Kõnesüntees peidetud Markovi mudelitega

Magistritöö

Rainer Metsvahi

Tallinn 2012

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutiteaduse instituut

Sisukord

Annotatsioon

Käesoleva magistritöö käsitleb uusima kõnesünteesi meetodi - Markovi peitmudelite *(HMM - Hidden Markov Model)* temaatikaga. Antakse ülevaade selle meetodi üldisest põhimõttest ning eelistest teiste meetodite ees.

Magistritöö põhieesmärgiks on realiseerida piiramatu sõnavaraga eestikeelne Markovi peitmudelitel põhinev sünteeshääl kasutades sünteesisüsteemi Festival, Markovi peitmudelitel põhineva kõnesünteesi jaoks tehtud tarkvarapaketti HTS ning Eesti Keele Instituudist saadud kõnekorpust *eki_et_liisi*.

Töö on kirjutatud eesti keeles ning koosneb neljast peatükist. Esimene peatükk annab ülevaate Markovi peitmudelitel põhinevast kõnesünteesist ning ühest selle implementatsioonist. Teine peatükk pakub põgusa ülevaate eestikeelse kõnesünteesi hetkeolukorrast ning probleemidest. Kolmas peatükk kirjeldab piiramatu sõnavaraga Markovi peitmudelitel põhineva eestikeelse sünteeshääle *eki_et_liisi_hts* realiseerimist ning neljas peatükk hindab saadud tulemusi.

Töö lisaks on CD plaat, mille pealt on magistritöö eesmärgiks realiseeritud Markovi peitmudelitel põhineva eestikeelse sünteeshääle moodul sünteesimootorile Festival, samuti skriptid piiramatu sõnavaraga eestikeelseks kõnesünteesiks, mis teostatakse Eesti Keele Instituudilt saadud text2pho-nimelise tekstitöötlusmooduli abil. Lisaks on plaadil saadud sünteeshäälega tekitatud helifaile ning näiteid hääle treenimiseks kasutatud märgendatud lähteandmetest.

Abstract

This Master's Thesis is submitted to the Institute of Computer Science at Tallinn University of Technology. It provides a short overview of HMM (Hidden Markov Model) based speech synthesis and its advantages over other speech synthesis methods.

The aim of the thesis is to implement an HMM based synthetic voice for Estonian with unlimited vocabulary. This is achieved using the Festival Speech Synthesis System, HTS (the HMM-based Speech Synthesis System) and a voice database called eki_et_liisi, courtesy of the Institute of the Estonian Language.

This work is written in Estonian and is comprised of 4 chapters. In the first chapter, an overview of HMM-based speech synthesis is presented. The second chapter deals with the accomplishments in the field of Estonian speech synthesis. The third chapter describes the process of building a HMM-based synthetic voice for Estonian. The last chapter discusses the results.

The thesis is delivered with a compact disc that contains the implemented Estonian HMM-based voice for Festival and is accompanied by some scripts that provide unilimited domain synthesis using the created voice in Festival and with the help of text2pho - a text-processing module for Estonian that was created for the Estonian diphone-based speech synthesis. In additon, some synthesized waveforms and label files can be found on the compact disc.

Sissejuhatus

Kõnesüntess on protsess, mille käigus teisendatakse kirjalik tekst kõneks. Oma olemuselt on see aga üsna keeruline kui mitte isegi võimatu tegevus, sest tekstilise info ning helina edastatava info kasutusvaldkonnad on erinevad. Kirja pandud tekst ei saa iial edastada kõike seda, mida võimaldab kõne. Ning teistpidi, inimene, kes peaks kõnelemise asemel oma mõtte edastama kirjalikult, teeks seda enamasti hoopis teistviisi kui lihtsalt oma kõnet tekstiks dikteerides. Seega on kõnesüntees protsess, kus kõneks tuleb muuta mõte, mille sisu on antud, kuid mille ettekandmisviis mitte. See on kõnesünteesi olemuslik keerukus. Käesolev kirjatöö tegeleb aga kõnesünteesi tehniliste aspektidega, mille tulemuste hindamisel tasub teinekord meeles pidada valdkonna üldist konteksti.[5]

Möödunud kümnendi keskpaigast alates on hakanud tuntust koguma statistilisi mudeleid pruukiv kõnesünteesi meetod, mis põhineb Markovi peitmudelitel (*HMM - Hidden Markov Model*). Selle lähenemise eeliseks teiste sünteesimeetodite ees peetakse sünteeskõne head kvaliteeti ning paindlikkust, mis võimaldab senisest vähema vaevaga muuta sünteeskõne iseloomu. Võrreldes teiste meetoditega on Markovi peitmudelitel põhineval sünteeshäälel rohkem *hinge* või *iseloomu,* ehk teisisõnu, arvuti poolt genereeritud kõne kõlab rohkem lähedane päris inimese kõnele.

Mainitud kõnesünteesi meetod on andnud muljetavaldavaid tulemusi inglise ning jaapani keele jaoks [6]. Eesti keele jaoks on hetkel olemas difoonsünteesil põhinev kõnesüntesaator [16] , mis valmis ligi 10 aastat tagasi.

Antud magistritöö eesmärgiks on pakkuda ülevaade populaarsust koguvast Markovi peitmudelitel põhinevast kõnesünteesist ning realiseerida selle meetodiga piiramatu sõnavaraga eestikeelne sünteeshääl. Selle ülesande lahendamiseks on kasutada Eesti Keele Instituudist saadud kõnekorpus *eki_et_liisi* ning olemasoleva eestikeelse difoonidl põhineva kõnesüntesaatori tekstitöötlusmoodul *text2pho*. Sünteeshääle tekitamiseks kasutatakse sünteesisüsteemi Festival [9] ning Markovi peitmudelitel põhineva kõnesünteesi tarkvara paketti HTS (*HMM-based Speech Synthesis System*) [2].

1 Markovi peitmudelitel põhinev kõnesüntees

1.1 Tutvustus

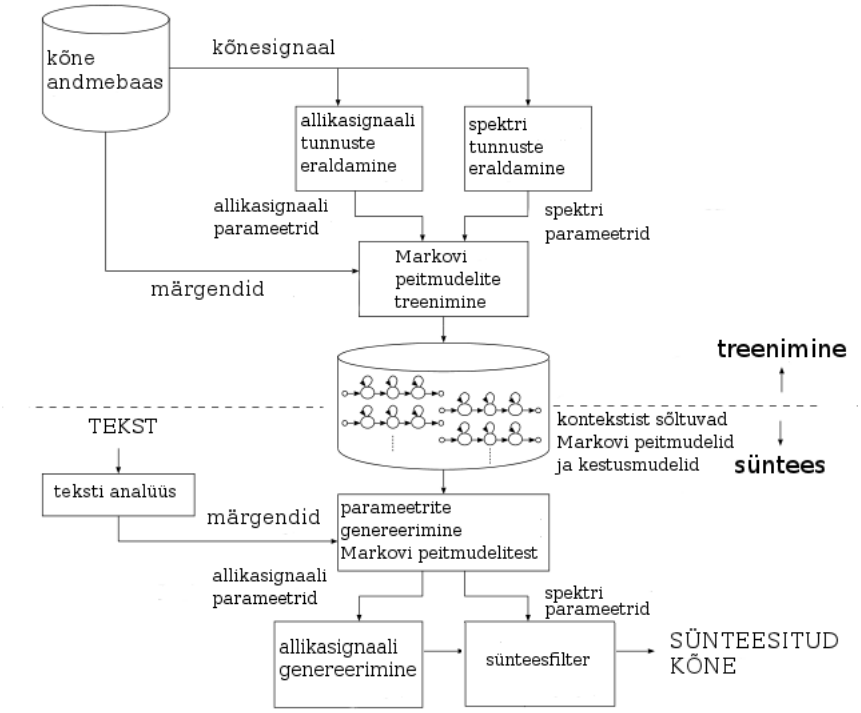
Tekst-kõneks sünteesi võib vaadelda kui kõnetuvastuse pöördprotseduuri. Iga tekst-kõneks süntesaatori eesmärk on võtta sisendiks sõnajada $w = \{w_1, ..., w_N\}$ ning tekitada sellest akustiline helisignaal $o = \{o_1, ..., o_T\}$. Tüüpiliselt määratakse keeletöötlusmooduli abil igale sõnajadale w kontekstuaalsed tunnused: rõhk, silbi ja fraasi piirid, sõnaliik jm., misjärel koostatakse w-le vastav kontekstist sõltuvate alamsõnade jada $u = \{u_1, ..., u_M\}$. Seejärel sünteesitakse u-st helisignaal o.[13]

