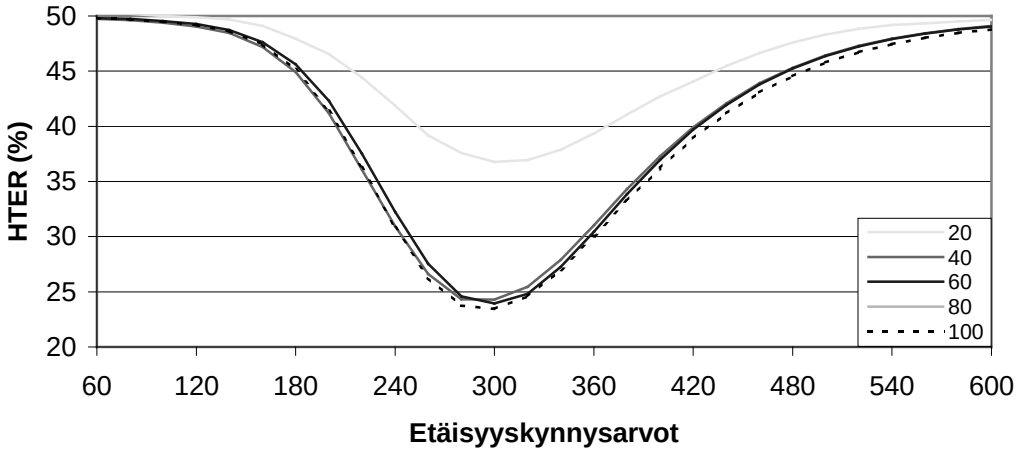


Kuva 7.7: Puhujien lukumäärän vaikutus järjestelmän EER-arvoihin käytettäessä suuria puhujien lukumääriä.

Kuvissa 7.8 ja 7.9 on FA- ja FR-arvojen pohjalta laskettujen HTER-arvojen kuvaajat.



Kuva 7.8: Puhujien lukumäärän vaikutus järjestelmän HTER-arvoihin arvoihin käytettäessä pieniä puhujien lukumääriä.



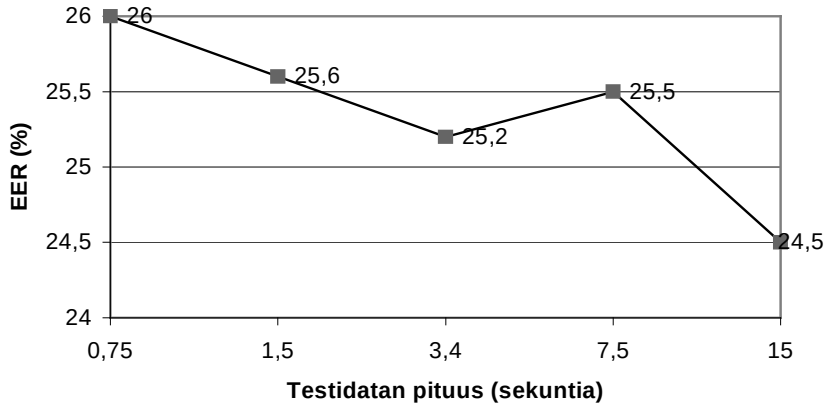
Kuva 7.9: Puhujien lukumäärän vaikutus järjestelmän HTER-arvoihin käytettäessä suuria puhujien lukumääriä..

7.3.3 Testidatan pituuden vaikutus

Testidatan pituuden vaikutusta järjestelmän FA-, FR-, EER-arvoihin vertailin ajamalla testiajoja erilaisilla testidatan pituuksilla (varioin testiohjelman –per –parametria). Testiajoissa oli kiinnitettynä seuraavat syöteparametrit.

