Martti Ronkainen

Puhesynteesi pop-musiikissa

Sähkötekniikan korkeakoulu

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin tutkintoa varten Espoossa 29.7.2011.

Työn valvoja ja ohjaaja:

Prof. Paavo Alku

Aalto-yliopisto
Sähkötekniikan korkeakoulu

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Martti Ronkainen

Työn nimi: Puhesynteesi pop-musiikissa

Päivämäärä: 29.7.2011 Kieli: Suomi Sivumäärä: 5+39

Signaalikäsittelyn laitos

Professuuri: Akustiikka ja äänenkäsittelytekniikka Koodi: S-89

Valvoja ja ohjaaja: Prof. Paavo Alku

Puhesynteesiä käytetään nykyään erilaisiin tarkoituksiin. Tässä työssä käsitellään synteesiäänien käyttöä musiikin osana. Työn lähtökohtana on tarve saada synteettinen, mutta tunnistettavasti tietyltä puhujalta kuulostava ääni erääseen elektronisen musiikin kappaleeseen.

Työssä kuvataan nykyaikaisia puhesynteesitekniikoita sekä sitä, miten näille voidaan luoda uusia ääniä olemassaolevan puhujan pohjalta. Käytetyt synteesitekniikat ovat konkatenoiva difonisynteesi, klusteroitu lausekeleikkaussynteesi sekä tilastollisesti parametroitu synteesi. Äänien luomiseen käytettiin Festival-ympäristöä.

Työssä kuvataan uuden synteesiäänien määrittelyä mainituille kolmelle synteesitekniikalle. Synteesiääniä varten äänitettiin kolme tietokantaa, joista luotiin yhteensä viisi erilaista synteesiääntä. Tietokannoista kahdessa käytettiin julkisesti saatavilla olevia korpuksia ja kolmatta varten luotiin oma korpus, joka sisälsi rap-lyriikkaa.

Yleensä puhesynteesin laatu määritellään puheen luonnollisuuden ja ymmärrettävyyden mukaan. Tämän työn kontekstissa synteesiääniä evaluoidaan ensisijaisesti niiden tuottaman äänen luonteen perusteella. Puheen prosodinen soveltuvuus musiikkiin katsottiin toiseksi ja puheen ymmärrettävyys kolmanneksi tärkeimmäksi valintakriteeriksi. Puheen luonnollisuus sen sijaan koettiin vähemmän tärkeäksi syntetisaattorin ominaisuudeksi. Luonnollisuus on tässä pikemminkin haitaksi koska syntetisaattoria ei luotu korvaamaan ihmistä vaan täydentämään sitä, joten syntetisaattorin tulee ainakin jollain tavalla erottua luonnollisesta puheesta.

Äänet luokitellaan näiden kriteerien mukaan ja yksi äänistä valitaan käytettäväksi levytetyssä kappaleessa.

Avainsanat: puhesynteesi, pop musiikki, synteesiäänen luonti, difonisynteesi, klusteroitu lausekeleikkaussynteesi, tilastollisesti parametroitu synteesi

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Martti Ronkainen

Title: Speech synthesis in popular music

Date: 29.7.2011 Language: Finnish Number of pages: 5+39

Department of Signal Processing

Professorship: Acoustics and Audio Signal Processing Code: S-89

Supervisor and instructor: Prof. Paavo Alku

Speech synthesis has been used for various purposes, from assisting the handicapped to information systems. This thesis is about using synthetic voices in popular music. Motivation for the work was to create a new synthetic, but recognizable speech synthesis voice based on an existing speaker. This voice was needed for a piece of electronic music.

This thesis describes modern speech syntesis technologies and engines and how new voices can be built for them. Synthesis technologies used are diphone, clustered unit selection (CLUNITS) and clustered unit generation (CLUSTERGEN) synthesis.

The process of building a new voice is described for all three synthesis technologies based on the Festival speech synthesis system. All in all five different synthesis voices were created using three different corpora. Two of the corpora were existing, well know and the last one was designed specially for this thesis. The designed and selection of the corpus, which consists of rap lyrics is also described.

Speech synthesizers are normally evaluated based on speech naturalness and intelligibility. In the context of this thesis, the synthetic voices are primarily ranked based on the characteristical sound of the synthesizer. Second and third ranking criteria was judged to be speech prosody and its suitability to the the musical environment and speech intelligibility. Speech naturalness was ranked least important. Naturalness is almost considered as a undesirable quality as the synthesizer is not designed to replace a human voice in the musical peace but to augment it so the synthesizer should be somehow distinctive from natural speech.

The created voices are ranked according to the specified criteria and one of the five voices was selected and used in a recorded piece.

Keywords: speech synthesis, popular music, creating synthetic voice, diphone synthesis, clustered unit selection, clustered unit generation

Esipuhe

Tämän työn juuret ovat jossain vuosina 2004 – 2005 Kari Saarilahden kanssa käydyissä keskusteluissa. Aikanaan saimme ajatuksen työssä puhesyntetisaattoriäänen luomiselle ja käyttämiselle. Ajatus jätettiin hautomaan ja siihen palattiin aina aika ajoin kunnes vuonna 2008 tähdet osuivat kohdalleen ja pystyimme tekemään ideasta todellisuutta. Haluaisinkin kiittää Kari Saarilahtea sekä Tommy Lindgreniä, joita ilman tätä työtä ei olisi koskaan tehty.

Lisäksi haluan kiittää valvojaani Professori Paavo Alkua työn valvonnasta ja ohjauksesta.

Espoossa, 25.7.2011

Martti Ronkainen

Sisältö

1 Johdanto

Työssä käsitellään synteesiäänien käyttöä populaarimusiikissa ja kuvataan tapaustutkimuksena kappaleella All Hope [1] olevan englanninkielisen syntetisaattoriäänen syntyä.

Koneellisia ja konemaisia on käytetty tietyissä musiikkigenreissä 1960-luvulta alkaen laajalti, erityisesti elektronisessa musiikissa sekä hip-hopissa. Alkuvaiheessa tätä tehtiin ainoastaan erilaisilla efekteillä, mutta myöhemmin puhesynteesitekniikoiden kehittyessä ja yleistyessä myös puhesyntetisaattoreita on käytetty musiikissa.

Konemaisia ääniä voidaan luoda syntetisoinnin lisäksi efektoimalla muun muassa käyttämällä vokooderia, ns. talkboxia, eräänlaista vireenkorjausalgoritmia tai esimerkiksi tietynlaisella kampasuotimella. Näistä tavoista vokooderi on ollut musiikissa kaikkein eniten käytetyin.

Efektoimalla saatu ääni vaatii aina jonkin lähdesignaalin jota prosessoidaan ja suurin osa efekteillä luoduista synteesimaisista äänistä säilyttää puhujan tunnistettavana. Syntetisaattori taas loivat alunperin geneeristä puhetta ilman tunnistettavuutta. Kuitenkin moderneilla synteesitekniikoilla synteesiääni luodaan jonkin olemassaolevan puhujan pohjalta, joka on yleensä tunnistettavissa syntetisoidusta äänestä. Kuitenkin julkisesti saatavat syntetisaattorit yleensä perustuvat yleensä johonkin tuntemattomaan henkilöön eli ääni jää tältä osin geneeriseksi.

Syntetisoitua ääntä käytettiin ensimmäisen kerran musiikin yhteydessä vuonna 1961, kun Bellin laboratorioiden kehittämä syntetisaattori lauloi laulun "Daisy Bell (Bicycle Built for Two)" [2, 3].

Puhesynteesin historia kulkee käsi kädessä elektronisen musiikin kehityksen kanssa koska synteettinen ääni sopii estetiikaltaan tähän huomattavasti paremmin kuin muihin musiikinlajeihin. Synteettinen ääni tuntuu muuten gimmickiltä kun taas elektronisen musiikin kontekstissa se toimii osana pyrkimystä tehdä puhdas synteettinen ympäristö.

Synteettisen kaltaisia ääniä luotiin alunperin vokooderilla, tällä saadaan muokattua puhesignaalista konemainen. Vokooderi efektinä vaatii aina jonkin syötteen, josta signaali lasketaan ja lähes aina syötteenä käytetään ihmisääntä. Vokooderia on perinteisesti käytetty oikean puhesyntesaattorin sijasta samanlaisen tunnelman luomiseen. Tähän on syynä ensinnäkin alkuperäisten puhesyntetisaattoreiden huono saatavuus ja kalleus sekä myös se, että vokooderin ohjaaminen ja signaalin tekeminen on syntetisaattorin ohjaamista huomattavasti helpompaa, erityisesti kun verrataan vokooderia laulavan syntetisaattorin käyttämiseen. Lisäksi vokooderi jättää alkuperäisen puhujan tai laulajan tunnistettavaksi kun taas puhesyntetisaattori kuulostaisi joko täysin geneeriseltä tai siltä, kenen pohjalta synteesiääni on tehty.

Vokooderia käytettiin laajasti 70- ja 80-luvuilla sekä varhaisessa elektronisessa että hip-hop-musiikissa. Ensimmäisen erityisesti muusikoille tarkoitetun vokooderin takana on muuan Robert Moog ja ensimmäiset levytykset tällä on tehnyt Wendy Carlos. Robert Moog tunnetaan paremmin Moog-syntetisaattorin kehittäjänä ja vokooderi oli yksi osa modulaarista Moog-syntetisaattoria. [4]

Uusimpana vaihtoehtona synteesimaiseen lopputulokseen pääsemiseen on nous-

sut vireenkorjausohjelmisto AutoTunen käyttö. Ohjelmisto on suunniteltu pienien taajuusmuutosten tekemiseen laulusignaalin, mutta kun korjaus säädetään oikein, niin lopputulos on perustaajuuden mukaan vahvasti kvantisoitu ja näin saadaan tehtyä äkillisiä, laajoja muutoksia perustaajuuteen. Tämä tunnetaan yleisesti ns. Cher-efektin kyseisen laulajan kappaleen "Do You Believe" pohjalta jossa tällainen efekti esiintyy ensimmäisen kerran. Kappaleen tuottajat tosin väittivät aluksi, että AutoTunen sijaan tässä olisi käytetty normaalia vokooderia [5]. Valheellinen tieto annettiin siksi, että tuottajat halusivat pitää uuden, keksimänsä tuotantotekniikan salassa.

Halu pitää asioita salassa on melko yleistä ja tämän takia varmaa tietoa levyillä käytetyistä tuotteista ja tavoista on usein vaikea saada. Yksi tämän työn tarkoituksista on omalta osaltaan purkaa tätä salassapidon kulttuuria ja tuoda käytetyt tekniikat ja tulokset tarkemmin ulos keskustelun luomiseksi.

Amerikkalaiset hip-hop/rap-artistit T-Pain sekä Kanye West ovat luoneet Autotunen käytöstä itselleen eräänlaisen tavaramerkin 2000-luvulla. Westin levy "808s & Heartbreak" [6] sisältää lähes ainoastaan tällä tavalla efektoitua laulua ja puhetta. Kyseinen teos julkaistiin saman aikaan kun tämän projektin kokeellista osaa tehtiin. Teoksesta saatiin tavallaan vaikutteita siihen, mitä tästä työstä ei haluttu tehdä koska kyseinen efekti tuntui loppuunkäytetyltä tuon jälkeen.

Varsinaisia puhesyntetisaattoreita on käytetty vähemmän kuin efektejä. Ensimmäinen kaupallisesti ja taiteellisesti onnistunut puhesynteesin käyttäjä on saksalainen Kraftwerk. Kraftwerk käytti aluksi hyvin paljon vokooderia, mutta siirtyi käyttämään oikeaa puhesynteesiä aika pian kun ensimmäisiä järkeviä synteesituotteita tuli markkinoille. Muun muassa levyllä Computer World käytetään suoraan Texas Instrumentsin Speak-n-Spell-tuotetta numeroiden lukemiseen [4]. Nykyään Kraftwerk on siirtynyt käyttämään enemmän Vocaloid-ohjelmistolla luotuja syntetisoituja lauluääniä vokooderin sijaan. [7]

Kraftwerkin lisäksi puhesynteesiä on käyttänt muun muassa englantilainen Radiohead kappaleessa Fitter Happier (MacTalk) [8] sekä U96. Puhesynteesiä on käytetty musikaalisessa kontekstissa myös varsinaisen musiikin ulkopuolella. Esimerkiksi amerikkalainen Man or Astro-Man? -yhtye on käyttänyt puhesyntetisaattorilla luettua tekstiä levyn alussa olevana tervehdyksenä [9] sekä levyn lopussa tekijätietojen lukemiseen sen sijaan että ne olisi kirjattu tavalliseen tapaan levyn kansilehteen [10].

Laulavia puhesyntetisaattoreita on kehitetty useita. Kaupallisesti menestynein on Yamahan Vocaloid-ohjelmisto, jolla tehdyistä kappaleista on tullut Japanissa om taiteenlajinsa. Euroopassa ja Amerikassa tämä ei ole vielä lyönyt itseään erityisesti läpi. [7]

Kaupallisten syntetisaattoreiden tapauksissa puhujan äänen omistusoikeus on mielenkiintoinen tapaus. Vanhoista systeemeistä poiketen nykyisillä synteesitekniikoilla puhe kuulostaa joltain tietyltä henkilöltä, koska tekniikat perustuvat vähintäänkin äänitysten analysointiin jos ei suoranaiseen kopiointiin. Hyvät laulajat tuppaavat olla muutenkin pinnalla joten tässä tapauksessa julkiseen levitykseen tehtävä synteesiääni on hieman huono urasiirto.

Tässä työssä pyritään yhdistämään nämä efektoinnilla saatava äänen tunnistettavuus puhesyntetisaattoriin ja luomaan yleinen puhesynteesiääni siten että se

muistuttaa selkeästi tiettyä henkilöä (laulajaa). Ääntä tullaan käyttämään osana musiikkikappaletta.

Tarkoituksena on luoda synteesiääni, joka kuulostaa tarpeeksi sekä tietyltä henkilöltä mutta kuitenkin koneelliselta. Synteesiäänen ei tarvinnut olla laadukas puhesyntetisaattoreiden perinteisten laatukriteerien mukaan. Syntetisaattorit useimmissa käyttötarkoituksissa arvotetaan puheen luonnollisuuden ja ymmärrettävyyden suhteen, koska yleensä syntetisaattorilla koitetaan korvata oikea ihminen tilanteissa, joissa oikeaa puhujaa ei voida tai haluta esimerkiksi kustannussyistä käyttää. Tällöin syntetisaattorin on järkevää kuulostaa mahdollisimman luonnolliselta ja/tai ymmärretävältä. Tässä työssä puheen luonnollisuus eikä ymmärrettävyys ollut ensisijainen päämäärä sillä oikeaa puhetta pystyttäisiin tarvittaessa äänittämään eikä syntetisaattorilla oltu korvaamassa ihmistä samalla tavalla kuin yleensä.

Ääni ei toisaalta saanut olla täysin epäluonnollinenkaan, koska tällöin puhe ei olisi ollut ymmärrettävää eikä puhuja tunnistettavissa. Puheen tarkoitus on tässäkin viestiä jotain ja siinä auttaa, mikäli puheesta saa selvää. Puhujan tunnistettavuus taas on tärkeää siksi, koska ilman tätä vaatimusta voitaisiin vain käyttää jotain valmista, geneeristä syntetisaattoria. Tämä olisi luonnollisesti huomattavasti helpompaa ja nopeampaa kuin uuden äänen luonti itse. Haluttua lopputulosta eli syntetisoitavaa ääntä kuvattiin sanoilla ”hyvä mutta sopivasti nyrjähtänyt”.

Puhesynteesiäänien luomiseen käytettiin Festival-ympäristöä koska tämä tarjosi hyvän geneerisen ympäristön synteesiäänien luomiseen erilaisilla tekniikoilla. Lisäksi äänien luominen on dokumentoitu hyvin ja sen avuksi on luotu erilaisia työkaluja, joilla erityisesti englanninkielisen syntetisaattorin luominen on suoraviivaista.

Syntetisaattorin kieli oli englanti, koska yhtye levyttää tällä kielellä. Käytetyn puhujan äidinkieli on suomi eli synteesi mallintaa vieraan kielen puhujaa. Puhujalla on kuitenkin melko neutraali, geneerisen amerikkalainen korostus joten vieraskielisyys koettiin pikemminkin mahdollisuudeksi kuin ongelmaksi.

2 Puhesynteesimenetelmät

Puhesynteesiteknologioita on useita ja tätä työtä varten ne voidaan jakaa kahteen kategoriaan. Ensimmäiseen kategoriaan kuuluvat metodit, joisssa synteesi perustuu ennalta äänitetyn puheen osien yhdistämiseen (konkatenoimiseen). Toiseen taas voidaan laskea kaikki muut metodit, jossa puhe luodaan erilaisten laskennallisten ja parametristen mallien mukaan. Kategorioiden väliin jää tilastollisesti parametroivat mallit, joissa puhe kyllä generoidaan laskennallisesti, mutta laskennallinen malli on luotu äänitetyn puheen puheen avulla automaattisesti. [2, 11, 12]

Tunnetuimpia ei-konkatenoivia synteesimalleja ova artikulatorinen sekä formanttisynteesi. Artikulatorinen synteesi perustuu ääniväylän täydelliseen mallintamiseen eli puhe pyritään tuottaamaan samalla tavalla kuin ihminen sen tekee. Näin saadaan teoriassa täydellistä puhetta, mutta käytännön toteutukset eivät ole olleet erityisen onnistuneita koska mallinnuksessa ei ole päästy tarvittavaan tarkkuuteen.