Hetkel peetakse kõige levinumaks kõnesünteesi meetodiks üksuste valikul põhinevat sünteesi *(unit selection speech synthesis)*, mille korral moodustatakse sünteeskõne eelsalvestatud kõnesignaalide andmebaasist sobivate tükkide valimise ja kokkuliitmise teel. Selline lähenemine võimaldab tekitada kõrgekvaliteedilist kõnet, mille taset on raske ületada. Teisalt aga on üksuste valikul põhineva sünteesi teel saadud kõne piiritletud väga jäigalt kõneandmebaasis leiduvate konkreetsete kõnelõikude stiili ja esituslaadiga. Selleks, et saada erineva iseloomu, stiili ja emotsioonidega sünteeskõnet ning samas säilitada ka sujuvat ning kvaliteetset tulemust, on vaja salvestada väga suuri andmebaase [4]. Selliste andmebaaside koostamine on aga keeruline ning väga töömahukas ning kallis protsess.

Tänapäeval põhinevad enamik kõnesünteesi süsteeme suurel kõneandmete hulgal. Sellist lähenemist kutsutakse korpusepõhiseks kõnesünteesiks ning see võimaldab hüppeliselt parandada sünteeskõne kõla loomulikkust võrreldes varasemate enamjaolt reeglipõhiste sünteesisüsteemidega.

Üks enimlevinumaid lähenemisi korpusepõhisele kõnesünteesile baseerub üksuste valikul (*unit selection*). Selle lähenemise korral jagatakse kõneandmed väikesteks ühikuteks, nt. poolhäälik, häälik, difoon (kahe järjestikuse hääliku üleminek) või silp ning salvestatakse. Kõne sünteesiks valitakse kontekstist sõltuvate alamsõnade jadale vastavad helisignaali üksused. Kõnesignaalide andmebaasist üksuste valimisel kasutatakse kõige sobivama üksuste leidmiseks hinnafunktsioone, mis arvestavad seda, kui hästi mingi üksus sobib lingvistiliselt (üksuse asukoht lauses või sõnas - algus, lõpp; rõhu ja silbi info) (*target cost*) ja signaalitöötluse aspektist (põhitooni vastavus ning kontuuri silumine lauses) (*concatenation cost*) etteantud konteksti.[13]

Teine üha rohkem populaarsust koguv lähenemine on statistilisi parameetreid pruukiv Markovi peitmudelitel põhinev kõnesüntees, mida kirjeldab joonis 1.1. Selle lähenemise korral treenitakse kontekstist sõltuvad Markovi peitmudelid loomuliku kõne pealt ning sünteeskõne moodustamiseks kasutakse saadud treenitud mudeleid. Selline süsteem pakub võimaluse modelleerida erinevate omadustega hääli ilma, et peaks salvestama väga suuri andmebaase.



Joonis 1.1: Markovi peitmudelitel põhinev kõnesünteesi süsteem[4]

Süsteem koosneb kahest osast - treenimise osa ja sünteesi osa. Treenimise osa on sarnane protsessiga, mida kasutatakse kõnetuvastuses. Peamine erinevus on aga selles, et nii spektri tunnused (mel-kepsteri kordajad ning nende dünaamilised tunnused) kui ka allikasignaali tunnused(logaritmilised põhitooni sagedused ning nende dünaamilised tunnused) eraldatakse kõne andmebaasist ning nende abil treenitakse kontekstist sõltuvad Markovi peitmudelid, mis hõlmavad foneetilist, lingvistilist ning prosoodilist tasandit. Kõikidel Markovi peitmudelitel on lisaks veel oleku kestuse jaotustiheduse funktsioonid, mis kirjeldavad kõne ajalist mõõdet. Selle tulemusena modelleerib sünteesi süsteem spektri, allikasignaali ja kestuse ühtses Markovi peitmudelite raamistikus.[4]

Sünteesi osa kujutab endast kõnetuvastuse pöördoperatsiooni. Esmalt töödeldakse soovitud sisendtekst ning saadakse kontekstist sõltuvate märgendite jada. Seejärel koostatakse soovitud lausungi jaoks Markovi peitmudel, mis saadakse kontekstist sõltuvate Markovi peitmudelite liitmisel nii, et need vastaksid saadud mär-

gendite jadale. Seejärel lisatakse lausungi *(utterance)* Markovi peitmudelile oleku kestused, mis saadakse olekute jaotustiheduse funktsioonidest. Järgmiseks koostab kõne parameetrite genereerimise algoritm spektraalsed ja allikasignaali parameetrid, mille alusel genereeritakse sünteesfiltri abil lõpuks helisignaal.[4]

Markovi peitmudelitel põhineva kõnesünteesi süsteemi üheks kõige tähtsamaks omaduseks peetakse võimalust suhteliselt lihtsalt muuta väljundhääle iseloomu, kõnestiili ja emotsioonikust. Seda kõike annab teha sünteesfiltri parameetrite muutmise teel.

1.2 HTS

HTS *(HMM-based Speech Synthesis System)*[2] on HTS töögrupi poolt loodud Markovi peitmudelitel põhineva kõnesünteesi jaoks tehtud tarkvarapakett. Töögruppi kuuluvad teiste seas inimesed Nagoya Tehnoloogia Instituudist, Tokyo Tehnoloogia Instituudist, Edinburghi Ülikoolist, Tokyo Ülikoolist ning Carnegie Mellon Ülikoolist. HTS-i treenimise osa on realiseeritud kõnetuvastuse tarkvarapaketi HTK[3] muudetud versioonina ning seda jagatakse kui paika (*patch code*) HTK jaoks. Paika ise jagatakse vabavaralise litsentsi alusel, kuid peale paiga HTK-le paigaldamist tuleb alluda HTK kasutustingimustele, mis keelab levitamise ning ärikasutuse.