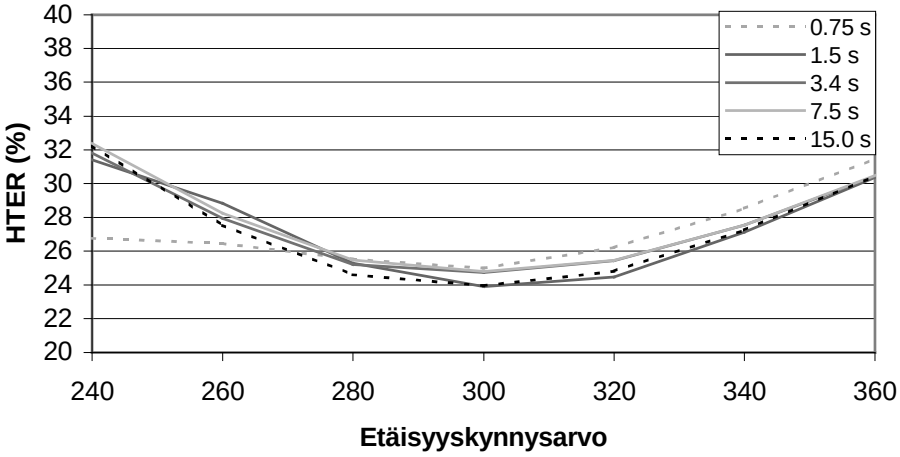
```
verify_err --minT 0.01 --maxT 1000  --inc 20 --seg 30
```

Koodikirjan kooksi kiinnitin 64, ja puhujien lukumääräksi 60. Koodikirjan koon ja puhujien lukumäärää kiinnitin näihin arvoihin, koska aiempien testiajojen perustella pystyin arvioimaan niiden riittävän tarpeeksi tarkan tuloksen saamiseksi. Kuten kuvista 7.3, 7.6 ja 7.7 voidaan havaita, koodikirjan koon muuttaminen 128:aan tai puhujamäärän kasvattaminen 64:ta suuremmaksi ei vaikuta järjestelmän EER-arvoon kovin merkittävästi. Kuvassa 7.10 on testiajon tuloksena saatujen EER-arvojen kuvaaja.



Kuva 7.10: Testidatan pituuden vaikutus järjestelmän EER-arvoihin.

Kuvassa 7.11 on testiajoista saatujen FA- ja FR-arvojen pohjalta laskettujen HTER-arvojen kuvaajat.



Kuva 7.11: Testidatan pituuden vaikutus järjestelmän HTER-arvoihin.

7.4 Tulosten pohdinta

Tässä kohdassa pohdin testiajoissa saamieni tulosten järkevyyttä suhteessa muualla saatuihin tuloksiin ja testijärjestelmän kehityskeinoja tulosten parantamiseksi. Kaiken kaikkiaan testiajojen tulokset olivat rohkaisevia. Vertaillessani saamiani tuloksia muiden saamiin tuloksiin huomasin, että EER-arvot jäivät testeissäni suhteellisen korkeiksi. Tähän on kuitenkin yhtenä merkittävänä selityksenä se että käytin testeissäni sisältöriippumatonta puhedataa. Toisin sanoen opetusdatassa ja testidatassa olevien puhenäytteiden sisältö oli täysin erilainen. Kirjallisuudesta saamani käsityksen mukaan varmennusjärjestelmissä käytetään usein sisältöriippuvaista toteutustapaa, jolla saadaan yleisesti parempi varmennustulos.

7.4.1 Koodikirjan koon vaikutus

Tulosarvojen pohjalta laadituista kuvista (7.1, 7.2, 7.3, 7.4 ja 7.5) voidaan havaita, että koodikirjan koon kasvattaminen pienentää EER:n arvoa ja näin tarkentaa järjestelmän varmennustulosta. Tuloksesta voidaan päätellä, että koodikirjan koon kasvattaminen parantaa varmennustulosta pienentämällä järjestelmän keskimääräistä virhealttiutta. Koodikirjan koon vaikutukset varmennustuloksiin olivat odotettuja.