Varhaisimmat puhesyntetisaattorit pyrkivät mallintamaan artikulatorista prosessia. Nämä olivat mekaanisia laitteita, joilla pyrittiin kirjaimellisesti fyysiseen mallinnukseen eli ääniväylän kopiointiin. Tällaisten varhaishistoria ulottuu aina 1700-luvun lopulle asti, jolloin venäläinen Christian Kratzenstein loi ensimmäisen laitteen vokaaliformanttien syntetisointiin. Tämän ja muiden varhaisten syntetisaattoreiden takana oli tarve ymmärtää ja esitellä puheen tuottamisen fysiikkaa pikemminkin kuin halu syntetisoida ääntä. [12]

Ensimmäisenä varsinaisena puhesyntetisaattorina pidetään Homer Dudleyn vokooderin pohjalta kehittämää VODERia. Laite perustuu vokooderin äänen generointiin tarvitsemien parametrien luomisella elektromekaanisen ohjausjärjestelmän avulla. Vokaalit luotiin pianon näppäimistöä muistuttavalla laitteella kun taas prosodian generointiin käytettiin pedaaleita. [4]

Varsinaisesti laadukasta puhetta saatiin ensimmäistä kertaa aikaan formanttisynteesiin perustuvilla tekniikoilla 1960-luvun jälkeen. Näillä pyritään mallintamaan puheen spektria ja saamaan tämä aikaan lähde-suodin-mallin avulla. Näin saadaan melko hyvää ja ymmärrettävää puhetta mutta mallin ohjausdatan generointi on työlästä ja puhe kuulostaa tunnistettavasti synteettiseltä. [12]

Konkatenoivien syntetisaattoreiden suurimpana ongelmana oli pitkään tietokannan vaatima suuri muistin määrä, sekä yhdistämisessä vaadittavan signaaliprosessoinnin vaikeus. Ongelmat ovat vuosien saatossa tulleet tietotekniikan kehityksen ansiosta täysin järkevästi ratkaistaviksi, ja tämän takia konkatenoivat syntetisaattorit ovat yleistyneet 1990-luvun alusta alkaen.

Kehitys on antanut vapauksia kasvattaa konkatenoinnissa käytettyjen yksiköiden pituutta, ensimmmäiset yleiset syntetisaattorit käyttivät yksiköinään difoneja, kun taas nykyään pystytään käyttämään vaihtelevaa yksikönpituutta aina fooneista kokonaisiin sanoihin tai lauseisiin. Konkatenoivista syntetisaattoreista voidaan erottaa kolme alakategoriaa, difonisynteesi, lausekeleikkaussynteesi ja kokonaisia sanoja yhdistävä rajatun sanaston synteesi.

Rajatun sanaston (engl. "limited domain") järjestelmillä tarkoitetaan syntetisaattoria, jolla on tarkasti määrätty käyttöalue. Tyypillisiä tällaisia järjestelmiä ovat erilaiset kuulutukset, puhelinpalvelut ja -valikot, puhuvat kellot ja niin edelleen. Täl-

laisissä yleensä vain yhdistellään kokonaisia sanoja tai lauseen osia yhteen eikä kyse ole varsinaisesti puhesyntetisaattorista. Konkatenoiva syntetisaattori sen sijaan voi myös yhdistellä tarvittaessa kokonaisia sanoja, mutta tässä tapauksessa syntetisaattorin on tarkoitus toimia geneeriselle tekstille.

Kuten alussa mainittiin, puhtaasti parametristen ja konkatenoivien syntetisaattoreiden väliin on noussut uusi syntetisaattorikategoria, tilastollisesti parametroivat syntetisaattorit. Tilastollisesti parametroivat mallit ovat melko uusi ala, tunnetuin synteesiteknologia on Nagoyan yliopistossa kehitetty HTS (HMM-based Synthesis System) joka nimensäkin mukaisesti perustuu piilo-Markov-mallinnukseen [13]. Näissä ääni generoidaan toistaiseksi parametrista synteesiä muistuttavilla tavoilla, mutta parametrien valintaan käytetään lausekelaikkaussynteesistä lainattuja metodeja sen sijaan että ne luotaisiin käsin.

Synteesiäänien kauppallinen historia on huomattavasti lyhyempi kuin tekninen historia. Ensimmäinen kuluttajakäyttöön tarkoitettu puhesyntetisaattori oli Kurzweilin Reading Machine for the Blind noin vuodelta 1976 mutta tämä oli järjestelmänä erittäin kallis. Varsinaisesti yksittäisten kuluttajien saataville puhesynteesi on tullut vasta henkilökohtaisen tietokoneen mahdollistamana. [12]

Seuraavissa luvuissa esitellään tarkemmin kolme tässä työssä käytettyä synteesitekniikkaa, joista kaksi on konkatenoivaa ja yksi perustuu tilastolliseen parametrointiin. Puhtaat parametriset mallit on rajattu tämän työn ulkopuolelle, koska niillä olisi vaikeaa tai mahdotona saavuttaa haluttu päämäärä. Ensisijainen tarkoitus on nimenomaan luoda synteesiääni, joka kuulostaa tietyltä henkilöltä ja puhtaat parametriset mallit yleensä kuulostavat geneerisiltä syntetisaattoreita.

Tässä työssä synteesimoottoreita ajetaan erityisesti tutkimuskäyttöön tarkoitetulla Festival-ympäristöllä [14]. Ympäristö sisältää erikseen varsinaisia puhesynteesimoottoreita, mutta myös hyvän ympäristön tekstin muuttamiseksi synteesimoottorin tarvitsemaksi syötteeksi. Lisäksi Festival-ympäristön synteesimoottoreille on saatavissa kattavat ohjeet uusien synteesiäänien luomiseen [15].

Festival-ympäristö soveltuu projektiimme myös vapaan lisensointinsa puolesta. Ohjelmisto ja dokumentaatio on lisensoitu X11-tyyppisellä lisenssillä, joka antaa vapauden tehdä muutoksia sekä kaupalliseen että epäkaupalliseen käyttöön ilman että muutoksia tarvitsee tehdä vastaavalla lisenssillä julkiseksi. Tämä soveltui tähän projektiin hyvin, koska tuloksien saanti oli epävarmaa ja esimerkiksi äänien vapaa julkaiseminen ei olisi mahdollista koska tulemme käyttämään tekijänoikeuden suojaamaa korpusta.

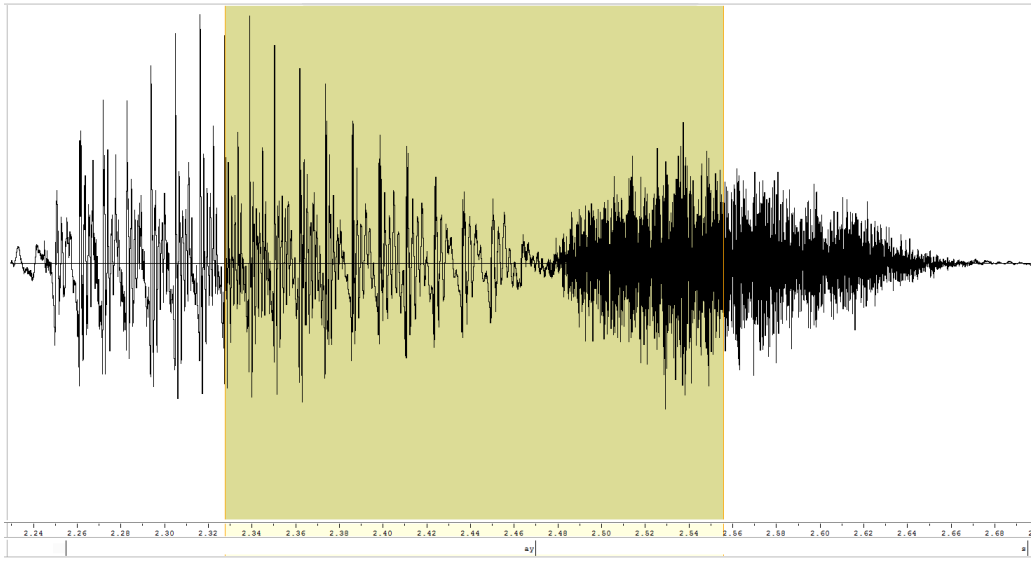
2.1 Difonisynteesi

Difonisynteesi on ensimmäinen konkatenaatioon eli äänitysten yhdistämiseen perustuva synteesitekniikka. Difonisynteesissä liitetään nimensä mukaisesti äänitetettyjä difoneita toisiinsa ja näin syntetisoidaan halutut sanat ja lauseet.

Difonit ovat kahden foneemin yhdistelmiä, joka alkaa ensimmäisen foneemin keskikohdasta ja loppuu seuraavan foneemin keskikohtaan. Difonit soveltuvat foneemeja paremmin konkatenoivan puhesynteesin yksiköiksi, koska foneemien yhdistäminen toisiinsa on ongelmallista koartikulaation takia. Koartikulaatio tarkoittaa pe-

räkkäisten foneemien vaikutusta toisiina, eli puheesta ei käytännössä voida erottaa yksittäistä foneemia /a/, koska /a/ yhdistelmässä /as/ on täysin erilainen kuin /a/ yhdistelmässä /ad/. Esimerkki difonisiirtymästä puheessa on esitetty kuvassa 1.

Koartikulaation takia foneemit eivät esiinny puheessa idealisoidun puhtaina ja siirtymä foneemista toiseen on pikemminkin liukuva kuin äkillinen. Kuvan 1 signaalista siirtymän liukuvuutta demonstroi hyvin signaalin verhokäyrä, joka on sinimäinen. Difonisiirtymän raja on yleensä kyseisen verhokäyrän paikallinen maksimi.



Kuva 1: Foneemit /ay/ ja /s/ sekä difonisiirtymä ay-s. Difonisiirtymä merkitty korostuksella ja foneemien rajat signaalin alapuolella olevilla viivoilla.

Koartikulaation takia konkatenoivassa puhesynteesissä ei ole järkevää syntetisoida yksittäsiä foneemeja. Sen sijaan ongelma voidaan ratkaista käyttämällä synteesiyksikköinä difoneita. Näin yksiköiden liittäminen on foneemeita helpompaa, koska liitokset tehdään silloin kun signaali on monissa tapauksissa stationaarisimmillaan. Difonisynteesiä varten tarvitaan vähintään yksi äänitys jokaisesta kielen difonista joiden määrä riippuu sekä syntetisoitavasta kielestä että murteesta. Difoneita on näissä vaihteleva määrä, tässä työssä difonit on koottu amerikanenglannin Bostonin yliopistoradion korpuksesta jossa niitä on 1369. Difoneita on englannissa periaatteessa $38 \times 38 = 1444$, mutta kaikkia mahdollisia äänneyhdistelmiä ei kielessä esiinny joten ne voidaan jättää pois. Tässä on laskettu jokainen yhdistelmä kerran eli difonien määrä lähes kaksinkertaistuu tästä mikäli ajatellaan difonin /ay-s/ olevan eri kuin difonin /s-ay/.

Synteesia varten difonit voisi periaatteessä äänittää erillisinä, mutta näin ei saataisi luonnollista lopputulosta koska difonien lausuminen ilman täysin ilman kontekstia on hankalaa. Käytännössä difonitietokanta äänitetään siten, että luodaan sanalista jossa jokainen difoni esiintyy sanassa kerran. Sanoina käytetään tähän tarkoitukseen luotuja keksittyjä sanoja eli logatomeja (engl. nonsense words), koska listan täyttäminen oikeilla sanoilla on hankalaa. Keksityillä sanoilla voidaan vakioi-

da difonin paikka sanassa helposti jolloin äänityksistä tulee mahdollisimman konsistentteja. Luonnollisia sanoja käytettäessa difoneita ei saataisi kovin helposti täysin samaan kohtaa ja näin esimerkiksi intonaatioon tulisi eroja.

Logatomit muodostetaan vakioidulla alukkeella ja lopukkeella. Tässä työssä käytetyssä difonisanalistassa alukkeena oli "taa"ja lopukkeena "aa". Sanat muodostetaan siten, että yksittäinen difoni saadaan äänitettyä kumpaankin suuntaan mahdollisimman pienellä vaivalla eli esimerkiksi difonille /d-ae/ tarkoitettu sana on "taa-d-ae-d-aa". Tästä siis käytetään saadaan molemmat difonit /d-ae/ ja /ae-d/. Erikoistapauksena otetaan mukaan difonit, joiden toinen osa on ns. taukodifoni pau.

Difonisynteesissä äänitysten tarkoitus on sisältää difoni geneerisimmillään ja äänitysten tulisi olla mahdollisimman tasalaatuisia. Tämä johtuu siitä, että synteesivaiheessa äänitettyjen difonien paikka voi olla sanassa ja lauseessa missä tahansa ja prosodia tehdään yhdistämisen yhteydessä.

Prosodian luomiseen voidaan käyttää erilaisia tapoja. Festivalin difonimoottori käyttää signaalin luomiseen jäännösherätteistä lineaariprediktiosyntetisaattoria (jatkossa LPC). Tässä käytetään äänityksistä laskettuja LPC-parametreja ja jäännöstä muokkaamalla luodaan prosodia ja aksentit. Jäännösherätteinen LPC on varsinaisesti puhekoodauksen yksi tapa, joten vaikka synteesiin käytetäänkin parametrista mallia, synteesi kuulostaa edelleen alkuperäiseltä puhujalta.

Toinen vaihtoehto olisi tehdä yhdistäminen MBROLA-algoritmilla (engl. Multi-Band Resynthesis Overlap and Add), jossa äänitykset synkronoidaan perustaajuuden suhteen liitosvaiheessa. Lisäksi liitokset tehdään liukuvasti eli käytetään tavallaan crossfadea. MBROLA-synteesienginestä ja äänen luontityökaluista ei ole saatavissa vapaasti käytettävää versiota, joten tätä menetelmää ei tässä työssä käytetty.

Difonisynteesin hyvä puoli on lausekeleikkaussynteeseihin verrattuna tarvittavan tietokannan pienempi koko. Kokoa pienetää tarvittavien äänitysten pienempi määrä sekä erityisesti LPC-synteesiä käytettäessä lineaariprediktion sivutuotteena syntyvä tiedon tehokas pakkaus. Lisäksi difonisynteesillä yleensä saadaan puhtaita parametrisia synteesejä luonnollisempaa ääntä aikaan.

Difonien tarkka äänittäminen vaikeaa, lisäksi nauhoitukset täytyy segmentoida (labeloida, äänitiedostosta merkitään ylös jokaisen foneemin alku-, keski- ja loppukohta) hyvin huolellisesti jotta synteesissa liitokset saadaan mahdollisimman puhtaiksi eikä niihin tule mitään ylimääräistä. Lisäksi vaikka logatomit ovat paras vaihtoehto difonien äänittämiseen, käytännössä niidenkin äänittäminen on vaikeaa koska konemainen puhe ei tule luonnollisesti. Logatomien äänittäminen on ihmisen kannalta hankalaa myös siksi, että sanalistan peräkkäiset sanat ovat yleensä hyvin samankaltaisia.

Prosodian luominen puhtaasti mallintamalla ei aina tuota hyvää lopputulosta, koska lauseista ei aina voida tulkita automaattisesti miten painotusten tulisi mennä. Toisaalta lausekeleikkaussynteesistä poiketen prosodiaa pystytään tarvittaessa muokkaamaan synteesivaiheessa käsin, joten näin voidaan korjata joitain automaattisen mallinnuksen ongelmia erikoistapauksissa. Prosodian generoinnilla ei yleisesti ottaen päästä yhtä hyvään tulokseen kuin hyvällä korpus-pohjaisella järjestelmällä, koska kaikkia koartikulaatiomahdollisuuksia on vaikea saada generoitua ilman huolellista analyysia.

Laulavan syntetisaattorin tekeminen tällä tekniikalla on mahdollista. Esimerkiksi Festival-ympäristöön on tehty laulava syntetisaattori Flinger [16], josta ei ikävä kyllä ollut työn tekohetkellä saatavilla julkaistua versiota.

2.2 Lausekeleikkaussynteesi

Konkatenoiva lausekeleikkaussynteesi (unit selection tai clustered unit selection, suomeksi myös yksikkövalintasynteesi) [17] on difonisynteesistä seuraava looginen askel. Kuten difonisynteesissä, lausekeleikkaussynteesi perustuu äänitetyn puheen analysointiin etukäteen ja synteesivaiheessa yhteenliittämiseen. Suurin ero difonisynteesiin verrattuna on se, että lausekeleikkaussynteesissä pyritään liittämään yhteen pidempiä yksiköitä kuin pelkkiä difoneita. Tätä varten äänitetyn materiaalin sisällön on käytännössä oltava monipuolisempaa, materiaalina käytetään kokonaisia lauseita yksittäisten sanojen sijaan.

Syntetisoitavan yksikön pituus vaihtelee synteesimoottorista riippuen. Erikoistapauksena voidaan liittää kokonaisia sanoja mutta yleensä käytettävät yksiköt ovat lyhyempiä, joko foneemeja, difoneja trifoneja tai tavuja. Lisäksi voidaan käyttää vaihtelevan pituisia yksiköiden pituudet voivat vaihdella. Tässä työssä käytetyssä synteesimoottorissa käytetään yksikkönä analyysivaiheessa foneemeja, mutta käytännössä konkatenointi tehdään lähes aina difoneittain.