HTS-i sünteesi osa on realiseeritud erinevate skriptidena, mis kasutavad HTK vormingus treenitud mudeleid. HTS-il ei ole oma tekstitöötlusmoodulit, kuid ta ühtib Festivali sünteesi süsteemiga (alates versioonist 2.0) (inglise k., hispaania k., jt)[9], DFKI MARY tekst-kõneks süsteemiga (saksa k., inglise k., jt)[10], Flite+hts_engine (inglise k.)[11], Open JTalk (jaapani k.)[12]. HTS-iga tulevad kaasa erinevad näidisskriptid koos lähteandmetega (helifailid ning märgendatud tekstid), mille abil saab ise endale treenida nii ühe kõnejuhiga kui ka kõnelejast sõltuva inglise keelse hääle. Samuti on paketis veel lisaks skriptid portugali, jaapani ning laulva jaapani keelse hääle treenimiseks. Käesoleva magistritöö raames loodud eestikeelne sünteeshääl on samuti saadud kasutades HTS-i.

Sarnaselt teiste andmetel põhinevate kõnesünteesi meetoditega on HTS-il kompaktne keelest sõltumatu moodul: kontekstist sõltuvate tunnuste nimekiri, mille saab tekitada kasutades Festivali raamistikku. Seetõttu saab seda süsteemi hõlpsalt laiendada teiste uute keelte jaoks, kuigi HTS-i esmane versioon oli teostatud jaapani keelt silmas pidades. HTS-i käitusfaasi mootori *(run-time engine)* oluline eelis on tema suurus, mis jääb enamasti alla 1 MB (ilma teksti analüüsi moodulita).[14]

2 Eestikeelne kõnesüntees

2.1 Ajalugu

Kui maailmas on kõnesünteesiga tegeletud edukalt juba üle 60 aasta, siis Eestis on antud valdkonnaga tegemist tehtud veidi üle 30 aasta. Algselt olid uurimissuundadeks põhiliselt formansünteesi mudelite ja nende juhtimiseks tarvilike reeglisüsteemide loomine eesti-, vene- ja soomekeelse kõne sünteesiks. Eelmise aastakümne alguses valmis Eesti Keele Instituudi, Tallinna Tehnikaülikooli Küberneetika Instituudi ja OÜ Filosoft koostöös eestikeelse tekst-kõneks süntesaator[16]. Antud süntesaator baseerub Belgias Mons'i Ülikoolis loodud MBROLA (Multi-Band Resynthesis Overlap Add) sünteesimootoril, mis sünteesib kõnesignaali salvestatud loomulikust kõnest saadud difoonide kokkuliitmise teel. Lisaks sellele tegeldakse tänasel päeval ka Eesti Keele Instituudis eestikeelse kõnesünteesiga [17].

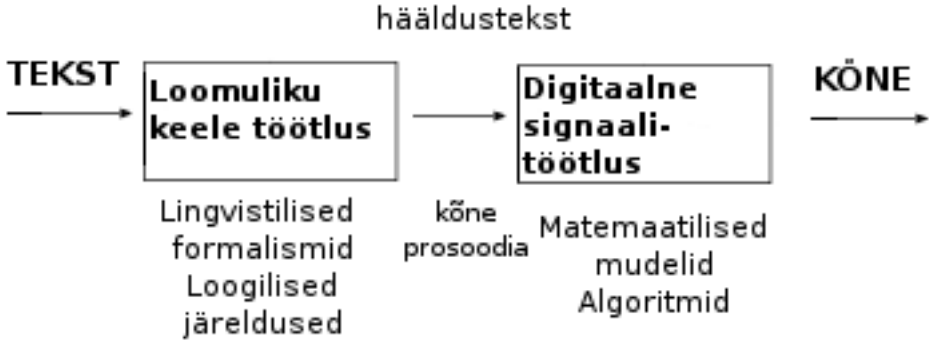
2.2 Kõnesünteesi meetodid

Analoogselt inimese lugemisprotsessiga vajab ka kõnesüntesaator loomuliku keele töötlemise moodulit, mis teisendab sisendteksti häälduste kstiks koos soovitud intonatsiooni ning kõnerütmiga. Digitaalne signaalitöötlusmoodul teisendab sisendis oleva sümbolijada kõneks. Loomuliku keele töötluse moodul annab teksti foneetilise kirjelduse ning seab paika ka kõne prosoodia. Üldjuhul sisaldab tekstitöötlus keele erinevaid lingvistilisi tasandeid: foneetikat, fonoloogiat, morfoloogiat, süntaksi ning semantikat.[36]

Joonisel 2.1 on toodud kõnesünteesi lihtsustatud mudel. Loomuliku keele töötluse moodul on keelespetsiifiline ning enamasti sõltumatu väljundkõne tekitamise moodulist, mida on võimalik realiseerida erinevate digitaalsete signaalitöötlusvahenditega.

Eesti keele jaoks on olemas mitu erinevat sünteeskõne tekitamise moodulit (joonisel 2.1 parempoolne kast):

- Formantsüntees, kus väljundkõne tekitatakse inimkõne häälikutele vastavate sageduste genereerimise teel. Selle sünteesimeetodi realiseerimiseks eesti keelele on hetkel arendamisel eSpeak sünteesaatori [37] moodul.
- Difoonsüntees, kus sünteeskõne moodustatakse eesti keele difoonide andmebaasi ning MBROLA sünteesimootori [15] abil, liites lühikesed salvestatud



Joonis 2.1: Üldistatud tekst-kõne sünteesi mudel [36]

loomuliku kõne tükid kokku soovitud lauseteks. Ühe meeshäälega difoonsüntesaator [16] on eesti keelele olemas alates 2000-ndate aastate algusest.

- Markovi peitmudelitel põhinev süntees, kus väljundkõne tekitamiseks treenitakse eelnevalt kõneandmebaasi peal statistilised mudelid ning sünteesimiseks otsitakse tekstitöötluse mooduli sisendile vastavad kõne spektri, allika ning prosoodilised parameetrid. Käesoleva magistritöö praktiliseks tulemuseks ongi antud sünteesimeetodi eestikeelse mooduli realiseerimine.

2.3 Tekstitöötlus

Kõnesünteesi sisendteksti keeletöötluse tulemusena teisendatakse ortograafiline tekst foneetiliseks tekstiks. Kuigi võib esmapilgul tunduda, et eestikeelne kirjatekst on kergesti hääldatav, siis kõnesünteesi jaoks arvutile eesti keele õiget hääldust õpetada on kohati väga keeruline, sest eesti keele ortograafia ei ole päris foneetiline[18]. Kirjapildis ei ole teine ja kolmas välde üldjuhul eristatavad (*Lapsed mängivad kooli juures / Lapsed lähevad kooli*). Samuti ei ole võimalik eristada palataliseeritud konsonante palataliseerimata konsonantidest (*Eesti keskmine palk on 850 eurot / See palk on kuus meetrit pikk*), lisaks ei ilmne pika üü diftongeerumine rõhutu silbi lühikese vokaali ees (*müüa* hääldatakse *müija*) ja palju muud.[18]

Lisaks välte ning palatalisatsiooni märkimisele leitakse lingvistilise töötluse käigus veel liitsõnapiirid, sõnarõhud ning silbipiirid, mis on vajalikud selleks, et sünteeskõnele anda sobiv prosoodia. Samuti teisendatakse tekstilisele kujule sisendis esinevad numbrid ja lühendid.[18]

Kirjalikes tekstides on suur hulk tähejadasid ning märke, mis pole ilme eelneva töötluseta süntesaatorile arusaadavad. Mittesõnade interpretaatori eesmärgiks on tundmatute jadade ning märkide analüüs ning nende lahtikirjutamine sõna- või tähthaaval. Mittesõnadeks on kõnesünteesi kontekstis tekstis esinevad lühendid (nt., jne., Ü.R.O, USA), kellaajad, numbrid ja arvud ning erimärgid (%, $, €) [19, 20].