Koodikirjan koon kasvaessa lisääntyy puhujaa kuvaavien piirrevektorien määrä. Tämän takia puhujan piirrevektorit sijaitsevat keksimäärin lähempänä toisiaan suurempia koodikirjoja käytettäessä. Näin EER:n etäisyyskynnysarvo pienenee koodikirjan kokoa kasvatettaessa. EER-arvot, eli järjestelmän keskimääräiset virhealttiudet eri koodikirjan koilla laskettuna jäivät keskimäärin melko korkeiksi. Niiden pienentämiseksi tarvittaisiin luultavasti verify_err – ohjelman optimointia ja testiin valittavan puhujadatan läpikäynti siinä mahdollisesti olevan puutteen selvittämiseksi. Huomattavaa on myös se että koodikirjan koon kasvaessa HTER-kuvaajan hajonta pienenee (kuvat 7.4 ja 7.5). Tämä johtunee siitä että suuremmalla koodikirjalla varmennustulos tarkkenee, eikä väärällä kynnysarvolla päästä yleisesti niin tarkkaan varmennustulokseen kuin pienillä koodikirjoilla.

7.4.2 Puhujien lukumäärän vaikutus

Tulosarvojen pohjalta laadituista kuvista (7.6, 7.7, 7.8 ja 7.9) voidaan havaita, että puhujien lukumäärän kasvattaminen ei juurikaan vaikuta järjestelmän EER-arvoon, eikä näin heikennä varmennustulosta kovinkaan paljon. Tulos oli hieman yllättävä. Odotin puhujien lukumäärän varioinnin vaikuttavan varmennustulokseen heikentävästi suunnilleen yhtä paljon kuin koodikirjan koon varioinnin. Toisaalta myös testiajoissa käytetty puhujamateriaali saattaa olla rakenteeltaan sellainen, että puhujamäärällä 40 testiin mukaan tulevat puhujat ovat hankalasti toisistaan erotettavissa ja tämä aiheuttaa hypyn kuvassa 7.7 olevaan EER-kuvaajaan.

Kuitenkin testeissä saamieni järjestelmän FA- ja FR-tyyppisten virhekuvaajien arvojen kuvaajista voidaan havaita, että puhujamäärän kasvaessa FA-tyyppisten virheiden varianssi on hieman suurempi kuin FR-tyyppisillä virheillä. Jatkotesteissä olisi hyvä varioida puhujien lukumäärää sadasta eteenpäin ja tutkia testiajojen pohjalta, vaikuttaako puhujien lukumäärä EER-arvoon ratkaisevasti suuremmilla puhuja lukumäärillä.

7.4.3 Testidatan pituuden vaikutus

Testiajojen pohjalta laadituista kuvista (7.10 ja 7.11) voidaan päätellä, että testidatan pituus vaikuttaa järjestelmän EER-arvoon. Pidemmällä testidatalla EER-arvo pienenee, joka oli odotettu tulos. Kun järjestelmällä on käytettävissä varmennukseen enemmän puhetta, järjestelmä päätyy yleisen tietämyksen mukaan varmempaan varmennustulokseen. Ainoa odottamaton tulos oli 7,5 sekunnin testidatalla saatu EER-arvo, joka jäi huonommaksi kuin 3,4 sekunnin testidatalla saatu arvo.

Jatkotutkimuksessa olisi hyvä testata järjestelmää viittätoista sekuntia pidemmillä testidatoilla ja tutkia vaikuttaako pidentäminen enää merkittävästi EER-arvoon.