Lausekeleikkaussyntetisaattoria varten äänitetään tietokanta, jonka korpus on suunniteltava hyvin. Korpuksen tyylilajin tulisi vastata syntetisoitavaksi tarkoitetun materiaalin tyylilajia. Lisäksi olennaisesti korpuksen tulisi sisältää yleisimmät yksiköt mahdollisimman kattavasti eri kohdista. Sen sijaan harvinaisemmista yksiköistä riittää pienempi kattavuus. [18]

Äänitysten segmentointi on lausekeleikkausynteesissäkin hyvin olennainen osa prosessia, mutta koska liitoksia tulee vähemmän niin rajojen tarkkuus ei ole yhtä kriittistä. Samaten kokonaisten lauseiden automaattinen segmentointi on helpompaa kuin yksittäisten logatomien ja käytetty lausekeleikkaussyntetisaattori tarvitsee segmentoinnista ainoastaan foneemien rajat eikä difoniliitoskohtia kuten edellä.

Korpuksen äänittämisen jälkeen siitä analysoidaan foneettinen sekä prosodinen sisältö ja tämän pohjalta luodaan synteesitietokanta. Kannan luomiseen tarvittavia askeleita käsitellään tarkemmin luvussa 3.3.2.

Synteesivaiheessa yksiköiden valinta on optimointiongelma, jossa pyritään löytämään haluttua tekstiä mahdollisimman hyvin vastaava yksikkösekvenssi puhetietokannasta. Koska täysin vastaavaa sekvenssiä ei etenkään kokonaisia lauseita syntetisoitaessa luultavasti löydy, joudutaan yksiköitä valitessa tekemään valintoja siitä, mistä käytetyt yksiköt otetaan. Tietokanta sisältää aina useita instansseja samoista yksiköistä koska muuten tämä ei juuri eroaisi difonisynteesistä.

Yksiköiden valintaa varten on kehitetty erilaisia tapoja. Käytetyssä klusteroidussa lausekeleikkaussyntetisaattorissa ensin luodaan CART-puun avulla yksikköjoukko eli -klusteri, joka täyttää halutut paramterit mahdollisimman hyvin. Tämän jälkeen klusterista valitaan paras kandidaatti kahden parametrin, kohdekustannuksen (engl. target cost) ja liitoskustannuksen (engl. join cost) perusteella. [17]

Klusteroidussa lausekeleikkaussyntetisaattorissa yksikön kohdekustannuksella tar-

koitetaan etäisyyttä yksikköklusterin keskustasta, toisin sanoen miten kaukana kyseinen yksikkö on teoreettisesta yksiköstä T, joka täyttää halutun yksikön kaikki parametrit täydellisesti. Käytetyssä synteesimoottorissa yksiköiden kohdekustannukset voidaan laskea tietokannan kokoamisvaiheessa. Tämä on laskennallisesti melko raskas operaatio, joten synteesivaihe nopeutuu. Kohdekustannuksen laskemista käsitellään tarkemmin luvussa 3.3.2.

Liitoskustannus lasketaan algoritmilla, joka optimoi samalla myös liitoskohdan. Näin liitokset saadaan hyvin suunnitellussa tietokannassa siirrettyä yksiköiden (foneemien) rajoilta yksiköiden keskelle. Varsinainen liitoskustannus lasketaan yksinkertaisena kahden yksikön etäisyytenä perustaajuuden ja mel-kepstrikertoimien suhteen. Liitoksissa painotetaan perustaajuutta, koska sen epäjatkuvuuskohdat huonontavat synteesin laatua huomattavasti.

Liitoskohdan liikkuvuus ja optimointi aiheuttaa sen, että liitokset ovat lähes aina difoniliitoksia mutta tätä ei varsinaisesti erikseen pakoteta vaan tässä luotetaan siihen, että difoniliitoksen kustannus on aina pienempi kuin ei-difoniliitoksen. Liitoksen paikan laskeminen joka kerta erikseen helpottaa äänitysten segmentoitia, sillä tässä ei tarvitse merkata erikseen difonien rajoja vaan ne optimoidaan aina tapauskohtaisesti. Tämä auttaa korjaamaan difoniäänten jäykkyyttä ja tekee liitoksista luonnollisemman kuuloisia.

Klusteroidussa valinnassa pyritään löytämään yksikköjoukko, jolla osakustannuksista laskettu kokonaiskustannus C minimoituu. Kokonaiskustannus lasketaan kaavalla

$$C = \sum_{i=1}^{N} Tdist(U_i) + W * Jdist(U_i, U_{i-1}), \tag{1}$$

jossa $Tdist(U)$ on yksikön U etäisyys klusterin keskustasta, $Jdist(U_i, U_{i-1})$ yksiköiden U_i ja U_{i-1} välisen optimaalisen liitoksen kustannus. Parametrilla W asetetaan liitoskustannukselle erillinen painoarvo. Parametrin tarkoitus on mahdollistaa hyvien liitosten painottaminen akustisen tarkkuuden sijaan koska hyvät liitos johtaa yleensä parempaan lopputulokseen kuin yksittäisten äänteiden itsenäinen hyvyys. Tämä on loogista kun muistetaan, että liitokset ovat foneemien kaikkein vaihtelevin osa kun taas itse äänne pysyy melko vakiona. Erikoistapauksena äänitetyssä puhunnoksessa peräkkäin olevien yksiköiden liitoskustannus on 0, koska luonnollisen liitoksen koetaan olevan paras mahdollinen. [17]

Kokonaiskustannuksen C minimoivan yksikköjonon $[U_1...U_n]$ etsimisessä käytetään Viterbi-hakua. Algoritmi laskee halvimman reitin peräkkäin asetettujen yksikköklustereiden läpi.

Yksikköjoukon etsiminen hidastuu tietokannan koon kasvaessa, koska samoja yksiköitä on tietokannassa paljon ja yksittäisen yksikön valintaa varten tehtävien vertailujen määrä aina kasvaa. Hakua voidaan helpottaa karsimalla yksiköiden määrää siten, että niitä verrataan keskenään ja keskiarvosta liikaa poikkeavat yksiköt tiputetaan tietokannasta pois. Tämä analyysi tehdään tietokantaa koottaessa, jolloin synteesin ajaminen nopeutuu.

Karsiminen saattaa heikentää laatua jonkin verran koska joitain luonnollisia lii-

toksia voi kadota, mutta käytännössä yksiköistä voidaan poistaa jopa 50 % ennen kuin synteesin laatu huononee merkittävästi, mikäli poistettavat yksiköt on valittu oikein [17, 19]. Poistettaviksi voidaan valita sekä yksiköitä, jotka ovat liian kaukana klusterin keskipisteestä, jolloin niiden voidaan ajatella olevan huonolaatuisia esimerkiksi huonon artikulaation takia. Huonolaatuisten lisäksi voidaan myös poistaa hyvin samankaltaisia yksiköitä, mutta samankaltaisuuden määrittely on vaikeaa joten tätä ei kannata tehdä kuin kaikkein yleisimmille yksiköille. Yksiköiden poistamisesta voi aiheutui joidenkin liitosten huononemista, koska sopivaa luonnollista liitosta ei enää välttämättä poistojen jälkeen löydy.

Synteesi tuottaa parhaimmillaan hyvin luonnollisen kuuloista puhetta, koska synteesissä toistetaan pitkiäkin äänityksiä sellaisenaan ilman mitään prosessointia. Toisaalta mikäli tietokannasta ei löydetä sopivaa yhdistelmää, puheen laatu tippuu radikaalisti, tällaisille tapauksille ei ole olemassa mitään fallback-järjestelmää. Lausekeleikkaussynteesi soveltuukin erityisen hyvin tapauksiin, jossa tarkoitus on generoida puhetta laajasti mutta jollain tasolla ennakoidusti.

Periaatteessa mikään ei estä tekemästä synteesivaiheessa vilunkia ja muokata esim perustaajuutta enemmän tarvittaessa. Tätä ei yleensä tehdä, koska halutaan välttää rankkaa prosessointia koska tämä tekisi laadusta liian vaihtelevaa. Prosessoidut kohdat kuulostaisivat huomattavan erilaisilta "puhtaisiin" osiin verrattuna ja tämän koetaan laskevan synteesin laatua enemmän kuin muutamat huonot liitokset. Tällä myös kannustetaan tietokannan huolelliseen suunnitteluun ja äänittämiseen koska virheitä ei korjata jälkituotannossa.

Huonoin puoli on vaaditun tietokannan suuri koko. Synteesin laatu on yleensä sitä parempi, mitä pidempiä yksiköitä saadaan tietokannasta valittua ovat. Tämä taas vaatii sen, että tietokanta on mahdollisimman kattava eli pitkä. Lisäksi nauhoitusten tulee olla käytettävissä kokonaisuudessaan synteesivaiheessa joten suurempi tietokanta vie levytilaa enemmän. Äänitykset voidaan periaatteessa pakata, mutta tämä ei ole yleensä järkevää koska pienten osasten hakeminen pakatusta datasta on vaikeaa.

Tietokannan koko kasvaa, mikäli halutaan tehdä syntetisaattorista ilmeikäs. Tällöin mikäli ei haluta ruveta prosessoimaan, tarvitsee yksiköt äänettää kaikilla mahdollisilla ilmeillä. Lisäksi ilmeikkyys vaatii tekstianalysaattorilta sen, että syntetisaattoria osataan ohjata käyttämään oikeaa ilmettä, toisin sanoen yksiköiden kohdekustannukset täytyy tulkita tekstistä oikein tai ne on asetettava käsin. Lisäksi luonnollisesti äänitetyn tietokannan analyysin on osattava ottaa vektorit oikein.

Yleiskäyttöistä, hyvälaatuista syntetisaattoria varten vaaditaan puhetta vähintään joitain tunteja. Esimerkiksi noin 1300 puhunnosta sisältävä ARCTIC-korpus on äänitettynä noin puolitoista tuntia. Tällöin jokainen lause on äänitetty kertalleen yhdellä tyylillä eli tietokannassa itsessään ei ole vaihtelua. Puhetyylien lisääminen kasvattaa tarvittavan kannan kokoa jonkin verran, mutta koko korpusta ei tarvitse äänittää kaikilla tyyleilllä. Sen sijaan tarvitaan vain riittävä määrä äänityksiä prosodiamallien luomiseen [20]. Tutkimusta on myös tehty tunteikkuuden lisäämisestä laskennallisilla metodeilla [21] neutraaliin tietokantaan.

Tietokannan laatu ja monipuolisuus vaikuttavat synteesin laatuun suoraan. Ongelmia aiheuttavat tilanteet, joissa joudutaan liittämään lyhyitä yksiköitä toisiinsa

siten, että niiden prosodiset ominaisuudet eivät ei aivan täsmää. Tällaisia tilanteita on vaikea ennustaa etukäteen ja esitetyllä synteesimoottorilla ajon aikana asiaa ei voida korjata. Tämän takia lausekeleikkaussynteesilla on mahdollista saada sekä äärimmäisen hyvä- että huonolaatuista synteesiä pahimmillaan samassa lauseessa. Tämä tekee syntetisaattorin laadun arvioimisesta äärimmäisen vaikeaa koska laatu on täysin eri tasoa eri syötteillä.

2.3 Tilastollisesti parametroiva puhesynteesi

Tilastollisesti parametroiva puhesynteesi on tässä esitetyistä synteesitekniikoista uusin, ensimmäiset tutkimukset julkaistiin 1990-luvun puolivälissä ja ensimmäiset varsinaiset synteesimoottorit 2000-luvun alussa. Tilastollinen parametrointi on syntynyt puheentunnistuksen sivutuotteena, tekniikka perustuu puheesta luotuun piilo-Markov-malliin.

Vastaavia HMM-malleja on käytetty tunnistamaan puhetta ja tässä on tavallaan käännetty tunnistin toisin päin. Sen sijaan että signaalia analysoitaisiin, sitä generoidaan. Tilastollisesti parametroivassa syntetisaattorissa tavallaan tehdään sellainen parametrijono, minkä tunnistin odottaa saavansa tietylle tekstijonolle.

Tilastollisesti parametroiva synteesi pyrkii yhdistämään puhtaasti parametreihin perustuvan syntetisaattorin monipuolisuuden ja vapauden lausekeleikkaussynteesin puheen luonnollisuuteen ja foneettiseen tarkkuuteen. Vastaavasti lausekeleikkaussynteesin tavoin käytetään äänitettyä puhetta, jota käytetään hyväksi parametrisen mallin rakentamiseen ja optimoimiseen, joka on ollut parametristen mallien heikkous.

Puheen malli luodaan äänitetyn korpuksen pohjalta itseoppivalla järjestelmällä, joka tavallaan pyrkii matkimaan äänitetyn korpuksen puhetapaa. Syntetisaattori ei siis tavallaan syntetisoi puhetta, vaan pyrkii luomaan annetusta foneemijonosta signaalin, mikä muistuttaa annetuilta parametreiltaan (signaalin spektri ja perustaajuus) tietokannan sisältöä.

Tässä työssä käytetään Festivalin klusteroitua yksikkögenerointisynteestiä (engl. Clustered Unit Generation, jatkossa Clustergen tai tilastollisesti parametroiva synteesi) [22], jonka toimintaa kuvataan seuraavassa. Muita tilastollisesti parametroivia synteesimoottoreita on olemassa ja ne toimivat pääpiirteittäin samalla periaatteella, mutta yksityiskohdissa on eroja.

Synteesissä käytetään hyväksi piilo-Markov-mallinnusta, jossa tiloina ovat foneemit ja tilasiirtymät parametroidaan ja opetetaan malliin äänitysten perusteella. Mallin luomista käsitellään tarkemmin luvussa 3.3.3.

Synteesivaiheessa tilojen (foneemien) eri parametrit, muun muassa perustaajuus ja foneemin sisäiset muutokset, ja kestot ennustetaan mallin perusteella. Tilat sisältävät parametreinaan perustaajuuden sekä 24 mel-kepstri-kerrointa. Käytetyssä synteesimoottorissa jokaiselle tilalle ennustetaan kolme vektoria ja näistä lasketaan liukuva keskiarvo jotta saadaan lopullinen parametrivektori.

Varsinainen synteesi tehdään lopullisista parametrivektoreista MLSA-suotimella (Mel Log Spectrum Approximation, logaritminen mel-spektriapproksimaatio), joka rekostruoi signaalin kepstrikertoimien perusteella. Suodin ei mallinna ääntöväylän

herätettä, joten ääni on vokooderimainen eli läheskään yhtä kirkas kuin lausekeleikkaussynteesissä. Vokooderimaisuus aiheuttaa sen, että puhe ei kuulosta läheskään yhtä kirkkaalta kuin lausekeleikkaussynteesissä, mutta muilta osin, puhe kuulostaa luonnolliselta, esimerkiksi foneemien liitokset ovat keskimäärin parempia kuin lausekeleikkaussynteesissä. Lisäksi parametrointi säilyttää puheen ylemmän spektrin hyvin joten tietokannan alkuperäinen puhuja on tunnistettavissa.

Tilastollinen parametroinnin ja keskiarvoistamisen ansiosta lausekeleikkaussynteesin akileen kantapää eli ajoittaiset todella huonot liitokset saadaan poistettua lähes kokonaan. Keskiarvoistus kuitenkin hieman huonontaa laatua hieman, koska erityisesti korkeampien formanttien taajuuskaistat leviävät jonkin verran. Tämä on kuitenkin pienempi ongelma kuin huonot liitokset.

Yllämainitussa jokainen vektori ennustetaan itsenäisesti mikä toimii kyllä. Kuitenkin jo difonisynteesissa tuli esille puheen keskinäisriippuvuus joten synteesivaiheessa laatua voidaan parantaa, mikäli vektoreiden ennusteissa käytetään hyväksi edellisiä vektoreita. Tämä voidaan tehdä mallintamalla yksittäisten vektoreiden lisäksi niiden liikerataa (engl. trajectory modelling), jolloin saadaan mallinnettua spektrin muutokset foneemin sisällä tarkemmin.

Tällä hetkellä tilastollisesti parametroivien syntetisaattorien suurin heikkous on vokooderimainen äänenlaatu. Erityisesti lausekeleikkaussyntetisaattoreihin verrattuna vokooderin ”pörinä” on huomattavasti häiritsevää ja ja puheen muu luonnollisuus ja ymmärrettävyys jää helposti huomioimatta. Toisin sanoen puhe on kyllä ymmärrettävää, mutta ei täysin luonnollista. Tilastollisesti parametroiva synteesi on kuitenkin tekniikkana melko uusi ja herätemallien parantamiseksi on tehty tutkimusta.

MLSA-suodin ei ole synteesin kannalta olennainen osa ja sen korvaajaksi on tutkittu erilaisia vaihtoehtoja. Esimerkiksi on ehdotettu tekniikoita STRAIGHT (eräänlainen, parempi vokooderi) [23], HNM (Harmonic Noise Modeling, signaali generoidaan siniäänesten ja moduloidun kohinan summana) [24] tai glottiksen käänteissuodatuksen käyttämisestä herätteenä [25]. Näillä kaikilla saataisiin puhe kuulostamaan ainakin jonkin verran enemmän luonnolliselta.

Tässä työssä käytetylle Clustergen-moottorille ei kuitenkaan ollut saatavilla mitään näistä tavoista joten käytössä oli MLSA-suodin. Työn tarkoituksen kannalta tämä oli sikäli hyvä asia, että näin saatiin selkeästi soundiltaan lausekeleikkauksesta poikkeava tapa mukaan synteesiäänien vertailujoukkoon.

Luonnollisuuden puutteesta huolimatta tilastollisesti parametroivat äänet on rankattu ymmärrettävyyskokeissa melko hyviksi. Esimerkiksi kohinaista puhetta arvioidessa tällä tekniikalla tehdyt äänet on rankattu lausekeleikkaussynteesiä paremmiksi [26].