2.4 Prosoodia modelleerimine

Sünteesitava teksti jaoks on vaja genereerida ka prosoodilised parameetrid - häälikute ja pauside kestused ning lause ja sõna meloodiakontuurid (põhitoonikontuurid). Häälikute kestused sõltuvad paljudest teguritest - üldine kõnetempo, lause pikkus ja tüüp, sõna asend lauses, sõna välde, silpide arv sõnas, silbi rõhulisus, silbi tüüp, hääliku asend silbis, naaberhäälikute kvaliteet. Eesti keele prorosoodiline andmebaas sisaldab ligi sada kontekstist sõltuvat häälikukestust. Lisaks reeglipõhisele mudelile, on loodud ka eestikeelse kõne statistiline kestusmudel [36]. Sõna põhitooni kontuuri genereerimisel lähtutakse silbi rõhulisusest - eesti keeles on rõhk tüüpiliselt esisilbil ja seda väljendab põhitooni kõrgem väärtus võrreldes teiste silpidega. Lause põhitooni kontuuri modelleerimisel lähtutakse lausetüübist, mille määravad ära kirjavahemärgid. Kõnemeloodia akustilisel analüüsil on leitud eri tüüpi lausete stiliseeritud meloodiakontuurid, mis eesti keele puhul on valdavalt langevad, st lause alguses on põhitooni sagedus kõrgem kui kause lõpus. Sünteesitava lause meloodia varieerub ajas langevate deklinatsioonijoonte vahel sõltuvalt lokaalsetest sõnarõhkudest ja lause pikkusest.[38]

3 Eestikeelne kõnesüntees Markovi peitmudelitega

3.1 Hääle treenimiseks kasutatud kõnekorpus

Eestikeelse sünteeshääle treenimiseks kasutasin ma Eesti Keele Instituudist saadud *eki_et_liisi* kõnekorpust, mis sisaldab 1928 *.wav* vormingus helifaili ning nendele vastavaid Festivali[9] sünteesimootori *.utt* vormingusse viidud lausungite märgendifaile (üks näide märgendifailist on käesoleva tööga kaasasoleva CD peal [21]). Helifailid on salvestatud stuudios monosignaalina, kvantimissagedusega 16 kHz, 32 bitti iga väärtuse kohta. Salvestatud lausungid on erineva pikkusega - lühemad vaid poole sekundi pikkused, kõige pikemad aga veidi üle 20 sekundi. Kokku on kõnekorpuses 3 tundi ja 40 minutit häälematerjali.

3.2 Sünteeshääle treenimine HTS-iga

3.2.1 Keskkonna ettevalmistus

HTS-i treeningkeskkonna viimase versiooni (2.2) kasutamiseks on vajalik paigaldada päris arvestatav kogus sõltuvusi, millest enamus tuleb lähtekoodist kompileerida:

- HTK-3.4.1
- HDecode-3.4.1
- libx11-dev, libsnack2, tcl (saadaval eelkompileeritud pakkidena)
- speech_tools
- festival
- SPTK-3.5 (Speech Signal Processing Toolkit)
- hts_engine

Pärast HTS-i ning tema sõltuvuste paigaldamist võtsin aluseks HTS-iga kaasas olnud ingliskeelse hääle treenimise demo[26]ning muutsin selle eki_et_liisi korpuse parameetritele vastavaks käsuga:

```
./configure \
```

```
--with-tcl-search-path=/usr/lib/tcl8.5 \
--with-fest-search-path=/m/f/festival/examples \
--with-sptk-search-path=/usr/local/SPTK/bin \
--with-hts-search-path=/usr/local/HTS-2.2beta/bin \
--with-hts-engine-search-path=/m/flite_hts_engine-1.03/bin \
SPEAKER=liisi DATASET=eki_et SAMPFREQ=16000 FRAMELEN=400 FRAMESHIFT=80
FREQWARP=0.42
```

HTS-i treenimise skript eeldab, et helisignaalid on ilma päiseta *.raw* kujul, seega tuli kõik *.wav* failid programmiga `sox` õigesse vormingusse viia.

3.2.2 Uue hääle treenimise etapid

Uue hääle treenimine võttis veidi üle 24 tunni arvuti protsessoriaega aega ning see käis alljärgnevate etappidena:

1. Helifailidest mel-kepsteri kordajate eraldamine
2. Helifailidest logaritmilise põhitooni info eraldamine
3. Saadud mel-kepsteri ning põhitooni infost HTK tarkvarapaketi vormingus treeningandmete tekitamine
4. Igale lausungifailile *(.utt* fail) hääliku tasemel ning kontekstist sõltuvate märgendite tekitamine. Neist esimene [22]sisaldab iga hääliku alguse ning lõpu ajatähiseid vastavas helifailis ning teine [23] sisaldab iga hääliku kohta lisaks veel teavet teda ümbritsevate häälikute kohta, silpide arvu selles sõnas, sõnade arvu lauses, milles sõna esineb, infot, kas tegu on helilise või helitu häälikuga, rõhu asukohaga lauses jne [25]
5. Markovi peitmudelite treenimine.
6. Helisignaalide genereerimine kontekstist sõltuvate märgendite alusel.

Viimase sammu tulemusena tekivad sünteesitud helisignaalid, mille sisendiks on põhimõtteliselt samad märgendifailid, mida kasutati ka treenimisel, ainult selle vahega, et sealt on eemaldatud häälikupiiride ajatähised [24].

Kuigi HTS-i viimane etapp on helisignaali genereerimine, pole see reaalajas toimivaks suvalise sisendteksti kõneks sünteesimiseks mõeldud. Nimelt vajab HTS helisignaali tekitamiseks eelpoolkirjeldatud kontekstist sõltuvaid märgendeid, millele otsitakse vastavad spektri ning allikasignaali kontekstist sõltuvad Markovi peitmudelid. Seega oskab HTS sünteesida vaid selliseid lausungeid, millele vastavad kontekstist sõltuvad Markovi peitmudelid tal treeningu käigus tekkisid. Selleks, et sünteesida lausungeid, mida treeningandmetes ei esinenud, tuleb sünteesida lausungile vastavad kontekstist sõltuvad Markovi peitmudelid. Selle saavutamiseks tuleb läbi

käia olemasolevad Markovi peitmudelid ning genereerida kontekstide sarnasuse alusel puuduvad mudelid. See võib aga aega võtta sõltuvalt treeningandmete suurusest mitmeid minuteid. Pigem on HTS-i tekitatud heliväljund mõeldud selleks, et näidata, millist kvaliteeti on põhimõtteliselt võimalik saavutada ning ülejäänud on jäetud hetkel edasiarendamiseks.

Treeningu tulemusel tekkis Markovi peitmudelitel põhineva kõnesüntesaatori jaoks naishääl *eki_et_liisi_hts* [29] .