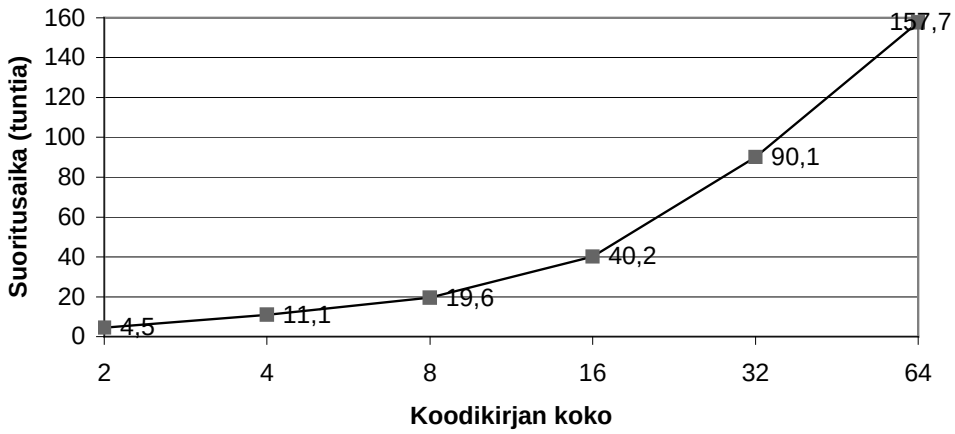
7.4.4 Testiajojen suoritusajat

Suoritin suoritusaikamittauksia Testiajojen suoritusajat olivat lähes odotettuja, mutta suurilla koodikirjoilla ja puhujamäärillä (koodikirjan kokoina 64 ja 128, puhujamäärinä 80 ja 100 pu-

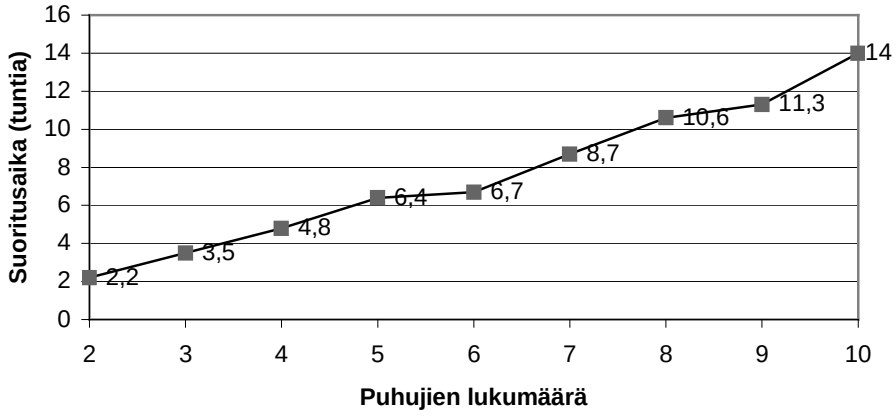
hujaa) suoritusajat pitenivät todella pitkiksi testiajojen suoritukseen varatun ajan ja samanaikaisten ajojen rajoitettujen mahdollisuuksien puitteissa.

Koodikirjan koon ja puhujamäärän tuplaaminen, molemmat, kaksinkertaistivat aika tarkasti järjestelmän suoritusajan. Tulokset olivat odotettuja. Suoritusaikamittaukset ovat kuitenkin lähinnä suuntaa-antavia, eivätkä mahdollista tarkkoja järjestelmän analyysejä johtuen siitä että Joensuun yliopiston Tietojenkäsittelytieteen laitoksen UNIX-keskuskoneella suorittamieni ajojen suoritusaikoihin vaikutti vaihtelevasti muu keskuskoneella ollut kuormitus.