Tilastollisesti parametroiva syntetisaattori tarvitsee lausekeleikkaussyntetisaattoria pienemmän puhetietokannan hyvän laadun saavuttamiseen. Jopa 200 puhunnoksen tietokannalla päästään hyväksyttäviin tuloksiin ja hyvänlaatuista puhetta saadaan hieman yli 500 puhunnoksen tietokannalla [22].

2.4 Tekstistä puheeksi

Edellisissä kappaleissa kuvatut synteesimoottorit ovat puhesynteesin viimeinen ja näkyvin osa. Niillä siis generoidaan varsinainen haluttu puhesignaali. Tätä osaa prosessista kutsutaan puhesynteesin signaaliprosessoinniksi [27]. Varhaisimmat synteesimenetelmät vaativat ajonaikaista parametrointia, ensimmäinen vapaata tekstiä syntetisoiva kokonainen järjestelmä julkaistiin vuonna 1968 [2].

Synteesimoottorit eivät varsinaisesti osaa generoida puhetta suoraan tekstistä. Sen sijaan ne vaativat syötteenään joko vähintään foneemijonon tai foneemijonon sekä näille prosodisia parametreja, muun muassa perusjakson (intonaatio), foneemin keston sekä foneemin tehon (painotus). Näiden parametrien analysoimiseen tekstistä tarvitaan järjestelmä, joka tekee tekstille lingvistisen esikäsittelyn ja analysoi tekstistä halutut parametrit. Näitä kutsutaan TTS- eli Text-to-Speech-järjestelmiksi (tekstistä puheeksi).

Tekstianalyysin tärkein funktio on tekstin saneistus ja normalisointi. Saneistuksella tarkoitetaan tekstin pilkkomista hallittaviin osiin, lähes aina käytetään virkettä saneistuksen kohdeyksikkönä koska puhesynteesimoottorit haluavat foneemivirran tällaisissa paloissa. Tällöin esimerkiksi intonaatiomalli toimii jokaisen virkkeen sisällä itsenäisesti ja jokaisen virkkeen jälkeen on tauko.

Tekstianalyysissa on useita erilaisia vaikeuksia: samalla tavalla kirjoitetut sanat joilla eri merkitys tai jotka muuten lausutaan eri tavalla, lyhenteitä luetaan eri tavoilla auki, välimerkit eivät ole yksiselitteisiä, esimerkiksi piste voi merkitä virkkeen loppua, lyhennettä tai järjestyslukua.

Tekstin normalisoinnilla taas tarkoitetaan vapaamuotoisen tekstin muuttamista sanalistaksi. Tätä tarvitaan etenkin numeroille ja lyhenteille, mutta myös osa välimerkeistä on tarkoitettu luettavaksi ääneen joten nekin pitää muuttaa sanoiksi. Normalisoinnissa myös saatetaan joutua järjestelemään sanoja uudelleen, puheessa kyllä yleensä teksti luetaan järjestyksessä mutta poikkeuksen tähän luo esimerkiksi merkintä ”US$ 12” eli kaksitoista dollaria.

Normalisoinnin jälkeen alkaa varsinainen sanojen ääntämisen päättely. Ensimmäisenä lauseille tehdään morfologinen analyysi, jossa jokainen sana pyritään palauttamaan perusmuotoonsa jotta niiden merkityksiä voidaan analysoida. Tässä hajotetaan myös yhdyssanat osiinsa. Sanojen perusmuotoja tarvitaan, jotta niiden analyysi on jatkossa helpompaa, erityisesti yhdyssanojen ääntäminen vaatii onnistuneen morfologisen analysoinnin koska muuten sanojen yhdistelmä äännetään todennäköisesti väärin.

Tämän jälkeen tekstin käsittelyä jatketaan syntaksin analysoimisella. Tämä auttaa monitulkintaisten sanojen yksikäsitteistämiseen sekä ääntämisen että painotuksen suhteen. Samalla tavalla kirjoitettu sana voi tarkoittaa eri konteksteissa eri asioita, ja ne tällöin ne luultavasti lausutaan tai painotetaan eri tavalla. Lisäksi syntaktisella analyysillä saadaan virkkeiden prosodiaa tulkittua, muun muassa tauotuksen suhteen.

Lopulta kirjaimet pitää muuttaa äänteiksi. Tämä prosessi on hyvin erilaista eri kielillä, esimerkiksi suomessa kirjainten suhde äänteisiin on hyvin suoraviivainen. Sen sijaan englanninkielisessä syntetisaattorissa tarvitaan satoja erilaisia sääntöjä

ja sen lisäksi ääntämyssanasto, joka kattaa vähintään yleisimmät poikkeukset. [11]

Perinteisesti TTS-järjestelmään on kuulunut myös tekstin halutun prosodian. Tämä on kuitenkin tarpeen vain vanhemille synteesimoottoreille, kuten puhtaille parametrisille syntetisaattoreille mutta myös tässä käytetylle difonisynteesille. Sen sijaan tekstikorpuksiin perustuvat synteesiäänet käyttävät prosodian luomiseen tietokanta hyväkseen eikä TTS-järjestelmällä voida suoranaisesti vaikuttaa prosodisiin piirteisiin.

Esimerkiksi käytetty lausekeleikkaussyntetisaattori ei sisällä minkäänlaista mallia foneemien kestoille, sen sijaan uskotaan siihen, että yksiköiden valinta-algoritmi tekee parhaan mahdollisen valinnan tämänkin suhteen. TTS-järjestelmää olisi mahdollista käyttää prosodian luomisen apuna, vaikka suurin osa prosodiasta luotaisiin korpuksen pohjalta. Tässä tapauksessa TTS-järjestelmä tekisi synteesimoottorille eräänlaisia ehdotuksia halutusta prosodiasta eikä generoisi vastaavasti tarkkoja perustaajuuslukemia.

Nämä ehdotukset ja vihjeet sitten otettaisiin huomioon yksiköiden valintaan tai synteesissä. Kuitenkin tässä työssä käytetyistä äänistä clunits- ja clustergen-syntetisaattoreita varten TTS-järjestelmää käytetään ainoastaan tekstin muuntamiseksi foneemijonoksi eli yllämainittua ominaisuutta ei oltu toteutettu mutta tämä on enemmänkin kyseisten synteesimoottorien kuin varsinaisesti tekniikan puute. [14]

Oman haasteensa TTS-järjestelmälle asettaa tarve syntetisoida ilmaisuvoimaista ja ilmeikästä puhe. Puheen ilmaisutapoja kuvataan prosodisilla piirteillä, mutta halutun tunteen päättely tekstistä on hankalaa tai mahdotonta. Yksi ratkaisu on varustaa teksti metadatalla, jossa annetaan TTS-systeemille vihjeitä halutusta puheen ilmaisutyyleistä. Esimerkiksi XML-pohjainen puhesynteesin ohjaukseen tarkoitettu kuvauskieli SABLE [28] tarjoaa valmiit ratkaisut painotusten, intonaation, prosodian sekä perustaajuuden hallintaan. Näillä pystytään ohjaamaan syntetisaattoria, mutta halutut painotukset tarvitaan jostain. Rajoitetuille syntetisaattoreille voidaan korpus varustaa näillä merkinnöillä melko pienellä vaivalla, mutta yleiselle syntetisaattorille olisi hyvä saada tämä tehtyä jollain tasolla automaattisesti.

SABLEn avulla voidaan myös antaa lisätietoja syntetisaattorille edellämainittujen erikoistapausten käsittelyyn. Esimerkiksi tekstin tekijä voi antaa vihjeitä tekstityylistä tai sisällöstä, jolloin tekstianalyysin taakka helpottu ja tulos on luultavasti parempi kuin automaattisesti tulkittu. Tietty laajemman tekstin osa voidaan merkitä vaikkapa puhelinnumeroksi, osoitteeksi tai joksikin vieraskieliseksi sanaksi. Yksinkertainen SABLE-esimerkki on liitteessä B.

Festivalin TTS-järjestelmä tukee periaatteessa SABLEa ja sen tarjoamia edellämainittujen prosodisten parametrien ohjausmahdollisuuksia, mutta ainoastaan difonimoottorin nämä on implementoitu kattavasti. Eli muut synteesimoottorit eivät huomioi prosodisia parametreja juuri ollenkaan. Tämä on sikäli ymmärrettävää, että näissä moottoreissa on nämä asiat tavallaan mallinnettu valmiiksi, mutta voisi olla hyödyllistä, mikäli moottoreille voisi eksplisiittisesti määritellä, minkälaista prosodiaa tekstissä halutaan. Luonnollisesti prosodian käsin määrittely olisi vielä hyödyllisempää, mikäli synteesimoottori oikeasti voisi tuottaa helposti esimerkiksi mielivaltaisia painotuksia.

Tämän työn kontekstissa etenkin painotusten ja tauotusten parempi ja tarkempi ajonaikainen hallinta olisi ollut hyödyllistä. Rap-musiikin yhteydessä synteesiä käytettäessä puheen rytmityksellä ja painotuksella on normaalia suurempi rooli ja näitä asioita jouduttiin säätämään syntetisaattorin ulosannista lopulta käsin.

3 Synteesiäänien luonti

3.1 Esivalmistelut

Synteesimetodeista päädyimme käyttämään kolmea hyvin erilaista tekniikkaa: difonisynteesiä, klusteroivaa lausekeleikkaussynteesiä (jatkossa pelkkä lausekeleikkaussynteesi tai clunits) sekä tilastollista parametrointisynteesiä (jatkossa clustergen). Tärkeimpänä valintakriteerinä oli se, että näihin kolmeen oli saatavissa kattavin ohjeistus uusien äänien luomiseen. Näiden metodien synteesienginet olivat olleet äänien tekohetkellä valmiina jo jonkin aikaa joten niiden oletettiin olevan laadullisesti kypsiä.

Mainittujen metodien lisäksi harkittiin myös Festivalille saatavissa olevia MultiSyn- ja HTS-synteesimoottoreita (engine). Nämä jätettiin kuitenkin projektin ulkopuolelle, sillä äänien luomisen dokumentaatio ja työkalut koettiin huonommiksi kuin clunits- ja clustergen-moottoreille saatavissa olevat. Etenkin Windows-yhteensopivuuden kanssa oli ongelmia, sopivaa Linux-konetta ei ollut saatavilla. Toisena syynä oli myös se, että kyseiset moottorit olivat hyvin samankaltaisia kuin clunits ja clustergen (vastaavasti) ja tehdyistä äänityksistä pystyttäisiin jälkikäteen luomaan MultiSyn- ja HTS-äänet mikäli tarvetta ilmenisi.

Äänitettävän materiaalin valinta difoniäänelle oli helppoa, tähän käytettiin generoituja nonsense-sanoja jotka ovat muotoa taa shaa shaa. Sanalista generoitiin festivalilla. Sanoja on siis 1369.

Clustergen ja clunits -synteesiäänet luodaan lausekorpusten kohjalta joten korpukset tuli valita. Korpuksen sisällön tulisi vastata sitä, mitä halutaan syntetisoida. Esimerkiksi jos syntetisaattoria käytetään äänikirjojen lukemiseen, parasta materiaalia korpukselle ovat äänikirjat

Lausekorpusten tulisi olla tarkoitukseensa mahdollisimman kattavia. Mikäli synteesiääni on tarkoitettu yleisen tekstin syntetisointiin, tulee korpuksen kattaa täysin minimissään kaikki difonit. Jos halutaan maksimoida synteesin laatu, tulee korpuksen olla huomattavasti tätä isompi.

Päädyimme käyttämään kahta eri korpusta siten, että molemmista tehtiin sekä lausekeleikkaus- että clustergen-äänet. Ensimmäinen käytetty korpus on CMU:n ARCTIC-korpus [29], joka sisältää Jack Londonin tekstejä. Korpus on foneettisesti balansoitu ja suunniteltu yleissyntetisaattorikorpukseksi. Lähdemateriaalin puolesta tämän sanotaan soveltuvan erityisen hyvin äänikirjojen lukemiseen tarkoitettuihin syntetisaattoreihin.

Toinen korpus luotiin itse. Tähän koottiin yhtyeen edellisten levyjen sanoitukset siten, että jokainen säe otettiin omaksi lausekkeeksi. Korpuksessa oli aluksi noin 1600 lauseketta, joista karsittiin pois liian lyhyet ja pitkät säkeet sekä kaksoiskappaleet. Lisäksi liian erikoisia sanoja sisältäviä säkeitä poistettiin korpuksesta.

Siistimisen jälkeen jäljelle jäi 752 lausetta. Korpusta olisi voinut vielä karsia enemmän foneettisten samankaltaisuuksien perusteella, mutta tässä vaiheessa korpus todettiin riittävän pieneksi eikä karsinnalle koettu tarvetta. Festvoxin dokumentaatio suosittelee korpuksen minimikooksi lausekeleikkausäänille 1000 ja clustergenäänille 500 lausetta [15]. Korpuksen karsiminen tehtiin osittain käsin ja osittain sii-

nä käytettiin apuna Festvoxin make_nice_prompts-skriptiä. Korpukselle annettiin nimi DJBB yhtyeestä yleisesti käytetyn lyhenteen mukaisesti.

DJBB-korpuksen sisältö on rap-lyriikkaa. Lähes kaikki lauseet ovat prosodisesti muotoa (alku) (tauko) (loppu). Osassa lauseita tauko tulee kieliopillisesti luonnollisesti sivulauseesta kun taas osassa on käytetty taiteellista vapautta tauotuksen suhteen. Äänitysvaiheessa lauseet puhuttiin kuten ne oli tarkoitettu jotta syntetisaattorin prosodia saataisiin vastaamaan mahdollisimman hyvin alkuperäistä.

DJBB-korpus piti saattaa Festivalin haluaamaan tiedostoformaattiin. Tähän tiedostoon piti koota korpuksen kaikki lauseet lauseet riveittäin näin:

```
( tietokannanNimi_lauseenNumero "lauseenTeksti" )
```

Tiedoston luomista varten ensin kaikki sanoitukset tuotiin ulos SQL-tietokannasta yhteen tekstitiedostoon. Sanoitukset oli tallennettu SQL-tietokantaan muita tarkoituksia varten aikaisemmin. Tekstitiedostosto muokattiin tarvittavaan muotoon Emacsin sekä Microsoft Excelin avulla.

Tämän korpuksen sisältö on tekijänoikeuden suojaamaa. Oikeudet ovat kuitenkin yhtyeen itsensä omistuksessa ja materiaalia käytettään tässä työssä luvalla.

Kaikista korpuksista luotiin syntetisoidut referenssilauseet, eli korpuksien kaikki lauseet syötettiin kertalleen valmiin puhesyntetisaattorin läpi. Referenssilauseita suositellaan käytettäväksi erityisesti difoniääniä nauhoitettaessa, koska sanojen lausuminen ei ole läheskään aina täysin selvää. Lausekorpukset äänitettiin ilman referenssilauseita, koska lausunta oli asiayhteydestä selvää ja referenssien soittaminen olisi vain tarpeettomasti hidastanut prosessia. Referenssilauseita sen sijaan käytetään äänien luomisessa useassa kohtaa apuna, muun muassa segmentointissa.

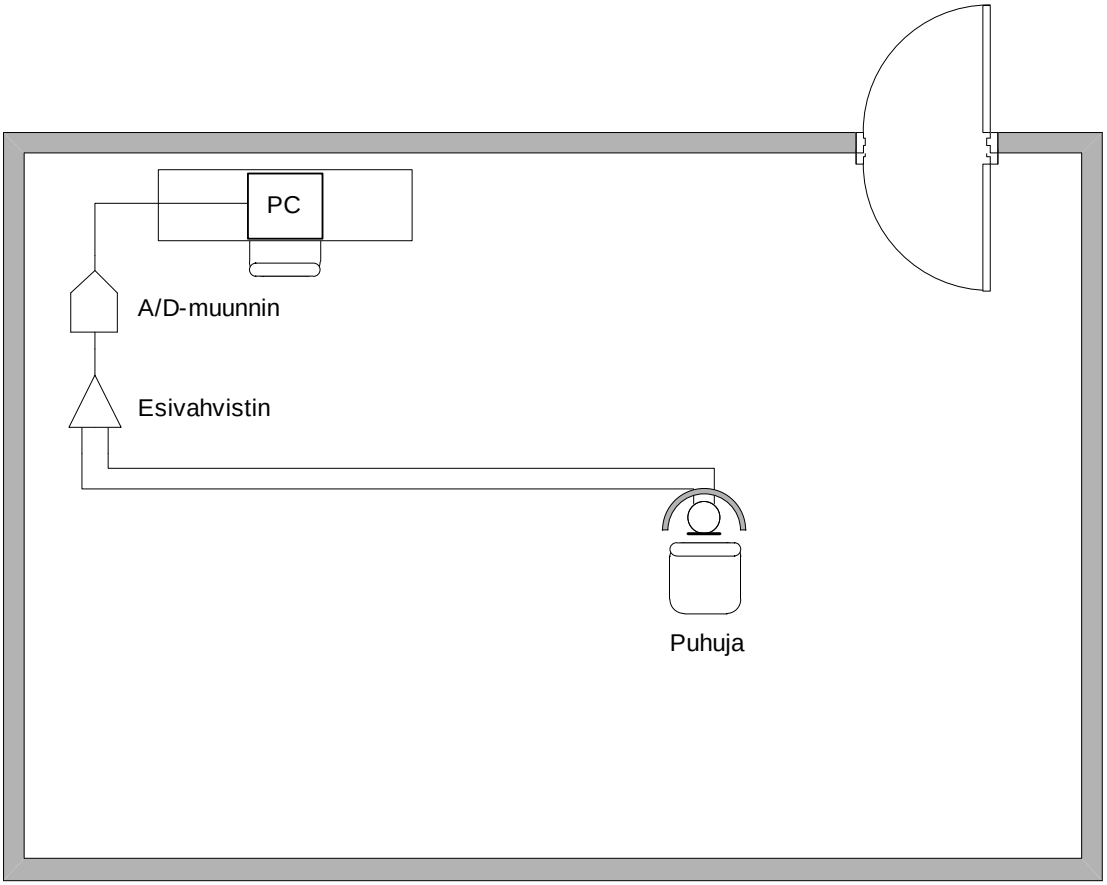
Synteesiääniä luotiin yhteensä viisi: yksi difoni- sekä kaksi molemmille clunits- ja clustergen-moottorille. Sekä Clunits- että Clustergen-moottoreille tehtiin äänet sekä ARCTIC- että DJBB-korpuksen pohjalta. Äänistä käytetään jatkossa nimiä difoni, ARCTIC_clunits, ARCTIC_clustergen, DJBB_clunits sekä DJBB_clustergen. Neljä viimeisessä siis ensimmäisenä käytetty korpus ja jälkimmäisenä käytetty synteesimoottori.