3.2.3 Piiramatu sõnavaraga kõnesüntees Festivalis

Selleks, et HTS-is treenitud Markovi peitmudelitel põhinev hääl suvalist sisendteksti kõneks oskaks teha, kasutasin kõnesünteesi süsteemi Festival, millel on alates versioonist 2.0 HTS häälte tugi. Festivali jaoks on peale inglise keele olemas muuhulgas tugi ka hispaania, tšehhi, itaalia, vene ja soome keelele. Eesti keele tugi hetkel aga puudub. Kuna kõik kõnesüntesaatorid koosnevad alati kahest suurest osast - tekstitöötluse moodul ning kõne tekitamise moodul, siis tähendab see, et päris täielikku eesti keele tuge Festivalile veel teha ei saa. Sellele vaatamata on aga võimalik HTS-iga treenitud Markovi peitmudelite peal töötav eestikeelne hääl Festivali abiga reaalajas rääkima panna. Selleks tuli tekitada minimaalse funktsionaalsusega sisendteksti keeletöötlusmoodul, mille ülesanne on pakkuda Festivali HTS-i sünteesimootorile häälikute jada, millele vastavate kontekstist sõltuvate Markovi peitmudelite abil treenitud hääl kõnet hakkab tootma.

Alustuseks defineerisin Festivali jaoks häälikutähestiku ehk eesti kirjakeeles esinevatele tähtedele vastavad foneetilised ühikud. Selleks tuli kirjutada eesti keele spetsiifiline fail [27], millest järgnevalt on toodud näitena eesti täishäälikute defineerimine:

```
(defPhoneSet eki_et (

; phone vc vl vh vf vr ct cp cvoicing

(a + s 3 3 - 0 0 0) ;;

(a: + l 3 3 - 0 0 0) ;;

(e + s 2 1 - 0 0 0) ;;

(e: + l 2 1 - 0 0 0) ;;

(i + s 1 1 - 0 0 0) ;;

(i: + l 1 1 - 0 0 0) ;;

(o + s 2 3 + 0 0 0) ;;

(o: + l 2 3 + 0 0 0) ;;

(u + s 1 3 + 0 0 0) ;;

(u: + l 1 3 + 0 0 0) ;;

(q + s 2 3 - 0 0 0) ;; õ

(q: + l 2 3 - 0 0 0) ;;
```

```
(ae + s 3 1 - 0 0 0) ;; ä
(ae: + l 3 1 - 0 0 0) ;;
(c + s 2 1 + 0 0 0) ;; ö
(c: + l 2 1 + 0 0 0) ;;
(y + s 1 1 + 0 0 0) ;; ü
(y: + l 1 1 + 0 0 0) ;;
```

Iga hääliku järel on defineeritud hääliku tüüp (vokaal või konsonant), kas tegu on pika või lühikese häälikuga, ees- või tagavokaaliga; keskkõrge või madala vokaaliga ja kas vokaali moodustamisel toimub huulte ümardamine. Konsonantide korral kirjeldatakse analoogselt täishäälikutega sellised olulised tunnused nagu hääliku moodustamise asukoht, moodustusviis ning kas tegemist on helilise või helitu häälikuga.

Järgmiseks koostasin Festivali tarvis väga triviaalse leksikoni [28], mis paari lause jagu sõnade jaoks defineerib, millisteks foneetilisteks ühikuteks need tuleb teisendada. Täisväärtuslikus tekst-kõne sünteesisüsteemis vastutab selle eest teksitöötlusmoodul, kuid siiski võimaldab see piiratud sõnu juba Festivali käsureal sünteesida:

```
festival>(SayText "President Toomas Hendrik Ilves")
festival>(SayText "Juhtparteis voolab sularaha ojadena")
```

Selleks, et ikkagi oleks võimalik suvalist lauset saadud eestikeelse häälega tekitada, lisasin leksikoni väga lihtsad reeglid teisendustabeli jaoks, mis võimaldab välja kutsuda kõiki hääles olevaid häälikuid. Kuna leksikonis olevateks kirjeteks on tavaliselt sõnad, siis tuleb häälikute kaupa kõne sünteesimisel häälikud tühikutega eraldada. Seega on nüüd võimalik sünteesida suvalist eestikeelset lauset, kui soovitud tekst sisestada Festivali interpetaatorile tähekaupa:

```
festival>(SayText "t e r e h o m m i k u s t")
```

3.2.4 Eestikeelne naishääle tugi Festivalile

Nagu eelpool mainitud, toetab sünteesisüsteem Festival alates versioonist 2.0 HTS-i abil loodud hääli. Seega tuli treenitud eestikeelse hääle kasutamiseks Festivalis tekitada õige struktuuriga kataloogid ning uue keele jaoks vajalikud keeletöötlusmoodulid [30]. Selleks, et suvalises arvutis, kus Festival on paigaldatud, *eki_et_liisi_hts* häält kõnesünteesiks kasutada on vajalik kopeerida käesoleva tööga kaasas oleva CD plaadi pealt kataloog `festival/et` arvutis oleva Festivali kodukataloogi alamkataloogi `lib/voices`. Näiteks, kui arvutis asub Festival kataloogis `/usr/lib/festival`, siis tuleb CD plaadil olev kataloog `festival/et` kopeerida kataloogi:

`/usr/lib/festival/lib/voices/`. Seejärel tuleb käivitada Festivali interpretaator ning sisestada Festivalile käsk kasutada kõnesünteesiks *eki_et_liisi_hts* häält:

```
festival>(voice_eki_et_liisi_hts)
```

Seejärel on võimalik sünteesida kõnet viisidel, nagu eelmises punktis viidatud. Festivali käsk *SayText* suunab helisignaali otse arvuti helikaarti. Selleks, et sünteesitud kõne faili salvestada, tuleb Festivali interpretaatorile sisestada alljärgnevad käsud:

```
festival>(set! lause (Utterance Text "t e r e h o m m i k u s t"))

festival>(utt.synth lause)

festival>(utt.save.wave lause "tere_hommikust.wav")
```

Selle tegevuse tulemusel tekitatakse failisüsteemi helifail *tere_hommikust.wav*.

Selline protsess on aga veidi kohmakas ning eestikeelse sünteeskõne tekitamise hõlbustamiseks Festivalis kirjutasin mõned skriptid [31], millega saab otse linuxi käsurealt lause või faili kaupa eestikeelset kõnet sünteesida ning ka faili salvestada.

Nagu eelpool mainitud, ei ole hetkel Festivalile eesti keele jaoks tekstitöötlusmoodulit. Selleks, et käsurealt saaks ikkagi tavalist kirjakeelt sisestada, kasutasin ma Eesti Keele Instituudis tehtud eestikeelse difoonsüntesaatori [16] tarvis mõeldud keeletöötlusprogrammi *text2pho*, mis tekitab ortograafilisest sisendtekstist foneetiliselt märgendatud teksti. Kuna *text2pho* keeletöötluse väljund on eestikeelse difoonsüntesaatori spetsiifiline, siis kirjutasin pythoni skripti [32], mis selle Festivali *eki_et_liisi_hts* häälele kasutamiseks sobivaks teeb.