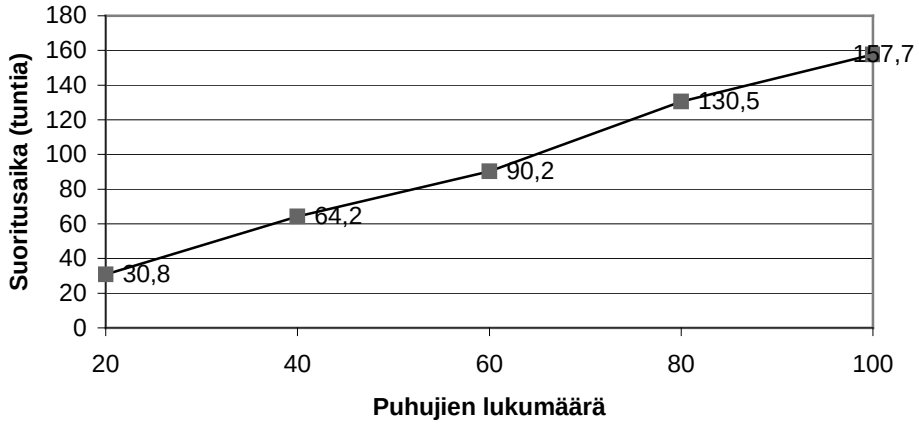
Kuvassa 7.13 on koodikirjan kokoon vaikutusta mittaavien testiajojen ja kuvissa 7.14 ja 7.15 puhujien lukumäärän vaikutusta mittaavien testiajojen suoritusaikakuvaajat.



Kuva 7.13: Testijärjestelmän suoritusajat. Koodikirjan kokona 2, 4, 8, 16, 32, 64 ja 128. Puhujien lukumääränä 100 puhujaa.



Kuva 7.14: Testijärjestelmän suoritusajat. Koodikirjan kokona 64. Pieni puhujien lukumäärä.



Kuva 7.15: Testijärjestelmän suoritusajat. Koodikirjan kokona 64. Suuri puhujien lukumäärä.

8 YHTEENVETO

Automaattinen puhujanvarmennus on jo tällä hetkellä käyttökelpoinen vaihtoehto moneen henkilöllisyydenvarmentamistilanteeseen. Järjestelmien sisältämän päätöslogiikan kehittäminen etenkin kynnysarvojen valinnan kannalta ansaitsee paljon tutkimusta jatkossakin. Automatisoitu kynnysarvojen määrittäminen eri puhujille, järjestelmän keskimääräisen varmennuskyvyn siitä kärsimättä, on yksi varmennusjärjestelmien kehitykseen kohdistuvista suurista haasteista tulevaisuudessa. Puhujakohtaisten kynnysarvojen määrittely on suurempi ongelma sisällöstä riippumattoman, kuin sisällöstä riippuvan järjestelmän tapauksessa. Syynä tähän on opetusvaiheessa kerätyn datan vähäisyys. Sisällöstä riippuvan järjestelmän kynnysarvojen laskentaan EER tuo nopean ja luotettavan menetelmän.

Puhujanvarmennusjärjestelmien päätöslogiikan automatisointi voisi tulevaisuudessa mahdollistaa kynnysarvon automaattisen muokkaamisen käyttäjän jokaisella sisäänkirjautumiskerralla kerätyn puhedatan perusteella. Näin järjestelmän käytettävyys pysyisi koko ajan korkeana, riippumatta käyttäjän puheäänessä tapahtuvista nopeistakaan muutoksista (käyttäjän vilustuminen). Käytettävyydellä tarkoitan tässä järjestelmän antamien FR-tyyppisten virheiden minimointia myös turvallisuuskäyttöön tarkoitetussa varmennusjärjestelmässä. Kynnysarvon reaaliaikaiseen muokkaamiseen on jo kehitetty joitain menetelmiä, mutta mikään menetelmistä ei ole vielä yleistynyt kovin laajalti. Tämä johtunee osittain menetelmien kykenemättömyydestä kynnysarvon reaaliaikaiseen muokkaamiseen tosielämän järjestelmissä, johtuen muokkaamisen vaatimasta suuresta aikavaativuudesta.