Käytännön prosessina äänet luotiin osittain rinnakkain ja osittain peräkkäin. Ensin tehtiin kaikki äänitykset kerralla järjestyksessä difoni, ARCTIC ja viimeisenä DJBB. Tämän jälkeen äänitysten prosessointi ja synteesiäänien teko aloitettiin difoniäänestä jonka jälkeen edettiin järjestyksessä ARCTIC_clunits, ARCTIC_clustergen, DJBB_clunits ja viimeisenä DJBB_clustergen. Jokainen ääni tehtiin aina kerrallaan ensin siten valmiiksi, että siitä saatiin syntetisoitua puhetta jollain tasolla ennen kuin siirryttiin seuraavaan. Tällä pyrittiin takaamaan se, että seuraavaa ääntä tehdessä voitaisiin ottaa aina opiksi edellisten tekemisestä. Lisäksi molemmat ARCTIC-äänet haluttiin tehdä ensin, koska niiden tekoon oli olemassa hyvät ohjeet, joita voitiin noudattaa ongelmatapauksissa kirjaimellisesti. Sen sijaan DJBB-äänissä oli enemmän omaa työtä joten ARCTIC-äänet toimivat eräänlaisina harjoittelukappaleina.

3.2 Äänitykset

Äänitykset suoritettiin Helsingin Kaapelitehtalla lokakuussa-marraskuussa 2008. Äänitettävää oli melko paljon, noin 1400 nonsense-sanaa difoniääntä varten sekä yhteensä noin 1800 lausetta kahdesta korpuksesta. Äänitykset aloitettiin yksittäisten sanojen äänityksillä 19.10. ja samana päivänä äänitettiin myös puolet ARCTIC-korpuksesta tauon jälkeen. Toinen osa ARCTIC-korpuksesta äänitettiin 21.10. ja viimeisenä 2.11. DJBB-korpus. Äänityskertojen välillä laitteistoa eikä tilaa käytetty muuhun joten olosuhteet pysyivät vakiona tältä osin. Puhujan osalta ääni pysyi hyvin vakiona, jokainen kanta pyrittiin äänittämään mahdollisimman nopeasti jotta vaihtelu saataisiin minimoitua.

Äänitystilana toimi kellarissa sijaitseva huone, jonka koko on noin 6 m x 5 m x 2,5 m. Tila on akustoitu melko kuivaksi. Tila ja äänitysjärjestelyt on esitetty kuvassa 2. Puhuja äänitettiin istualtaan.



Kuva 2: Äänitystila ja -järjestelyt

Signaaliketju ja äänityslaitteisto on esitetty taulukossa 1. Mikrofonina käytössä oli Neumannin TLM103, joka on herttakuvioinen, laajakalvoinen kondensaattorimikrofoni, jonka taajuusvaste on suora välillä 50 Hz – 5 kHz. Viiden kilohertsin yläpuolella on 4 desibelin hyllykorostus.

A/D-muuntimen ja äänikortin välillä käytettiin optista SP/DIF-liitäntää. Signaalitason maksimina pidettiin −12 dBFS:ää ja etuvahvistimessa käytettiin 50 Hz:n ylipäästösuodinta (6 dB / oktaavi).

Käytetty tila ei ollut tarkoitukseen täysin optimaalinen, sillä laitteisto oli kokonaan samassa tilassa mikrofonin kanssa. Kuitenkin hyvälaatuisen analogisen signaaliketjun ja huolellisuuden avulla äänitysten laatu saatiin riittävän hyväksi. Taustamelun minimoimiseksi muun muassa tietokoneen tuuletin pakotettiin pois päältä äänitysten ajaksi. Lisäksi kuten kuvasta 2 käy ilmi, laitteisto pyrittiin sijoittamaan mahdollisimman kauas mikronista siten, että mikronin suuntakuvio osoitti poispäin. Lisäksi mikrofonitelineeseen oli kiinnitetty puolikuun muotoinen absorberi kaiunnan ja taustamelun vähentämiseksi. Äänityksien signaali–kohina-suhde on minimissään noin −32 dB, mikä on tarkoitukseen riittävä.

Taulukko 1: Äänityksen signaaliketju ja käytetty laitteisto

1. Mikrofoni	Neumann TLM103
2. Esivahvistin	Avalon AD 2022
3. A/D-muunnin	RME ADI-8 DS
4. USB-äänikortti	SoundBlaster USB
5. PC	Lenovo ThinkPad T60

Äänityksissä käytettiin apuna modifioitua festivalin prompt_them-skriptiä jonka alkuperäinen tarkoitus on ensin toistaa syntetisoituna haluttu lause/sana sekä esittää sama tekstillä ruudulla. Syntetisoidun esimerkin toistamista ennen sanan äänitystä käytettiin ainoastaan difoniäänien osalta. Alkuperäiseen prompt_themiin verrattuna tehtiin sellainen muutos, että äänittäjä pystyi aina äänityksen jälkeen painamaan joko Enteriä siirtyäkseen seuraavaan tai R:ää jos haluttiin tehdä äänityksestä uusi otto. Samaten käytetyssä skriptissä äänittäjä katkaisee äänityksen käsin, alkuperäinen versio teki äänitykset kiinteillä, syntetisoiduista esimerkeistä arvioiduilla ajoilla. Alkuperäisessä järjestelmässä siis äänitys alkaa ja loppuu automaattisesti. Tämän tarkoitus on helpottaa äänittämisen automaattista etenemistä, mutta koimme paremmaksi mikäli äänittäjällä pystyy kontrolloimaan prosessia käsin.

Äänitys käytännössä etenee siten, että puhuja lukee lauseen kerrallaan ja teknikko painaa nappia jolloin prompt_them lopettaa tiedoston äänityksen ja siirtyy seuraavaan. Skriptin avulla saadaan jokainen lause helposti omaan tiedostoonsa oikealla tunnisteella. Lisäksi skripti soittaa äänitetyn lauseen tarkistusta varten. Järjestelmä on suunniteltu siihen, että puhuja pystyy itse ohjamaan prosessia ilman erillistä äänittäjää, mutta tässä tapauksessa koimme paremmaksi pitää tietokone ja puhuja erillään, koska näin puhuja pystyy keskittymään ainostaan tekstiin eikä mihinkään muuhun. Lisäksi äänittäjän on helppo kontrolloida äänitysten etenemistä ja pyytää yksittäisen lauseen uudelleenäänittämistä tarvittaessa. Äänityksiä jouduttiin uusimaan melko paljon etenkin difonisanojen osalta.

Varsinaiseen äänidatan tallennukseen käytettiin ohjelmaa SoX [30]. Äänitykset tehtiin 44,1 kHz:n näytteenottotaajuudella ja 16 bitin lineaarisella kvantisoinnilla. Synteesiäänien luontia varten äänitteet näytteistettiin alaspäin 16 kHz:iin. Alunperin tarkoitus oli uudelleennäytteistystä silmäläpitäen tehdä äänitykset 48 kHz:n näytteenottotaajuudella mutta A/D-muunninta saatiin synkronoitua tietokoneeseen vain 44,1 kHz:n näytteenottotaajuudella. Uudelleennäytteistys tehtiin soxilla laadulla "high".

Lopputuloksena äänitettyä materiaalia oli lopulta yhteensä noin kolme tuntia. Difonien nonsense-sanojen yhteispituus on noin 45 minuuttia, ARCTIC-korpuksen 1 t 30 min ja DJBB-kannan 45 minuuttia. Äänityssessiot itsessään olivat pisimmillään kolmen tunnin mittaisia.

3.3 Synteesitietokantojen luominen äänitysten pohjalta

Tietokoneen käyttöjärjestelmänä oli Windows XP vaikka lähes kaikki käytetyt ohjelmistot ovat lähtökohtaisesti Unix-pohjaisia. Tässä käytettiin apuna cygwin-ympäristöä jota kaikki käytetyt ohjelmistot tukivat lähes moitteetta. Jonkin verran ongelmia aiheutti ympäristön luominen ja ohjelmistojen asennus, Festivalin käytetty versio (1.96 Beta) oli hyvin valikoiva käännösympäristön suhteen eikä tukenut gcc:n silloista uusinta versiota ollenkaan.

Muuten koko prosessi meni aika lailla Festvox-dokumentaation [15] mukaisesti. Ensimmäiset versiot saatiin toimiviksi nopeasti ilman kovin suurta ylimääräistä virittelyä – oletusparametrit osoittautuivat monessa tapaukessa tässä vaiheessa toimiviksi riittävän hyviksi.

Eri synteesitekniikoiden äänien luonnin dokumentaatiota läpikäydessä tuli selvästi esille, että Festvox-projektissa äänien tekoprosessit tukiohjelmistot olivat kehittyneet vuosien varrella, mutta uusille tekniikoille kehitettyjä apukeinoja ei oltu aina tuotu vanhoihin prosesseihin. Difoniäänen teko-ohjeistus ja tukiohjelmat olivat kaikkein vanhimmat ja Clustergen-äänen taas uusimmat. Tämä näkyi esimerkiksi segmentointin apuohjelmistoissa. Tässä prosessissa pyrittiin mahdollisuuksien mukaan käyttämään kaikkien äänien teossa samoja aputekniikoita.

Yleisesti ottaen äänien luominen on etenkin ensimmäisiä versioita tehdessä erilaisten prosessien puoliautomaattista ajoa, jossa prosessointiin menee vielä enemmän aikaa kuin varsinaiseen virittelyyn. Prosessointiaika oli ääntä kohti muutamia tunteja käytetyllä tietokoneella, difoniääni syntyi nopeimmin noin tunnissa kun taas clustergen-äänien prosessointiin meni noin kuusi tuntia. Prosessia nopeutti se, että samoista korpuksesta tehtiin useampia ääniä jolloin esimerkiksi neljää ääntä varten äänitysten segmentointi tarvitsi tehdä vain kaksi kertaa. Tässä säästyi aikaa, koska lausekorpuksissa käytetty EHMM-segmentoija on oppimisprosessin takia melko hidas.

Segmentointi on synteesin tuloksen kannalta olennaisin askel. Lausekorpusten automaattisesti tehty segmentointi oli aika hyvin kohdallaan, difonien logatomeissa oli enemmän virheitä. Kuitenkin aluksi tyydyttiin kaikkien äänien osalta automaattisen segmentoinnin tuloksiin, koska se oli tässä vaiheessa riittävän hyvä. Ääniä oli viisi ja ensimmäisten versioiden tarkoitus oli saada jokaisesta äänestä sen luonne ja

soundi esille. Tarkoitus oli valita näistä yksi ääni lopulliseen käyttöön ja sen teknistä laatua sitten hiottaisiin enemmän. Lisäksi synteesiäänien referensseinä käytettiin Festvalin oletusääniä eli näitä kuunneltiin alkuvaiheen äänien rinnalla, jotta saataisiin selville kunkin moottorin paras mahdollinen synteesitulos. Oletuksena oli, että valmiit äänet olivat ”täydellisiä” esimerkkejä tekniseltä laadulta ja tässä työssä pystyisimme korkeintaan vastaavaan laatuun.

Seuraavissa aliluvuissa käsitellään äänien luomista tarkemmin synteesitekniikoittain. Prosesseissa on jonkin verran samoja askeleita joten nämä on kuvattu vain kun ne kohdataan ensimmäisen kerran. Lausekeleikkaus- ja clustergen-prosessit ajettiin kahteen kertaan samalla tavalla kahdella korpuksella, ja ajokerrat olivat lähtödataa lukuunottamatta täysin samanlaiset.

3.3.1 Difoni

Ensimmäinen valmiille äänityksille tehtävä asia on niiden segmentointi (labeling). Tällä tarkoitetaan difonien paikkojen ”merkitsemistä” äänitiedostoon siten, että jokaisesta foneemista selvitetään alku-, keski- ja loppukohdat. Peräkkäisten foneemien keskikohtia käytetään yksittäisen difonin rajoina. Segmentointi tehdään siten, että varsinaiseen äänitiedostoon ei tehdä muutoksia, sen sijaan tämä tieto pidetään erillisissä kuvaustiedostoissa.

Segmentointi tehdään lähes aina aluksi koneellisesti, jonka jälkeen merkintöjä korjataan käsin mikäli tarvetta esiintyy. Perinteisesti automaattiset metodit eivät ole olleet täydellisiä, joten olimme valmistautuneet automatian jälkien korjaamiseen käsin.

Automaattiseen segmentointiin on olemassa erilaisia lähestymistapoja ja käytimme eri äänien osalta erilaisia metodeja vertailun vuoksi. Difoniäänellä käytössä oli kepstrianalyysiin perustuva järjestelmä, jossa tehdään ensin mel-kepstrianalyysi sekä itse äänityksistä sekä äänitystä ennen syntetisoiduista samoista sanoista. Varsinainen segmentointi tapahtuu vertailemalla syntetisoidun ja äänityksen kepstrejä keskenään – syntetisoitujen näytteiden difonien paikat on tiedossa ja oletuksena on, että luonnollisen puheen kepstri vastaa ainakin jollain tarkkuudella syntetisoitua. Metodin on kuvannut tarkemmin Malère et al [31]. Kepstrianalyysi tehtiin parametreilla, jotka ovat 12. asteen mel-kepstri, 24. asteen suodinpankki, esikorjaus arvolla 0,97 sekä kehyspituus 5 kertaa paikallinen perusjakso.

Automaatiikka toimi melko hyvin mutta ei täydellisesti. Tämä tuli esille seuraavassa vaiheessa, jossa luodaan segmentoinnin perusteella indeksi difoneista. Indeksi toimii hajautustauluna, jonka avulla linkitetään difonit tiettyyn äänitteeseen. Indeksi luodaan yksinkertaisesti käymällä difonit läpi ja etsimällä difonia vastaava äänite segmentointi-informaation avulla. Ideaalitapauksessa ensimmäisellä ajokerralla indeksiin saadaan kaikki difonit, mutta käytännössä kaikkia difoneja ei ole luokiteltu oikein joten niitä ei löydetä. Tässä tapauksessa selkeästi väärin merkittyjä difoneita noin neljäkymmentä. Väärinmerkinnällä tarkoitetaan sitä, että automaattisegmentointi ei joko löytänyt difonia ollenkaan, tai sitten alku-, keski- ja loppukohdat olivat automatiikan jäljiltä väärässä järjestyksessä.

Normaalissa synteesiäänen tekoprosessissa nämä virheet olisi korjattu samantien.

Tällä erää ensisijainen tarkoitus oli saada syntetisaattorin jonkinlainen prototyyppi aikaan, joten tässä vaiheessa ei vielä korjaustöitä tehty. Vaikka loppuosa prosessista pitäisikin ajaa korjausten jälkeen uudestaan, siihen ei menisi kuin muutamia tunteja laskenta-aikaa kun taas difonien korjaus itessään olisi vaatinut vähintään noin päivän työpanoksen.

Indeksien luomisen jälkeen äänityksistä tulee difonisegmentointia vastaavasti erottaa puheen perusjaksot ja merkata nämä jokaiseen tiedostoon. Tämä tehdään jotta äänityksissä oleva perusjakson lievä luontainen vaihtelu voidaan tasata synteesivaiheessa kun yhdistellään osia eri äänityksistä. Festvox-dokumentin mukaan perustaajuudet saa luotettavimmin selville mittaamalla äänitysten aikana EGG-signaalin. Tätä ei tehty, koska tarvittavaa laitteistoa ei ollut saatavilla.

Perusjakson selvittämiseen signaalista käytettiin ohjelmaa "pitchmark", joka ensin taajuuspäästösuodattaa signaalin annetulle välille. Tämän jälkeen ohjelma laskee autokorrelaation avulla signaalin huippuarvot jotka otetaan perusjaksoiksi. Käytimme suodattimen päästökaistana 80 – 140 Hz:iä ja autokorrelaatiofunktion rajaarvoina 0,005 ja 0,012. Näitä arvoja suositellaan dokumentaatiossa hyvinä lähtökohtina miespuhujille. Soinnittomien äänteiden osalta käytetään interpolointia edellisten kehysten perusteella.

Seuraavana askeleena äänityksistä laskettiin signaalienergiat ja näiden perusteella normalisointikertoimet. Tämän tarkoituksena on normalisoida signaalitaso mahdollisimman hyvin ennen LPC-analyysiä. Normalisointikertoimet lasketaan ainoastaan vokaaleille, vokaalien paikat päätellään aikaisemmin tehdyn segmentoinnin perusteella eli mikäli segmentointi on pielessä, niin myös energia tullaan laskemaan väärin. Äänitetyssä kanssa normalisoinnin tarve oli melko pientä, kerrointen keskiarvo on 1,03 ja keskihajonta 0,17. segmentointin parantaminen luultavasti parantaisi näitä arvoja.