Selleks, et eelpool mainitud mugavusskripte kasutada, tuleb CD plaadil olev kataloog *tekstitöötlus* kopeerida kõvakettale ning seal olevates *.sh* laiendiga failides seadistada muutuja *FESTIVALI_ASUKOHT* viitama arvutis olevale Festivali kodukataloogile, näiteks:

```
FESTIVALI_ASUKOHT=/usr/lib/festival
```

Selle tulemusena on siis võimalik käsurealt sünteesida suvalist eestikeelset teksti:

```
$ ./loe_lause.sh "Ma olen sünteeshääl ning oskan eesti keeles rääkida"
```

Sünteeskõne faili salvestamiseks on käsk:

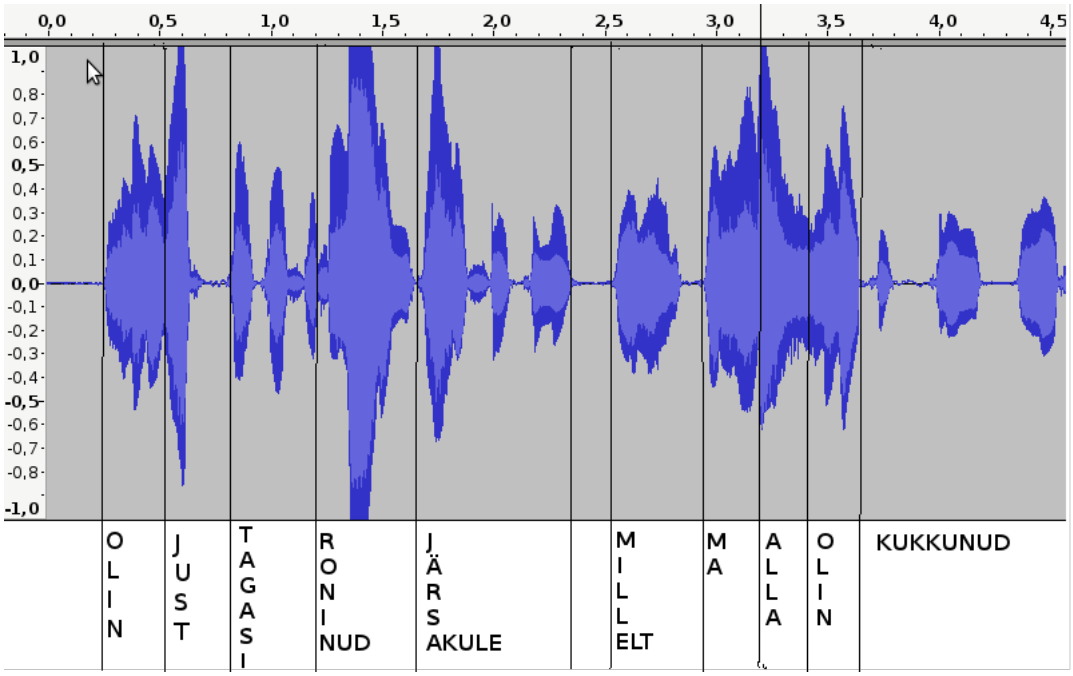
```
$ ./salvesta_lause.sh "Ma olen sünteeshääl ning oskan eesti keeles rääkida"

Sünteesitud kõne salvestati helifaili /tmp/tmp.9J9uGVfCMT.wav
```

Tekstifailist lugemiseks on samuti analoogsed skriptid *loe_fail.sh* ning *salvesta_fail.sh*.

4 Tulemus

Tarkvarapakett HTS võimaldab treenimisprotsessi seadistada enam kui 30 parameetri abil. Parameetrite vaikeväärtuste ehk inglise keelse hääle treenimise parameetreid aluseks võttes tekitatud seadetega osutus HTS-i abil sünteesitud *eki_et_liisi_hts* hääle sünteeskõne kohati väga ülevõimendatud. Joonisel 4.1 on toodud lausungi *"Olin just tagasi roninud järsakule, millelt ma alla olin kukkunud"* helilaine [33], mis on sünteesitud HTS-iga kasutades treeningparameetrite vaikeväärtusi.



Joonis 4.1: HTS-iga sünteesitud (globaalset variatiivsust kasutades) lause [33] helilaine

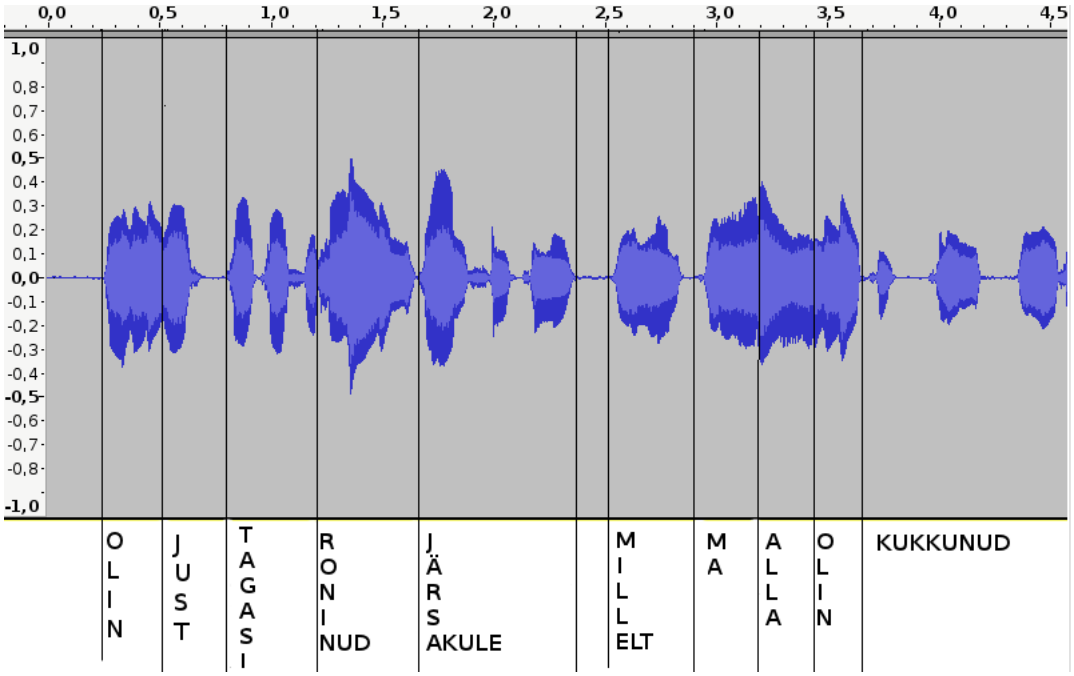
Helilainet vaadates ning helifaili kuulates on selge, et osade sõnade esimese silbi täishäälikud on tugevalt moonutatud. Lähemal uurimisel osutus, et eriti tugevalt on ülevõimendatud just nasaalhäälikud ning nendele eelnevad täishäälikud. Samas mingit selget seaduspära ülevõimendatud lausungi osade jaoks ma ei leidnud. Neid esines nii pikkades kui ka lühikestes lausungites; lause alguses, keskel ning ka lõpus.

4.1 Globaalne variatiivsus

Selleks, et nendest sünteeskõnes tekkivatest tehistest lahti saada, katsetasin hääle treenimise protsessi mitmete erinevate parameetritega uuesti. Katsetuste tulemusel selgus, et tehised kaovad ära, kui maha keerata *USEGV (global variance)* lipp. See lipp kontrollib seda, kas sünteesfiltris parameetrite genereerimisel kasutatakse globaalset variatiivsust või mitte.