Tutkielman alussa määriteltyihin tutkimusongelmiin saatiin vastaukset; käyttämällä puhujakohtaista kynnysarvoa päästään keskimäärin parempaan varmennustarkkuuteen (pohdintaa asiaan liittyen on tutkielman tekstiosuudessa), koodikirjan koon, puhujien lukumäärän ja testidatan pituuden vaikutuksia varmennusjärjestelmän virhearvoihin ja suoritusnopeuteen testasin tutkielman kokeellisessa osuudessa. Testien tuloksista havaittavat vaikutukset olivat pääosin odotettuja. Koodikirjan koon variointi 2 ja 128 välillä osoitti, että koodikirjan koon kasvattaminen parantaa järjestelmän varmennustulosta. Myös testidatan pituuden muutokset vaikuttivat tulokseen odotetulla tavalla; pidempi testidata toi luotettavamman varmennustuloksen. Puhujamäärän variointi 2 ja 100 puhujan välillä ei kuitenkaan vaikuttanut EER-arvoon yhtä merkittävästi kuin odotin. Syynä tähän on luultavasti se että puhujien lukumäärien vari-

ointi alkaa vaikuttaa ratkaisevasti vasta reilusti sataa puhujaa suuremmilla puhujien lukumäärillä.

Jatkotutkimuksessa olisikin kiinnostavaa ajaa uusia testiajoja suuremmilla puhujamäärillä. Testiajot varioituna sadasta viiteen sataan puhujalla olivat tämän tutkielman kannalta liian paljon aikaa vieviä.

VIITELUETTELO

[1] Biometrics, http://www.biometrics.org, Internet-sivu. Viitattu 7.9.2001.

[2] Bourland H., Morgan N.: ”Speaker Verification – A Quick Overview”, tutkimusraportti, Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), 1998.

[3] Campbell P.J.: ”Speaker Regocnition: A Tutorial”, *Proceedings of the IEEE*, vol. 85(9), September 1997, pp. 1437 –1462.

[4] Chen K.: ”Towards Better making a Decision in Speaker Verification”, *Special issue of* Biometrics in Pattern Recognition, vol. 35, 2002.

[5] De Veth J., Gallopyn G., Boulard H.: ”Limited Parameter Hidden Markov Models for Connected Digit Speaker Verification Over Telephone Channels”, *Proceedings of Acoustics, Speech and Signal Processing (ICASSP’1993)*, pp. 247-250, Minneapolis, USA, 1993.

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

[7] Duda R.O., Hart P.E.: *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[8] Encarta 94: Encyclopedia, cd-rom tietosanakirja, Microsoft, 1994.

[9] Equitz W.H.: A new vector quantization clustering algorithm, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, 1989, pp.1568-1575.

[10] Freiberger W., Grenander U.: *A Short Course in Computational Probability and Statistics*, Springer-Verlag, New york, 1971.

[11] Finan R.A., Sapeluk A.T., Damper R.I.: ”Impostor Cohort Selection for Score Normalisation in Speaker Verificarion”, *Pattern Recognition Letters*, vol.18, 1997, pp. 881-888.

[12] French S.: *Decision Theory. An Introduction to The Mathematics of Rationality*, Halsted Press: a division of John Wiley & sons, Great Britain, 1988.

[13] Fränti P., Kivijärvi J.: ”Randomized local search algorithm for the clustering problem”, *Pattern Analysis and Applications*, vol. 3 (4), 2000, pp. 358-369.

[14] Huang X., Acero A., Hon Hsiao-Wuen: *Spoken Lanquage Processing*, Prentice Hall, New Jersey, 2001.

[15] Jovanovic-Dolecek G.: ”Demo Program for the Central Limit Theorem”, *Circuits and Systems. Proceedings of the 40th Midwest Symposium on IEEE*, Sacramento, USA, 1998, pp. 638-641.

[16] Kinnunen, T: *Automaattinen puhujan tunnistus*, Pro gradu –tutkielma, Joensuun yliopisto, Tietojenkäsittelytieteen laitos, 1999.