Viimeisenä laskennallisena asiana tehdään LPC-analyysi. Nauhoituksista lasketaan sekä LPC-kertoimet että -residuaali. LPC-analyysi tehdään perusjaksoon synkronoiden. Laskentaan käytettiin 16. asteen LPC-analyysiä Hamming-ikkunalle, esikorjauksen parametrina käytettiin arvoa 0,95.

LPC-parametrien laskemisen jälkeen synteesiääntä pystytiin testamaan. Syntetisaattorista saatiin ääntä ulos, mutta laatu ei ollut kovin kummoinen. Ensisijaisena huomiona oli difoniliitosten huono laatu. Segmentointitiedostoja tarkasteltiin käsin ja pieniä ongelmia löytyi melko paljon, odotetusti erityisesti foneemien keskikohdat eli difonien rajat eivät aina olleet siellä missä niiden tulisi olla. Joitain tiedosto korjattiin harjoitustyön omaisesti, mutta laajempaan korjailuun ei tässä edellämainituista syistä lähdetty. segmentointitiedostojen tarkasteluun ja muokkaamiseen käytetiin wavesurfer-ohjelmistoa [32].

3.3.2 Lausekeleikkaus

Lausekeleikkausäänien luominen alkoi vastaavasti kuin difoniäänissä äänitysten segmentointilla. Lausekorpuksiin käytettiin difonikannasta poiketen kepstrianalyysia kehittyneempää EHMM-segmentoijaa [33], joka perustuu piilo-Markov-mallinnukseen.

Tämän jälkeen luotiin tekstistä kuvaus puhunnoksen rakenteelle (utterance structure). Tällä analysoidaan tekstistä erilaiset kielen segmentit (tavut, sanat, lauseet) sekä teoreettinen prosodia.

Yksiköiden klusterointia varten tarvitaan jokin tapa kuvata jokainen yksikön instanssi u. Tässä tapauksessa kuvausparametrina varten analysoidaan äänitetystä puheesta jokaiselle yksikölle (tässä yksikkönä siis foneemi) akustinen vektori, joka sisältää perustaajuuden, mel-kepstrikertoimet, delta-, teho-, ja perustaajuuskepstrin sekä signaalin tehollisarvot. Perustaajuuden selvittämiseen käytettiin vastaavaa metodia kuin difonisynteesiäänelle. Kepstrianalyysit laskettiin käyttäen kiinteää 10 ms:n ikkunaa, 12. asteen kepstrianalyysiä sekä 16. asteen suodinpankkia.

Vektoreista tehdään yksikkökohtaiset etäisyystaulukot. Keskinäiset etäisyydet lasketaan painotettuna mahalabis-etäisyytenä eli jokaisen parametrin osaetäisyys suhteutetaan parametrin keskihajontaan. Osaparametreja laskiessa yksiköiden kesto interpoloidaan samoiksi (lyhempää pidennetään) jolloin erilaisten kestojen vaikutus muihin ominaisuuksiin kumotaan. Sen sijaan kestojen ero otetaan huomioon yhtenä painokertoimena lopullisessa etäisyydessä. [17]

Lopulta klusterointi tehdään etäisyystaulukoiden ja puhunnosten rakenteiden suhteista. Generointiin käytetään regressiivistä CART-puuta (engl. Classification and Regression Tree, segmentointi- ja regressiopuu). Puu luodaan yksiköittäin siten, että periaatteessa voidaan ottaa huomioon kaikki osaparametrit.

Segmentoinnissa ja klusteroinnissa käytetään hyväksi yksikköä edeltävää ja seuraavaa foneettista sekä prosodista kontekstia (perustaajuus ja kesto), foneettinen paino, paikka tavussa sekä lausekkeessa. Osalle yksiköistä tietyt parametrit eivät ole relevantteja (esimerkiksi soinnittomille äänteille perustaajuus), joten nämä jätetään puun tekovaiheessa huomioimatta aina kun mahdollista.

Puun tarkoituksena on minimoida yksiköiden keskinäinen ”epäpuhtaus” (engl. impurity). Toisin sanoen yksiköt pyritään klusteroimaan siten, että ne ovat yllämainitun akustisen vektorin suhteen mahdollisimman lähellä toisiaan.

Valmis ARCTIC_clunits-ääni oli ensiajoissa parhaimmillaan hyvin vakuuttava ja synteesimoottori pääsi käyttämään kokonaisia sanoja kerrallaan. Toisaalta käytetylle synteesimoottorille tyypillisesti, testimateriaalista generoiduilla näytteillä huomattiin selkeitä huonoja liitoksia. DJBB_clunits-ääni oli yleisesti laadullisesti jonkin verran huonompi, siitäkin saatiin hyvää puhetta heti ulos tietyissä tapauksissa, mutta huonojen ja epäluonnollisten liitosten määrä oli suhteessa selkeästi suurempi kuin ARCTIC-äänessä. Molemmissa oli jonkin verran ongelmia perustaajuuden hyppimisen kanssa eli perusjaksojen selvittäminen äänityksistä ei ollut täysin onnistunut tässä vaiheessa.

3.3.3 Tilastollinen parametrointi

Tilastollisesti parametroivat Clustergen-äänet luotiin viimeisinä. Koska sekä ARCTIC että DJBB-tietokanta oli jo segmentoitu kertaalleen lausekeleikkausääniä varten, niitä ei segmentoitu uudestaan vaan sen sijaan käytettiin hyväksi aikaisemmin EHMM:llä luotua tietoa. Segmentointin lisäksi myös aikaisemmin generoituja puhunnoksen rakennetietoja (utterance structure) pystytiin käyttämään uudelleen.

Seuraavaksi äänitykset parametroitiin selvittämällä perustaajuudet ja mel-kepstri-kertoimet. Tämäkin on periaatteessa täysin sama prosessi kuin clunits-ääntä varten tehtiin, mutta tässä tapauksessa data laskettiin uudestaan. Syy uudelleengenerointiin oli se, että tässä tarvitaan mel-kepstri-kertoimet kiinteällä kehyspituudella, eikä perustaajuuden mukaan synkronisesti. Lisäksi Clustergen-ääntä varten sekä perustaajuus että mel-kepstri-kertoimet tallennetaan hieman erilaiseen muotoon kuin aikaisemmin.

Kepstrikertoimet yhdistetään perustaajuuden kanssa siten, että nauhoituksista luodaan vektorit, joissa on 25 parametria (perustaajuus ja 24 kepstrikerrointa). Vektorit vastaavat Clunits-synteesin yksiköiden vektoreita, mutta Clustergen-ääntä varten vektorit kuvaavat 5 ms osia eikä niitä eikä niitä tässä vaiheessa edes koiteta synkata foneettisen segmentoinnin mukaan.

Seuraavaksi generoitiin tiloille nimet klusterointia varten. Nämä ovat käytännössä englannin foneemit. Tilanimet generoidaan korpuksen perusteella ja tällä käytännössä varmistetaan, että korpuksessa esiintyy kaikki tarvittava.

Lopulta alkaa varsinainen mallin opettaminen ja tilojen klusterointi. Klusterointiin käytetään CART-puuta vastaavasti kuten lausekeleikkaussynteesissä ja pyrkimyksenä on edelleen minimoida yllämainittujen vektoreiden ”epähyvyys”. Suurimpana erona on edelleen se, että foneemien sijasta yksikköpituutena käytetään 5 ms:n kehyksiä, jotka klusteroidaan sen mukaan, mihin foneemiin ne EHMM-segmentointin perusteella kuuluvat. Eli tässä foneemit on jaettu huomattavasti hienojakoisemmin kuin lausekeleikkaussynteesissä.

Lausekeleikkaussynteesistä poiketen tässä joudutaan myös luomaan tilojen kestoajoille mallit, koska yksikköpituus on murto-osa varsinaisesta foneemista. Tämän puu luodaan vastaavasti kuin klusterointipuu.

Perusmuodossaan Clustergen-moottori ei ota synteesissä huomioon foneemien liitoskustannuksia vastaavasti kuin lausekeleikkaussynteesi. Tämä olisi mahdollista tehdä parametrien liikeratojen mallintamisella, jolla saataisiin spektrin muutokset ajan suhteen paremmin syntetisoitua, mutta tehdyissä äänissä tätä ei ensimmäisissä versioissa tehty, tarkoituksena oli tehdä tämä mahdolliseen lopulliseen synteesiääneen.

Kaiken kaikkiaan Clustergen-äänen luomisessa oli selkeästi vähiten parametreja joita olisi tarvinnut muokata. Sen sijaan itseoppiva malli tekee suurimman osan tällaisista päätöksistä itse korpuksen sisällön ja äänitysten perusteella. Tässä korostuu se, että ääni pyrkii vain ja ainoastaan mallintamaan äänityksiä ja synteesimoottori sisältää hyvin vähän puheelle spesifistä tietoa.

4 Äänien vertailu ja analysointi

4.1 Äänien vertailu

Tässä luvussa analysoidaan kvalitatiivisesti äänien eroja ja kuvataan valintaprosessia, jonka jälkeen yksi äänistä valittiin lopulliseksi, käytettäväksi ääneksi. Lopullista valintaa varten koottiin kahdeksan lyhyttä tekstikatkelmaa. Materiaaliksi valikoitui pop- sekä rap-lyriikkaa ja kaksi puhetta. Tarkoitus oli, että nämä edustaisivat mahdollisimman hyvin syntetisaattorille tarkoitettua lyriikkaa, joka ei tässä vaiheessa vielä ollut täysin valmis.

Käytetyt tekstikatkelmat lähteineen on listattu liitteessä A. Teksteiksi pyrittiin hakemaan tarkoituksella mahdollisimman pateettista ja tunteikasta materiaalia, jotta kontrasti tekstin sisällön ja synteettisen ulosannin välillä saataisiin maksimoitua.

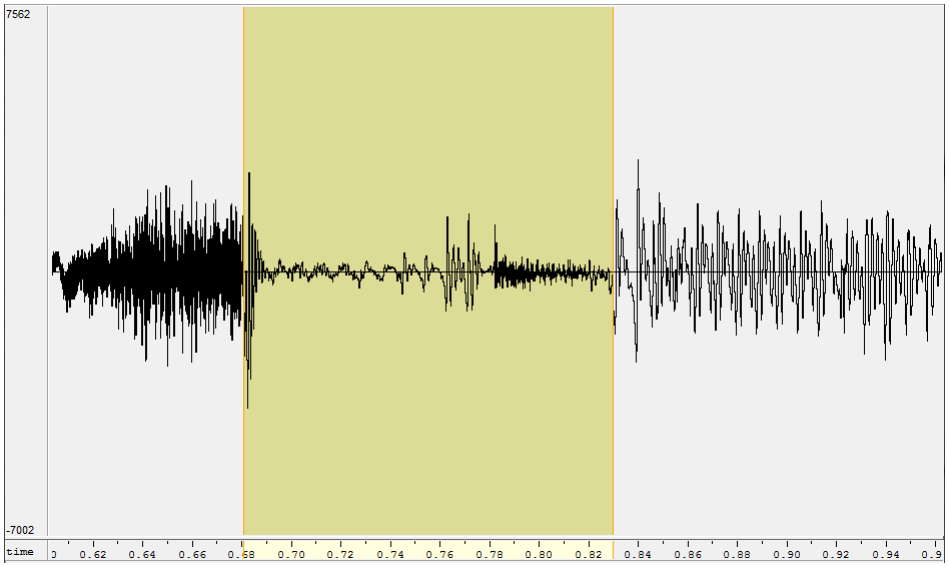
Testimateriaaliin otettiin myös tahallaan lauseita, jotka olivat sellaisenaan DJBB-korpuksessa. Tällä haluttiin selvittää syntetisaattorin ”siirtofunktio” eli miten syntetisaattori toimisi jos sitä käytettäisiin ainoastaan eräänlaisena efektinä. Yhtenä vaihtoehtona oli käyttää syntetisaattoria siten, että sanoitukset olisi otettu korpukseen ja tietokantaan sellaisenaan. Tällä haluttiin selvittää, miten luonnollista puhetta syntetisaattorista saadaan, oletus oli että erityisesti lausekeleikkaussyntetisaattori valitsisi yksiköt siten, että käytettäisiin mahdollisimman pitkiä ketjuja.

Kaikki katkelmat luotiin synteesiä varten SABLE-muotoon [28], joka on XML-pohjainen syntetisoitavaksi tarkoitetun tekstin kuvauskieli. Tällä haluttiin testata myös Festivalin ja synteesimoottoreiden tukea kuvauskielen tarjoamille ohjausmahdollisuuksille. Esimerkki SABLE-tiedostosta on liitteessä B. Demojen generoinnin helpottamiseksi kirjoitettiin yksinkertainen shell-skripti, jolla saatiin ajettua tekstit joko kaikilla äänillä tai vain yhdellä kerrallaan tarvittaessa.

Formaalia kuuntelukoetta materiaalista ei järjestetty koska sitä ei koettu hyödylliseksi. Syntetisaattorin tarkoitus poikkeaa normaalista joten perinteisistä evaluointitavoista ei olisi ollut tässä tapauksessa apua. Alkuvaiheessa näytteet kuitenkin kuuntelutettiin sokkona eli tiedostoihin ei oltu merkitty synteesimetodeja eikä kantoja. Äänien valintaraatiin osallistui yhtye kokonaisuudessaan sekä allekirjoittanut.

Synteesiäänien luonteet saatiin hyvin esille näiden demojen perusteella. Demomateriaalin kokoamisen syntetisoinnin tarkoitus oli saada äänistä esille niiden heikkoudet ja vahvuudet ja tämän takia materiaali oli mahdollisimman vaihtelevaa. Ajatuksena oli myös se, että kun äänet ovat tuttuja, niin tällöin lopullisten sanoitusten kanssa valinta voidaan tehdä mahdollisimman nopeasti. Etukäteen oli tiedossa, että lopulliset syntetisoidut versiot jouduttaisiin todennäköisesti tekemään aikapaineen alaisena, koska lopulliset sanoitukset eivät tulisi valmistumaan erityisen ajoissa.

Difoniäänen sanottiin kuulostavan eniten ”perinteiseltä” puhesyntetisaattorilta hyvässä ja huonossa. Tämän asemaa muihin verrattuna haittasi se, että difonikannassa äänitiedostojen segmentointi oli onnistunut huonoiten koska siinä käytettiin vanhaa, syntetisoitujen sanojen ja äänitysten kepstrivertailuun perustuvaa automaattista segmentoijaa eikä uudempaa EHMM-pohjaista tekniikkaa. Kepstrivertailulle hankalia segmentoitavia olivat erityisesti klusiilit, useassa tapauksessa klusiilin purkauma oli merkattu difonirajan väärälle puolelle ja tämä tulee silloin



Kuva 3: Difoniäänellä syntetisoidun sanan "storm"alkuosa "sto". Foneemi /t/ merkitty korostuksella

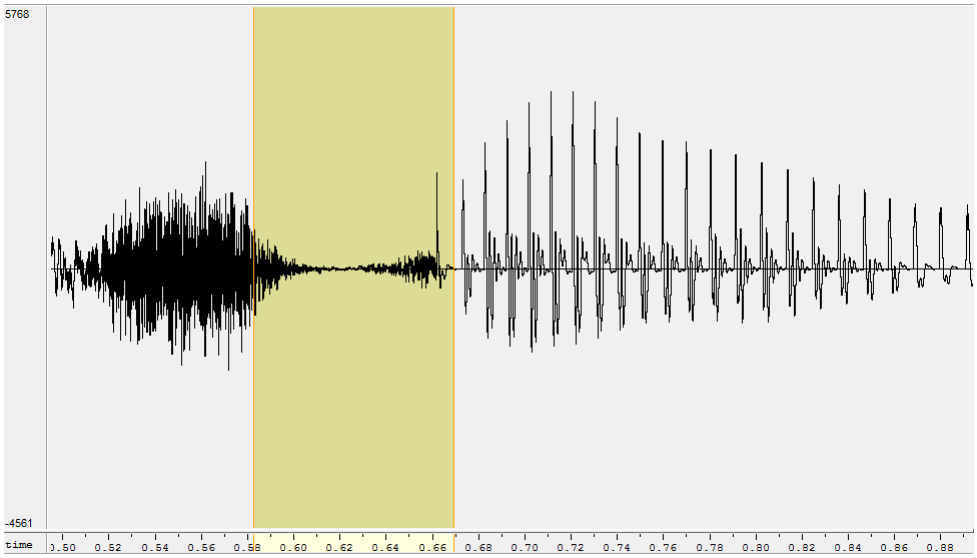
myös syntetisoituun puheeseen väärään paikkaan. EHMM-segmentoijan käyttöä difoniäänelle harkittiin. Se ei soveltunut tähän käyttöön, koska tämä EHMM merkitsi materiaaliin ainoastaan yksittäisten foneemien rajat. Difoniäänen kannalta olennaisempi difonirajoja EHMM-segmentoija ei sen sijaan tiedostoihin merkinnyt. Olisi ollut mahdollista yhdistää EHMM-tekniikalla tehdyt foneemisegmentit aikaisempiin kepstrisegmentteihin, mutta difonirajat olisi silti pitänyt tarkistaa käsin joten tästä ei koettu saatavan hyötyä.