Globaalse variatiivsuse abil on võimalik parandada sünteesiks kasutatava vokoodri parameetrite dünaamilisust ning liiga siledaid kontuure. Globaalse variatiivsuse rakendamise tulemusel saadakse üldjuhul loomulikumalt kõlavat sünteeskõnet. Varjupooleks aga ongi ülalpool välja toodud tehised ning moonutused, mis tekivad seetõttu, et sünteesfiltris parameetrite genereerimisel rakendatakse alati mingit variatiivsust ning seda ka kontekstide jaoks, mis ei varieeru väga palju. Sellisel juhul võib tulemuseks olla parameetrite ekstreemsed väärtused, mis avalduvad ülalpool kirjeldatud viisil väljundkõnes. [8]

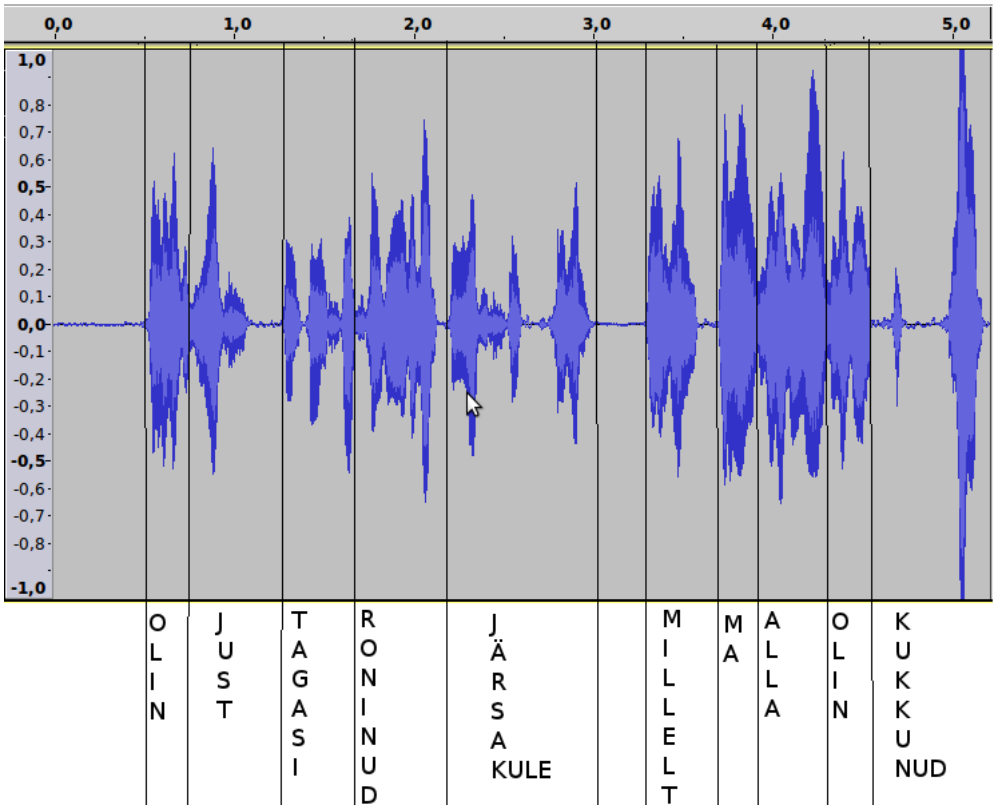
Joonisel 4.2 on kujutatud eespool toodud lause sünteesituna uuesti, kuid ilma globaalse variatiivsuseta. Helilaine on palju ühtlasem ning sünteesitud helifailis [34] ei esine moonutusi ning suuri amplituudi kõikumisi. Teisalt, kui võrrelda antud lauset eelmisega, siis on tajutav, et lause on ühtlasem ning iseloomult monotoonsem. See on selge tõend sellest, et globaalne variatiivsus lisab sünteeskõnele tubli annuse loomulikkust.



Joonis 4.2: HTS-iga sünteesitud (ilma globaalse variatiivsuseta) lause [34] helilaine

Tasub märkimist, et sama lause sünteesituna Festivali sünteesimootori abil kõlab

veidi teisiti kui HTS-iga saadud kõne (joonis 4.3). Ka seal leidub tehiseid ning moonutusi, kuid need asetsevad teistes kohtades kui eelmisel korral. Erinevused on tingitud sellest, et Festivali HTS-i moodul kasutab mudelite ning parameetrite tekitamise jaoks HTK tarkvarapaketis olevast erinevat algoritmi ning mootor ise ei ole seotud HTK-ga. See aga võimaldabki selle, et HTS-iga treenitud häälte tugi on vabavaralise sünteessüsteemi Festival osa.



Joonis 4.3: Festivaliga sünteesitud (globaalset variatiivsust kasutades) lause [35]helilaine

Hetkel ei toeta Festivalis olev HTS-i sünteesimootor globaalset variatiivsust välja lülitada, seega esineb käesoleva töö raames valminud sünteeskõnes kohati järske helitugevuse kõikumisi ning tehiseid. Nagu eelpool öeldud, ei ole need tingitud niivõrd treeningandmeteks kasutatud helisignaalide või märgendite kvaliteedist, vaid pigem teatud häälikukontekstide variatiivsusest ning esinemissagedusest. Suure tõenäosusega aitaks tehiste esinemist vähendada treeningandmete hulga suurendamine, et tõsta globaalse variatiivsuse rakendamise edukust.

4.2 Tulevik

Kuigi käesoleva töö raames sai realiseeritud piiramatu sõnavaraga eestikeelne kõnesünteesi moodul Festivali sünteesisüsteemile, kasutab selle kõige käepärasem lahendus siiski sisendiks saadud ortograafilise teksti töötlemiseks foneetiliseks tekstiks Eesti Keele Instituudis difoonsüntesaatori jaoks mõeldud programmi. Seega on tõsine puudus Festivaliga ühilduvast tekstitöötlusmoodulist eesti keele jaoks, seda enam, et Festivali kasutatakse tihtipeale teiste sünteesisüsteemide jaoks häälte ning moodulite koostamisel. Viimast punkti tasub eriti rõhutada arvestades nutitelefonide ning nende rakenduste arengut. iPhone'i Siri eestikeelse kolleegi valmimiseni on küll veel pikk maa käia, kuid heakvaliteedilisel ning kompaktsel kõnesüntesaatoril on selles süsteemis kandev osa. Oma suuruse ning arvutusvõimsuslike nõuete poolest sobib Markovi peitmudelitel põhinev kõnesüntees suurepäraselt erinevatesse mobiilsetesse seadmetesse.

Kuigi Markovi peitmudelitel põhineva kõnesünteesi puhul tõstetakse esile hästi treenitud hääle kvaliteeti ning loomulikkust, ei tähenda see, et selle valdkonnaga enam ei tegeldaks. Kaugel sellest. Sünteeskõne kvaliteedi tõstmisega tegeldakse näiteks väga tõsiselt meie põhjanaabrite pool Soomes, kus töögrupp Aalto Ülikoolist ning Helsinki Ülikoolist arendab GlottHMM [7] süsteemi, kus allikasignaali modelleeritakse inimese kõri ning kõnetrakti füsioloogilisi protsesse eeskujuks võttes. Senised katsetused selles vallas on näidanud häid tulemusi sünteeskõne kõla loomulikkuse tõstmisel.