[17] Kinnunen T., Kilpeläinen T., Fränti P.: "Comparison of clustering algorithms in speaker identification", *Proceeding of IASTED International Conference in Signal Processing and Communications (SPC 2000)*, Marbella, Spain, 2000, pp. 222-227.

[18] Li K.P., Porter J.E.: "Normalizations and Selection of Speech segments for Speaker Recognition Scoring", *Proceedings of Acoustics, Speech and Signal Processing (ICASSP’1988)*, Nex York, USA, 1988, pp. 595-598.

[19] Liu J. H., Chen K.: ”Pruning Abnormal Data for Better Making A Decision In Speaker Verification”, *Proceedings of 6th International Conference on Spoken Language Processing (ICSLP’2000)*, Beijing, China, 2000, pp. 1005-1008.

[20] Lötjönen M.: *Ääneen perustuva käyttäjän todentaminen puhelinverkon lisäarvopalveluissa*, Diplomityö, Lappeenrannan teknillinen korkeakoulu, Tietotekniikan osasto, 2001.

[21] Marques de Sá J.P.: *Pattern Recognition: Concepts, Methods, and Applications*, Springer Verlag, 2001.

[22] Matsui T., Furui S.: ”Likelihood Normalization for Speaker Verification Using a Phoneme- and Speaker-independent model”, *Speech Communication*, vol. 17, 1997, pp. 109-116.

[23] Naik J.M.: ”Speaker Verification: A Tutorial”, *IEEE Communications Magazine*, January 1990, pp. 42-48.

[24] Niemi-Laitinen T.: *Puhujantunnistus rikostutkinnassa*, Yleisen fonetiikan lisensiaatintutkimus, Fonetiikan laitos, Helsingin yliopisto, 1999.

[25] Olsen J.∅.: *Phoneme Based Speaker Recognition*. PhD Thesis. Center for PersonKommunikation, Aalborg University, Denmark, 1997.

[26] Ong S., Yang C.H.: ”A Comparative Study of Text-Independent Speaker Identification using Statistical Features”, *International Journal of Computer And Engineering Management*, vol. 6, 1998.

[27] Oppenheim A. V., Schafer R. W.: *Discrete-time Signal Processing*, Prentice Hall, New Jersey, 1989.

[28] Pierrot J.B., Lindberg J., Koolwaaij J., Hutter H.P., Genoud D., Blomberg M., Bimbot F.: ”A Comparison of a Priori Threshold Setting Procedures for Speaker Verification in the CAVE Project”, *Proceedings of Acoustics, Speech and Signal Processing (ICASSP’1998)*, Seattle, USA, 1998, pp. 125 – 128.

[29] Reynolds D.A., Rose R.C. ”Robust text-independent speaker identification using Gaussian mixture speaker models”, *IEEE Transactions on Speech and Audio Processing*, vol. 3(1), 1995, pp. 72 –83.

[30] Reynolds D.A., Quatieri T.F., Dunn R.B.: ”Speaker Verfication Using Adapted Gaussian Mixture Models”, *Digital Signal Processing*, vol. 10, 2000, pp. 19-41.

[31] Reynolds D.A., Heck L.P.: ”Automatic Speaker Recognition: Recent Progress, Current Applications and Future Trends”, *Presentation slides of Meeting Humans, Computers and Speech Symposium (AAAS’2000)*, Washington, D.C., USA, 2000.

[32] Scharf L.L.: *Statistical Signal Processing Detection, Estimation, and Time Analysis*, Addison-Wesley, Reading, 1991.

[33] Scheaffer R.L.: Introduction to Probability and Its Applicatications, Duxbury Press, 1990.

[34] Schmidt-Nielsen A.: ”Human vs. Machine Speaker Identification With Telephone Speech”, *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP’1998)*, Sydney, Australia, 1998.

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

[36] Toivanen J., Miettinen M.: *Puheentutkimuksen resurssit Suomessa*, CSC –Tieteellinen laskenta Oy, 2001.