Ilmiö näkyy hyvin kuvassa 3, johon on merkitty korostuksella foneemi /t/ jonka alkuosassa on selvä epäjatkuvuuskohta. Lisäksi klusiilin sisällä ei ole hiljaista aluetta ja klusiilin purkauma ei ole siististi äänteen lopussa vaan se on selkeästi levinnyt keskeltä alkaen.

Vertailun vuoksi sama kohta syntetisoituna ARCTIC_clustergen-äänellä on esitetty kuvassa 4, tämä klusiili näyttää ja kuulostaa, vokooderimaisuutta lukuunottamatta, luonnolliselta. Klusiili myös purkautuu hiljaisuuden jälkeen juuri ennen seuraava foneemia niinkuin pitääkin.

Osa difoniliitoksista oli suoranaisesti huonoja ja tästä aiheutui puheeseen katkoja. Erityisesti klusiilidifonien segmentointitiedostoja tarkastellessa huomattiin, että niissä oli lähes kaikissa pieniä, mutta synteesin kannalta ratkaisevia heittoja. Jotta synteesistä olisi saatu laadullisesti konsistentti, ainakin kaikki klusiilin sisältävät difonit olisi täytynyt käydä läpi ja tarvittaessa korjata. Difoniäänestä olisi saatu tarvittaessa parempi, mutta parantaminen päätettiin jättää ainakin tässä vaiheessa tekemättä.

Lisäksi difonisynteesin testimateriaalia kuunnellessa kävi ilmi, että osa difoneista oli puhujan vieraskielisyydestä johtuen äännetty väärin joten tehdyillä äänityksillä siitä ei välttämättä olisikaan saatu aivan täydellistä. Virheistä huolimatta difoniää-



Kuva 4: ARCTIC_clustergen-äänellä syntetisoidun sanan "storm"alkuosa "sto". Foneemi /t/ merkitty korostuksella.

nen luonne tuli tarpeeksi hyvin esille ja sen perusteella difoniääni jätettiin sivuun. Suurin syy tähän oli se, että difoniääni koettiin tekniikaltaan eräänlaiseksi väliinputoajaksi. Lausekeleikkaussynteesillä saatiin parempaa luonnollista puhetta kun taas Clustergen-äänillä saatiin parempaa robottimaisen synteettista puhetta. Difoniääni sen sijaan kuulosti sekä luonnolliselta mutta robottimaisela, mutta ei tarpeeksi kiinnostavasti kummaltakaan. Lisäksi difoniäänen puhujan tunnistettavuus oli kaikkein huonoin.

Tilastollisesti parametroivien clustergen-äänien ensisijainen piirre oli eräänlainen ”surina” jonka alta puhuja on kuitenkin tunnistettavissa. Suriseva ääni johtuu siitä, että herätemallina käytetään vokooderia jonka päälle laitetaan äänityksistä saatu akustinen malli. Vokooderimaisuus loi mielleyhtymän alan pioneerin Kraftwerkin tuotannon ja erityisesti kappaleeseen Ohm Sweet Ohm [34].

Clunits-äänet olivat odotetusti kaikkein ”kirkkaimpia” ja kuulostivat soundiltaan eniten luonnolliselta puheelta.

Eri korpusten pohjalta tehdyissä clustergen- ja clunits-äänissä oli selkeitä eroja. ARCTIC-korpuksesta luodut syntetisaattorit olivat selkeästi rauhallisempia ja hitaampia kuin DJBB-korpukseen pohjautuvat syntetisaattorit. Lisäksi ARCTIC-äänissä intonaatio oli lauseessa laskeva koska näin oli suurimmassa osassa korpuksen lauseitakin – lauseet olivat suurimmaksi osaksi suoria päälauseita ilman sivulauseita. DJBB-korpus taas oli tyyliltään rap-lyriikkaa, jossa tyylilajiin kuuluu tauko ja intonaation nosto säkeen keskellä.

Foneettisen ulosannin osalta ARCTIC-korpukseen pohjautuvat äänet olivat huomattavasti DJBB-korpuksesta luotuja ääniä parempia. Ero tuli huomattavan selkeästi etenkin clunits-äänissä esille. DJBB-korpuksella oli ongelmia joidenkin sanojen kohdalla ja tämä johtuu korpuksen huonommasta kattavuudesta. Clustergen-

äänien välillä yhtä selkeää laatueroa ei ollut, sillä tilastollinen keskiarvoistamisen ansiosta pienempi korpus on riittävä – yksittäinen puuttuva segmentti on vain yksi kolmesta synteesiin vaikuttavasta tekijästä.

Eräs hieman yllättävä ero ARCTIC- ja DJBB-äänien välillä oli myös äänen perustaajuudessa. ARCTIC-äänissä puheen perustaajuus on keskimäärin noin 85 Hz kun taas DJBB-korpuksen pohjalta tehdyissä äänissä perustaajuus on hieman korkeampi, noin 95 – 105 Hz. Ero on olemassa jo nauhoituksissa ja johtuu ainakin osittain siitä, että ARCTIC-kantaa äänitettäessä puhujalla oli juuri ollut flunssa jonka jäljiltä ääni oli hieman käheä. Lisäksi ARCTIC-korpuksen romaanimainen sisältö kannustaa käyttämään toteavampaa, hitaampaa ja matalampaa rekisteriä kuin DJBB-korpuksen nopeampi iskevämpi lyriikka.

Lisäksi Clustergen-moottorin kestomallinnus vähentää korpuksen laajuuden merkitystä, DJBB_clunits-äänessä oli ARCTIC_clunits-ääneen verrattuna useammissa testilauseissa vääriä (liian lyhyitä) vokaalikestoja kun taas clustergen-äänissä erot korpusten kesken olivat tältä osin pienempiä. Tämä havainto on hyvin linjassa luvussa 3.1 mainitun korpuksen kattavuussuositusten kanssa (clunits-äänelle 1000 lausetta, clustergenille 500).

4.2 Lopullisen version tekeminen

Lopullista valintaa tehdessä oli syntetisaattorille tarkoitetut sanoitukset valmiit. Sanoitukset luotiin SABLE-muotoon jotta ne voitaisiin ajaa helposti erissä skripteillä. Tässä käytettiin hyväksi demoja varten tehtyjä skriptien runkoja. Vaikka difoni- ja clustergen-äänet oli tässä vaiheessa jo periaatteessa hylätty, niillä kuitenkin tehtiin vielä syntetisoinnit. Tällä haluttiin vielä vahvistaa demojen suhteen tehdyt päätökset äänen lopullisessa kontekstissa.

Tässä vaiheessa myös syntetisaattoriäänen ympäristö eli kappale, jossa ääntä oli tarkoitus käyttää oli pohjiltaan valmis. Kappaleesta tehtiin raakaversiot siten, että jokaisella äänellä syntetisoidut sanoitukset asetettiin lopulliselle paikalleen kappalepohjaan. Tämän jälkeen äänien sopivuutta arvioitiin kuuntelemalla kappaleen eri versioita. Lopullinen valinta ja äänen optimointi oli prosessina iteratiivinen jonka takia tämä analyysikin hieman paloittainen.

Jokaisella synteesiäänellä tehdyt kappaleiden raakaversiot kuunneltiin läpi ja testimateriaalilla tehdyt havainnot olivat yhteneväisiä myös tässä kontekstissa. Lausekeleikkausäänet koettiin edelleen tarkoitukseen parhaiksi, koska puheen selkeä luonnollisuus toi oman, uuden mausteensa elektronisen soundin täyttämään loppuosaan. Clustergen-äänien ”pörisevä” soundi ei toiminut hyvin yhteen kappaleen kanssa, sillä se ei lopulta tullut tarpeeksi hyvin esille taustalla olevasta bassolinjasta ja syntetisaattoritaustasta vaikka puheesta saikin selvää.

Difoniäänessä taas oli useita ongelmia, äänen ymmärrettävyys oli huonoin, soundi ei ollut erityisen kiinnostava. Lisäksi puhujan tunnistettavuus oli edelleen kaikista synteesiäänistä huonoin, joten tämä hylättiin lopullisesti.

Näin ollen jatkoon otettiin äänet ARCTIC_clunits ja DJBB_clunits. Molempien äänien lauseiden ajoitusta koitettiin säätää käsin jonkin verran ja kuuntelukokeita jatkettiin näiden välillä.

DJBB_clunits osoittautui lopulta prosodialtaan ja erityisesti puhenopeudeltaan selkeästi ylivoimaiseksi tähän tarkoitukseen. Sikäli ei yllätys koska lopullinen syntetisoitavassa materiaali oli rytmitykseltään samankaltaista kuin korpuksen sisältö. Tämä on jälkikäteen tietysti selvää ja toisaalta voidaan kyseenalaistaa ARCTIC-äänien tekemisen järkevyys. ARCTIC-äänet tehtiin siksi, koska lyriikat eivät olleet alkuvaiheessa tiedossa ja periaatteessa niiden tyylilaji olisi voinut poiketa entisestä. Toinen syy ARCTIC-äänien tekoon oli yksinkertaisesti kokeilunhalu ja halu saada mahdollisimman paljon vertailuaineistoa jotta lopullinen päätös on helpompi tehdä.

Testimateriaalin ja varsinaisten sanoitusten välillä oli jonkin verran vaihtelua äänien keskinäissä suhteissa. ARCTIC-äänien liitokset ja sanojen luonnollisuus oli molemmilla materiaaleilla kuultavasti parempi, mutta lopullisessa materiaalissa ero oli pienempi. Ainakin osasyynä tähän on se, että syntetisoitava materiaalin oli samalta kirjoittajalta kuin DJBB-korpus itsessään.

Lopulta DJBB-korpuksen prosodinen paremmuus koettiin tärkeämmäksi kuin synteesin foneettinen laatu. Tämä on toisaalta yllättävää mutta toisaalta alkuperäisen suunnitelman mukaista – tarkoitus ei ollutkaan saada täydellisen luonollista syntetisaattoria. Lopulliseksi ääneksi valittiin siis DJBB_clunits. Erityisesti asian ratkaisi puheen nopeus, syntetisaattorin tuntuu huomattavasti vakuuttavammalta, koska tältäkin osin mallinnetaan alkuperäistä puhujaa ja synteesiääni on luontevaa jatkoa kappaleen muille, äänitetyille sanoituksille.

Clunits-synteesimoottorissa ei ole minkäänlaista tukea puhenopeuden muuttamiselle, mutta ARCTIC_clunits-äänellä tehtyä materiaalia koitettiin myös ajaa nopeuttaa jälkikäteen signaaliprosessoinnilla. Näin saatu lopputulos koettiin kuitenkin huonommaksi kuin DJBB_clunits vaikka puhenopeus saatiinkin sopivaksi. Signaalin nopeuttamiseen tällä metodilla jouduttiin käyttämään melko suurta kerrointa (yli 50%) jotta puhenopeus saataisiin suunnilleen vastaavaksi ja tämä aiheutti kuuluvia artefakteja ääneen, erityisesti vokaaleihin joten tästä luovuttiin.

Vaikkei päämääränä ollutkaan synteesin täydellisyys, niin silti DJBB_clunits-äätä hiottiin vielä jonkin verran tiettyjen ongelmallisten sanojen ja liitosten osalta. Tarkoituksena oli pitää synteesin laatu konsistenttina, pienet jatkuvat virheet hyväksyttiin mutta suuret poikkeamat koettiin häiritsevinä joten niistä haluttiin eroon. Erityisesti haluttiin saada puhe mahdollisimman ymmärrettävänä, ensimmäisessä versiossa oli muutama sana jotka äännetään täysin väärin eikä niistä ole helppo saada selvää.

Syntetisoitava teksti ajettiin yhtenä palana. Käytetty lyriikka oli tyyliltään DJBB-korpusta vastaava rap-lyriikkaa, jonka lauserakenne on muotoa (lause) (tauko) (lause). Synteesimoottorin <break />-tagin tuen puute kierrettiin sillä, että lauseosien väliin laitettiin aina pilkku. Lisäksi kaikki virkkeet lopetettiin huutomerkkiin, tällä pyrittiin pitämään intonaatio mahdollisimman korkeana loppuun asti. Tämän vaikutus ei ollut kovin iso, koska loppujen lopuksi korpuksessa ei ollut kovin montaa huutomerkkiin päättyvää lausetta. Jälkikäteen arvioituna olisikin ollut tarpeellista rytmittää korpuksen lauseet pilkun avulla vieläkin selvemmin, jolloin alkuperäisen tekstin lauseprosodiaa olisi voitu ohjata ja saatu mallinnettua tarkemmin.

Ensisijassa ongelmat johtuivat odotetusti siitä että osien segmentointi oli hiukan pielessä ja konkatenoinnissa tuli huonoja liitoksia. Näitä korjattiin ensin tut-

kimalla Festivan synteesiaikana luomaa puhunnostiedostoa. Tästä päästään käsiksi siihen, mitä yksiköitä synteesimoottori valitsee annetulle tekstille. Näin löydettiin oikea kohta, josta voitiin tutkia segmentoinnin paikkaansapitävyyttä. Korjaukset segmentointitietoihin tehtiin jälleen Wavesurfer-ohjelmistolla, jossa on tuki Festivalin käyttämälle segmentointitiedostoformaatille.

Clunits-äänessä segmentoinnin virheet olivat huomattavasti pienempiä kuin edellä mainitut difoniäänen virheet. Kuitenkin tässäkin virheet olivat tarpeeksi ratkaisevia aiheuttamaan konkatenointiin häiriöitä. Segmentointia parantamalla saatiin häiritsevimmät liitokset korjattua.

Lisäksi muutamissa kohtaa huomattiin liitoksissa olevan perustaajuuksissa hyppyjä. Tämä taas viittasi perusjaksomerkintöjen olevan joissain kohtaa väärin. Perustaajuusmerkintöjä korjattiin vastaavasti kuten segmentointiakin alussa muutamista kohtaa. Kuitenkin perustaajuuksien korjausta ei koettu niin tarpeelliseksi kuin segmentoinnin korjaamista, koska tämä ei vaikuttanut yhtä paljon puheen ymmärrettävyyteen.

Lopulta viimeiselle versiolle asetettiin kaksi tavoitetta. Ensinnäkin puheen ymmärrettävyys haluttiin maksimoida niin, että jokainen sana tulisi lausuttua mahdollisimman luonnollisesti. Tämä koettiin erityisen tärkeäksi, koska syntetisaattorin lukema teksti päättää kappaleen ja luetun tekstin sanoma oli olennainen kappaleen kokonaisuuden kannalta. Toinen tärkeäksi koettu asia oli sanojen ja lauseiden rytmi. Tämä haluttiin saada vastaamaan laulajan muuta materiaalia – tämä yhdessä samankuuloisuuden kanssa tekisi äänen kuulijalle välittömästi tunnistettavaksi.

Kriteereista johtuen huonojen liitoksien korjailussa keskityttiin sellaisiin paikkoihin, jossa liitokset vaikeuttivat sanojen tunnistusta. Erilaisia versioita tehtiin lopullisella äänellä noin viisi kappaletta, joista jokainen korjaa muutaman huonon liitoksen edellisestä. Korjaukset eivät aina tosin olleet aivan täysin onnistuineita, eräissä tapauksissa huonoista liitoksista kyllä päästiin, mutta lopputuloksena oli esimerkiksi hieman liian lyhyitä vokaaleita. Lopussa keskityttiin vain muutamaan ongelmalliseen lauseeseen, esimerkiksi sanat ”angry”, ”responsiblity” sekä ”playgroud” lauseen alussa oli hankala saada kuntoon.

Sen sijaan puheen perustaajuutta ja intonaatiota ei pyritty parantelemaan juuri ollenkaan. Näillä kriteereillä tähdättiin siihen, että sanoitukset olisivat ymmärrettäviä, jota pidettiin tärkeänä kappaleen teeman kannalta, mutta kuitenkin siten, että ääni oli tarpeeksi epäluonnollinen. Tällä haluttiin pitää kiinni siitä, että ääni kuulostaa edelleen tunnistettavasti puhesynteesiltä, alkuperäinen tarkoitus oli saada ääni kuulostamaan synteettiseltä sillä syntetisaattorin ei tarvinnut korvata oikeaa puhujaa vaan täydentää sitä.

Jotta ymmärrettävyys saatiin maksimoitua, joitain sanoja syntetisoitiin lausekontekstin ulkopuolella eli syntetisaattoria ajettiin sana kerrallaan. Tämä auttoi DJBB_clunits-syntetisaattoria suosimaan oikeaa yksikköpituutta perustaajuuden ja intonaation kustannuksella. Yksiköiden paremmuuden koettiin parantavan ymmärrettävyyttä ja liitoksista sekä prosodiasta priorisoitiin alemmaksi. Sama vaikutus olisi voitu saada säätämällä synteesiäänen parametreja ja erityisesti liitos- ja yksikkökustannusten painokertoimia, mutta tähän ei haluttu ryhtyä kuin vasta viimeisenä keinona.

Yksittäin syntetisoidut sanat liitettiin lausesynteesin käsin. Vastaavasti myös pidempiä säkeitä pilkottiin eri, jotta säkeen keskellä olevaa taukoa saatiin painotettua paremmin. Nämä pätkät sitten yhdistettiin yhteen äänitiedostoon, johon säkeiden väliset ajoitukset säädettiin käsin. Säkeiden väliin tehtiin noin 1,5 – 3 sekunnin tauko, säkeiden pituuden ja keskinäisen riippuuvuden mukaan. Viimeiset lauseet ovat yleisesti kauempana toisistaan, jolla haluttiin luoda häivyttävä tunnelma. Tauotus tehtiin täysin käsin kuuntelemalla ja kokeilemalla erilaisia taukojen pituuksia.