Kokkuvõte

Kuigi kõnesünteesiga on tegeldud üle poole sajandi, ei näi uutele arengutele kuskilt lõppu paistmas. Veel kümme aastat tagasi tunti veel üsna vähe kõnesünteesi meetodit, mille põhiprintsiibid tõestasid end hästi kõnetuvastuses - Markovi peitmudelitel põhinevat kõnesünteesi.

Kümme aastat tagasi valmis eesti keele jaoks difoonidel baseeruv kõnesüntesaator. Selle süntesaatori tekstitöötlusmooduli ning Eesti Keele Instituutist saadud *eki_et_liisi* kõnekorpuse abil valmis käesoleva magistritöö tulemusena piiramatu sõnavaraga eestikeelne Markovi peitmudelitel põhinev sünteeshääl *eki_et_liisi_hts* vabavaralisele kõnesünteesi süsteemile Festival. Saadud sünteeshäälega võib rahule jääda - kõne on üllatavalt loomulik, kuigi kohati esineb tehiseid, mis rikuvad saadud sünteeshääle sidusust. Nende tehiste likvideerimine on probleem, mis jäi hetkel Festivali süsteemis lahendamata. Hoolimata sellest, on loodud eestikeelne sünteeshääl täiesti kasutatav ning arvutite ja sidevahendite viimase aja arenguid vaadates pole üldse võimatu, et millalgi lähitulevikus on paljudel inimestel taskus seade, mis kõneleb nendega antud lõputöö käigus loodud hääle või selle järeltulija abil.

Viited

[1] E.Eide, A. Aaron, R. Bakis, W. Hamza, M.Picheny, J.Pitrelli. A corpus-based approach to <AHEM/> expressive speech synthesis. - *5th ISCA Speech Synthesis Workshop - Pittsburgh, 2004.*

[2] HMM-based Speech Synthesis System (HTS). [www] http://hts.sp.nitech.ac.jp/ (11.05.2012)

[3] HTK Speech Recognition Toolkit. [www] http://htk.eng.cam.ac.uk/ (11.05.2012)

[4] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W. Black, Keiichi Tokuda. The HMM-based Speech Synthesis System (HTS) Version 2.0. - *Proceedings of ISCA SSW6, Bonn, Germany, Aug. 2007.*

[5] Paul Taylor (2009). Text-to-Speech Synthesis. Cambridge: Cambridge University Press.

[6] HMM-based Speech Synthesis System (HTS) - Voice Demos [www] http://hts.sp.nitech.ac.jp/?Voice%20Demos (31.05.2012)

[7] Tuomo Raitio, Antti Suni, Martti Vainio, Paavo Alko. Recent developments of statistical parametric speech synthesis system GlottHMM. - *XXVII Foneetikan päivat - Phonetics Symposium 2012 : 17th-18th February 2012, Tallinn, Estonia, 2012.*

[8] Simon King. A beginners' guide to statistical parametric speech synthesis. [www] http://www.cstr.ed.ac.uk/downloads/publications/2010/king_hmm_tutorial.pdf (26.05.2012)

[9] The Festival Speech Synthesis System. [www] http://www.cstr.ed.ac.uk/projects/festival/ (11.05.2012)

[10] The MARY Text-to-Speech System. [www] http://mary.dfki.de (11.05.2012)

[11] hts_engine API. [www] http://hts-engine.sourceforge.net (11.05.2012)

[12] Open JTalk. [www] http://open-jtalk.sourceforge.net (11.05.2012)

[13] Heiga Zen, Keiichiro Oura, Takashi Nose, Junichi Yamagishi, Shinji Sako, Tomoki Toda, Takashi Masuko, Alan W. Black, Keiichi Tokuda. Recent development of the HMM-based speech synthesis system (HTS). - *Proc. 2009 Asia-Pacific Signal and Information Processing Association (APSIPA), Sapporo, Japan, October 2009.*

[14] Keiichi Tokuda, Heiga Zen, Alan W. Black. An HMM-based speech synthesis system applied to English. - *Proc. of 2002 IEEE SSW, Sept. 2002*

[15] The MBROLA Project. [www] http://tcts.fpms.ac.be/synthesis/ (13.05.2012)

[16] Eesti keele tekst-kõne süntesaator [www] http://phon.ioc.ee/synt (12.05.2012)

[17] Eesti keele kõnesüntees. [www] http://www.eki.ee/keeletehnoloogia/projektid/syntees/tks.htr (29.05.2012)

[18] Einar Meister. Kõnetehnoloogia olemusest. - *A & A*, 2002, 5, 20 - 33.

[19] Meelis Mihkla, Einar Meister. Eesti keele tekst-kõne-süntees. *Keel ja Kirjandus*, 2002, 2, 88-97.

[20] Aleksandr Tkatšenko. Named Entity Recognition for the Estonian Language: magistritöö, Tartu Ülikool, Tartu, 2010.

[21] Fail *treening/eki_et_liisi_l_02404.utt* kaasasoleva CD peal

[22] Fail *treening/eki_et_liisi_l_02404_mono.lab* kaasasoleva CD peal

[23] Fail *treening/eki_et_liisi_l_02404_full.lab* kaasasoleva CD peal

[24] Fail *treening/eki_et_liisi_l_02404_gen.lab* kaasasoleva CD peal

[25] Fail *treening/lab_format.pdf* kaasasoleva CD peal

[26] HTS speaker dependent training demo. [www] http://hts.sp.nitech.ac.jp/archives/2.2/HTS-demo_CMU-ARCTIC-SLT.tar.bz2 (29.05.2012)

[27] Fail *festival/et/eki_et_liisi_hts/festvox/eki_et_liisi_phoneset.scm* kaasasoleva CD peal

[28] Fail *festival/et/eki_et_liisi_hts/festvox/eki_et_liisi_lexicon.scm* kaasasoleva CD peal

[29] Kataloog *festival/et/eki_et_liisi_hts/hts* kaasasoleva CD peal

[30] Kataloog *festival/et/eki_et_liisi_hts/festvox* kaasasoleva CD peal

[31] Kataloog *tekstitöötlus* kaasasoleva CD peal

[32] Fail *tekstitöötlus/pho2fest_et.py* kaasasoleva CD peal

[33] Fail *helinäited/olin_kukkunud_hts_gv.wav* kaasasoleva CD peal

[34] Fail *helinäited/olin_kukkunud_hts_no_gv.wav* kaasasoleva CD peal

[35] Fail *helinäited/olin_kukkunud_festival_pho.wav* kaasasoleva CD peal

[36] Meelis Mihkla (2007). Kõne ajalise struktuuri modelleerimine eestikeelsele tekst-kõne sünteesile: väitekiri, Tartu Ülikool, Tartu Ülikooli kirjastus.

[37] eSpeak: Speech Synthesizer. [www] http://espeak.sourceforge.net/ (31.05.2012)

[38] Meelis Mihkla, Einar Meister. Eesti keele tekst-kõne-süntees. *Keel ja Kirjandus*, 2002, 3, 173 - 182.

Lisad

Lisa 1: Eestikeelne naishääl Festival sünteesisüsteemi jaoks ning skriptid (CD plaadil)