Sana ”responsibility” oli lopulta liian hankala DJBB_clunits-äänelle eikä siitä saatu järkevässä ajassa ymmärrettävää synteesiä tehtyä. Lopulliseen versioon ongelma korjattiin lievällä huijauksella, tämä yksittäinen sana korvattiin lauseessa ARCTIC_clunits-äänellä syntetisoidulla sanalla. Korvattua sanaa nopeutettiin hieman SoundForge-ohjelmiston nopeutustyökalulla mutta hienovaraisesti. Sanan erilaisen alkuperän voi erottaa tarkkaan kuuntelemalla, mutta musiikin seassa eroa ei huomaa.

5 Yhteenveto ja jatkokehitys

Lopputuloksena tehdyt syntetisaattoriäänet olivat melko vakuuttavia tasoltaan. Erityisesti ARCTIC-korpuksen pohjalta tehdyt äänet olivat peruslaadultaan hyviä. Kokonaisuudessaan äänien luominen oli työlästä ja vaati tarkkuutta ja huolellisuutta. Lisäksi prosessissa tarvittiin perehtyneisyyttä kieliteknologiaan ja -tieteeseen, signaalinkäsittelyyn sekä myös äänitystekniikkaan. Kuitenkin koska äänet tehtiin kielelle, jolle kaikilla syntetisaattorimoottoreilla oli jo tehty runsaasti erilaisia ääniä niin prosessista löytyi hyvää dokumentaatiota sekä tieteellisistä julkaisuista, että Festivalin postituslistan arkistoista.

Suurin jatkotutkimuksen ja -kehityksen tarve olisi syntetisaattorin saaminen ilmaisuvoimaisemmaksi ja helpommin kontrolloitavaksi tältä osin. Osaksi tätä voitaisiin saada paremmaksi, mikäli käytetyissä moottoreissa olisi ollut paremmat mahdollisuudet parametrien ohjaukseen. Toisaalta mikäli ilmaisua olisi haluttu kehittää, olisi myös korpusten oltava monipuolisempia, tässä työssä käytetyt korpukset äänitettiin loppujen lopuksi melko monotonisesti.

Tutkimusta syntetisaattoreiden ilmaisumahdollisuuksien kehittämisestä on tehty äänien tekemisen jälkeen ja erityisesti HTS-syntetisaattorimoottorille on saatavissa erilaisia kantoja, joilla pystytään syntetisoimaan vihaista puhetta [13].

Toinen pääasiallinen puute oli tilastollisen parametrisynteesin herätesignaali. Tilastollisessa synteesissä on olemassa pohja hyvälle synteesiäänelle ja tehtyjen äänien muokkaaminen käyttämään parempaa herätettä uudempien tutkimusten pohjalta olisi täysin mahdollista. Näin voitaisiin päästä tilastollisella parametroinnilla huomattavasti luonnollisempaan lopputulokseen ja ääni olisi näin monikäyttöisempi kuin nyt.

Syntetisaattorin käyttämistä yleisön edessä harkittiin, mutta tämän toteuttamiselle ei löytynyt sopivaa ajankohtaa. Tätä varten olisi tarvinnut luoda tarkoitukseen sopiva käyttöliittymä, jota kehitettiin hieman ideatasolla. Spontaanissa konserttitilanteessa toiseksi ongelmaksi olisi noussut clunits-syntetisaattorin hitaus, DJBB-kannalla synteesi tässä työssä käytetyllä tietokoneella oli hieman reaaliaikaa hitaampaa. Lyhyissä lauseissa tämä ei haittaa mutta pidempien puhunnosten syntetisointi on kohtuuttoman hidasta. Prosessoreiden vääjämätön nopeutuminen on tosin korjannut tätä ongelmaa jonkin verran ja synteesin käyttämistä yleisön edessä harkitaan taas.

Alkuperäisistä tavoitteista ainoastaan laulava syntetisaattori ei realisoitunut ollenkaan. Festivalin difonimoottoriin on toteutettu yksinkertainen laulava synteesi, tätä kyllä kokeiltiin mutta laulumoodi on toteutettu enemmänkin demotarkoitukseen kuin oikeaan käyttöön. Mahdollisuutta tehdä ääni Flinger-moottorille tutkittiin, mutta tälle ei ollut saatavilla mitään dokumentaatiota uuden äänen tekemiselle, joten tätä ei voitu toteuttaa. Laulavan synteesin tutkimusta on jatkettu muilla tekniikoilla ja HTS-projektissa on tehty muutamia esimerkkiääniä laulavasta synteesistä [13].

Ilmeikkyyden ja laulavan synteesin ansiosta looginen seuraava askel olisi HTS-äänien tekeminen olemassa olevilla aineistoilla. Vertailun vuoksi olisi myös mielenkiintoista tehdä lausekeleikkaussyntetisaattori Festivalin uudemmalla Multisyn-

moottorilla.

Nykyisiä ääniä olisi mahdollista tehdä vielä paremmiksi yleissyntetisaattoreiksi. Äänitysten automaattinen segmentointi onnistui keskimäärin hyvin, mutta niissä on vielä korjattavaa mikäli haluttaisiin maksimoida syntetisaattoreiden tekninen laatu. Lisäksi lausekeleikkaussynteesin kustannusfunktioiden painokertoimia olisi mahdollista koittaa optimoida. Tekninen laatu voisi hieman vielä parantua, mikäli synteesitietokannoissa käytettäisiin alkuperäisiä, 44,1 kHz näytteenottotaajudella tehtyjä nauhoituksia, tässä työssä käytetyt äänet on tehty 16 kHz:iin uudelleennäytteistettyjen versioiden pohjalta.

DJBB-korpus sisälsi yhtyeen kaikki sanoitukset eli periaatteessa se oli jo kertaalleen äänitetty. Olisi periaatteessa ollut mahdollista käyttää levytyksiä varten tehtyjen äänitysten raakaversioita syntetisaattorin luomiseen. Lopputulos olisi kuitenkin luultavasti ollut hyvin epäkonsistentti sillä alkuperäiset äänitykset ovat usean vuoden ajalta (1998 – 2006) ja tehty vaihtelevissa ympäristöissä erilaisilla laitteistoilla. Lisäksi alkuperäisten äänitysten tyylilaji ja puhenopeus vaihtelee hyvin paljon, materiaaliin kuuluu suoraa laulua, puhetta sekä äärimmäisen nopeaa rap-lausuntaa joten uusien, tasaisten äänitysten teko koettiin tarpeelliseksi. Toisaalta olisi mielenkiintoista tehdä ääni näinkin vaihtelevasta materiaalista, mutta suurimpana ongelmana olisi saada puhunnokset kuvattua jotenkin järkevästi synteesiäänen luontia varten, toisin sanoen jokaiselle korpuksen lauseelle tulisi olla jonkinlainen tyylilajiinformaatio olemassa.

DJBB-korpuksen ortografiaa olisi voinut muokata hieman paremmaksi. Sanoitukset otettiin tietokannasta, jonne ne oli syötetty useiden vuosien aikana ja tekstien välimerkkien käyttö ei ollut täysin konsistenttia. Jälkikäteen ajateltuna ainakin säkeiden keskellä olevat tauot olisi kannattanut merkitä aina pilkulla jotta lähdemateriaalin tämä aspekti olisi tullut äänissä vielä selkeämmin esiin. Nyt käytetyssä korpuksessa pilkkuja käytettiin identifioimaan taukoja melko satunnaisesti eikä aina englannin oikeinkirjoitussäännösten mukaisesti. Lisäksi painotusten merkkaaminen korpukseen eksplisiittisesti voisi auttaa ainakin clunits-moottoria paremman prosodian luomiseen etenkin mikäli vastaavia merkintöjä voitaisiin käyttää syntetisoitavan tekstin sivuinformaationa.

Työn tavoitteet saavutettiin lopulta hyvin: tarkoitus oli tehdä uusi synteesiääni ja käyttää tätä levyllä. Tässä siis onnistuttiin. Käytetty synteesiääni ei ole teknisesti täysin nykyaikainen eikä perinteisillä kriteereilla mitattuna kaikkein paras. Kuitenkin lausekeleikkaussynteesin perusominaisuudet eli luonnollinen soundi, ymmärrettävyys sekä laadukkaiden ja huonompien jaksojen vaihtelu tuo oman ääneen kontrastinsa, joka kuulostaa onnistuneelta sekä kappaleessa itsessään sekä laajemmassa genrekontekstissa.

Viitteet

[1] Don Johnson Big Band. Record Are Forever. Beat Back / Universal Music 270 433-7, Toukokuu 2009.

[2] Dennis Klatt. Review of Text-to-Speech Conversion for English. *Journal of the Acoustical Society of America*, 82(3), 1987. URL http://americanhistory.si.edu/archives/speechsynthesis/dk_737a.htm.

[3] The Sounds of Fighting Men, Howlin' Wolf and Comedy Icon Among 25 Named to the National Recording Registry. Library of Congress. URL http://www.loc.gov/today/pr/2010/10-116.html. Viitattu 25.7.2011.

[4] Dave Tompkins. *How to Wreck a Nice Beach*. Stop Smiling Books, 2010.

[5] Sue Sillitoe ja Matt Bell. Recording Cher's Believe. URL http://www.soundonsound.com/sos/feb99/articles/tracks661.htm.

[6] Kanye West. 808s & Heartbreak. Def Jam B001244102, Marraskuu 2008.

[7] Robotic voice effects. Wikipedia. URL http://en.wikipedia.org/wiki/Robotic_voice_effects. Viitattu 26.7.2011.

[8] Mac Randall. *Exit Music: The Radiohead Story*. Delta, 2000.

[9] Man Or Astro-Man? Made From Technetium. Touch & Go Records 180, Syyskuu 1997.

[10] Man Or Astro-Man? Eeviac: Operational Index & Reference Guide, Including Other Modern Computational Devices. Touch & Go Records 189, Huhtikuu 1999.

[11] Douglas O'Shaughnessy. *Speech Communications*. IEEE Press, toinen painos, 2000.

[12] Sami Lemmetty. Review of Speech Synthesis Technology. Diplomityö, Teknillinen Korkeakoulu, 2000.

[13] HMM-based Speech Synthesis System (HTS). URL http://hts.sp.nitech.ac.jp/. Viitattu 24.7.2011.

[14] Festival Speech Synthesis System. URL http://www.cstr.ed.ac.uk/projects/festival/. Viitattu 16.7.2011.

[15] Alan W. Black ja Kevin A. Lanzo. Building Synthetic Voices. URL http://festvox.org/bsv/. Viitattu 15.5.2011.

[16] Flinger. URL http://www.cslu.ogi.edu/tts/flinger/. Viitattu 24.7.2011.

[17] Alan W Black ja Paul Taylor. Automatically Clustering Similar Units for Unit Selection in Speech Synthesis. *Eurospeech97*, Rhodos, Kreikka, 1997.

[18] Florian Schiel, Christoph Draxler, Angela Baumann, Tania Ellbogen ja Alexander Steffen. The Production of Speech Corpora, Münchenin yliopisto, 2004. URL http://www.phonetik.uni-muenchen.de/Forschung/BITS/TP1/Cookbook/.

[19] A.J. Hunt ja A.W. Black. Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database. *ICASSP 1996 Proceedings*, sivut 373–376. IEEE, 1996.

[20] E. Eide, A. Aaron, R. Bakis, W. Hamza, M. Picheny ja J. Pitrelli. A Corpus-Based Approach to <ahem /> Expressive Speech Synthesis. *5th ISCA Speech Synthesis Workshop*, 2004.

[21] J. Tao, Y. Kang ja A. Li. Prosody Conversion from Neutral Speech to Emotional Speech. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4): 1145–1154, 2006.

[22] Alan W Black. CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling. *INTERSPEECH 2006 – ICSLP*, Pittsburgh, Yhdysvallat, Syyskuu 2006. URL http://www.cs.cmu.edu/~awb/papers/is2006/IS061394.PDF.

[23] H. Kawahara. STRAIGHT, Exploitation of the Other Aspect of VOCODER: Perceptually Isomorphic Decomposition of Speech Sounds. *Acoustical Science and Technology*, 27(6):349–353, 2006.

[24] Y. Stylianou. On the Implementation of the Harmonic Plus Noise Model for Concatenative Speech Synthesis. *ICASSP 2000 Proceedings*, osa 2, sivut II957–II960. IEEE, 2000.

[25] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio ja P. Alku. HMM-based Speech Synthesis Utilizing Glottal Inverse Filtering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(1):153–165, 2011.

[26] Tino Ojala. Auditory Quality Evaluation of Present Finnish Text-to-speech Systems. Diplomityö, Teknillinen Korkeakoulu, 2006.

[27] Thierry Dutoit. A Short Introduction to Text-to-Speech Synthesis. URL http://tcts.fpms.ac.be/synthesis/introtts_old.html. Viitattu 16.7.2011.

[28] SABLE: A Synthesis Markup Language. URL http://www.bell-labs.com/project/tts/sable.html. Viitattu 16.7.2011.

[29] CMU. CMU_ARCTIC speech synthesis databases. URL http://festvox.org/cmu_arctic/index.html. Viitattu 15.5.2011.

[30] SoX – Sound eXchange. URL http://sox.sourceforge.net/. Viitattu 16.7.2011.

[31] F. Malfrère ja T. Dutoit. High Quality Speech Synthesis for Phonetic Speech Segmentation. *Eurospeech97*, Rhodos, Kreikka, 1997.

[32] Wavesurfer. URL http://sourceforge.net/projects/wavesurfer/. Viitattu 18.7.2011.

[33] K. Prahallad, A.W. Black ja R. Mosur. Sub-phonetic Modeling for Capturing Pronunciation Variations for Conversational Speech Synthesis. *ICASSP 2006 Proceedings*. IEEE, 2006.

[34] Kraftwerk. Radio-Activity. EMI 581684, Lokakuu 1980.

[35] Joy Division. Closer. Factory FACT 25, Kesäkuu 1975.

[36] U2. Joshua Tree. Island 422-842298-2, Maaliskuu 1987.

[37] The Perceptionists. Black Dialogue. Definitive Jux Records 103, Maaliskuu 2005.

[38] Don Johnson Big Band. Breaking Daylight. Beat Back / Universal Music 038 592-2, Toukokuu 2003.

[39] Don Johnson Big Band. Don Johnson Big Band. Beat Back / Universal Music 985 434-1, Huhtikuu 2006.

[40] Barack Obama. Virkaanastujaispuhe, Tammikuu 2009. URL http://news.bbc.co.uk/2/hi/americas/obama_inauguration/7840646.stm. Viitattu 16.7.2011.

[41] Donald Rumsfeld, Helmikuu 2002. URL http://actuspurunen.blogspot.com/2005/11/donald-rumsfeld-poet-of-bush.html. Viitattu 16.7.2011.

[42] Martti Ronkainen. Äänien vertailuun käytetty demomateriaali. URL http://430am.fi/synth/.

A Synteesiäänien vertailuun käytetty materiaali

Synteesiäänien vertailua varten syntetisoitiin kaikilla äänillä osia seuraavien musiikkikappaleiden sanoituksista sekä artikkeleita:

- Joy Division: Heart & Soul [35]
- U2: With or Without You [36]
- The Perceptionists: Black Dialogue [37]
- The Perceptionists: Breathe in the Sun [37]
- Don Johnson Big Band: Broken Daylight [38]
- Don Johnson Big Band: No. 2 at the Hamburg Concept [39]
- Barack Obaman virkaanastujaispuhe [40]
- Donald Rumsfeldin "Known Unknowns"-katkelma [41]

Osa syntetisoiduista näytteistä on saatavilla tämän työn kumppanisivustolla [42].

B Syntetisoitavan materiaalin formaatti

Puhesynteesin ajamiseen käytettiin sable-tiedostoja. Tässä liitteessä on esimerkki siitä, miten luovalla välimerkkien käytöllä saatiin syntetisaattorin prosodiaa muokattua haluttuun suuntaan koska synteesimoottorin tuki puhetyylin kuvaukseen käytetyille tageille oli vajaa. Tässä käytetään merkintakielenä SABLEa [28]. Itse teksti on ©BeatBack 2009, käytetty luvalla.

```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
      "Sable.v0_2.dtd"
[ ]>
<SABLE>
<SPEAKER NAME="djbb_us_mrt_johnson_clunits" GENDER="">

Being angry, for the sake of being angry!
Brutality, as a justification for greater brutality!
Loud talkers, refusing to listen, to quiet talk!
Pursuing profits, for personal pleasure, and calling it something else!
Acting like the past or the future, are different from the present!
Having children to feign responsibility!
Refusing to see governments, as an image of ourselves!
Trusting words too much!
Equating charismatic with credible!
Doing less than what's required, when requirements are low!
Listening to music with your eyes open!
Supporting war!
Not being able to look at pictures of dead people!
Politics as a playground, of the highest bidders!
Praising childhood, and stripping it away from children!
Decaffeinated coffee!
Giant supermarkets, with millions of tiny packages!
Food without origin!
Acting like responsibility is a burden!
Trying to kill pain with a chemical substance!
Diagnosing discomfort!
Gender roles as an explanation for anything!
Pretending to believe, as a form of consolation!
The thought of thoughts, after the brain has died!
Valuing life only when, it is valuable!
Technology as a substitute, and not a tool for understanding!
The failure to talk and listen to young people!
Arrogance masked as affection!
Violence and love, in the same sentence!
Giving off the air of infallibility!
```

Treating art as property!
Only making records, if records are forever!
If records are forever!!!

</SPEAKER>

</SABLE>