NTNU
Det skapende universitet

Forbedret naturlighet i HMM-drevet talesyntese

Lene Mølmen

Master i kommunikasjonsteknologi
Oppgaven levert: Juni 2006
Hovedveileder: Torbjørn Svendsen, IET
Biveileder(e): Dyre Meen, IET

Norges teknisk-naturvitenskapelige universitet
Institutt for elektronikk og telekommunikasjon

Oppgavetekst

Skjulte Markov-modeller ("Hidden Markov Models", HMM) er en kraftig metode for modellering av stokastiske prosesser. Innen taleteknologi danner HMM grunnlaget for de fleste systemer for automatisk talegjenkjenning. HMM benyttes her til å skape en statistisk modell av taleproduksjonen. Talegjenkjennerens oppgave er å dekode det observerte talesignalet, dvs. å bestemme hvilken ordsekvens som med størst sannsynlighet har generert observasjonen.
Siden HMM i utgangspunktet er en modell for taleproduksjon, er teknikken også interessant for talesyntese. Modellen er stokastisk, og den vil derfor kunne ha bedre forutsetninger for å generere en uttale som inneholder mer av den variasjonen vi finner i naturlig tale enn det som er oppnåelig med regelstyrte syntetisatorer. HMM-basert syntese har i tillegg den fordelen at den setter små krav til minne, lager og prosessorkraft. For enkelte anvendelser er nettopp det at talesyntetisatoren har lite "fotavtrykk" av stor betyding og HMM er i denne sammenhengen en attraktiv teknikk.
Denne oppgaven går ut på å undersøke muligheter til å forbedre naturligheten til norsk, syntetisk tale basert på HMM. Det skal tas utgangspunkt i programvarebiblioteker for HMM-basert talegjenkjenning (HTK), generell tekst-til-talesyntese (Festival) og for HMM-basert talesyntese (HTS) og en basisversjon av norsk, HMM-basert talesyntese.

Oppgaven gitt: 16. januar 2006
Hovedveileder: Torbjørn Svendsen, IET

Forord

Våren 2006 ble det gitt en masteroppgave ved institutt for Teleteknikk ved Norges Teknisk-Naturvitenskapelige Universitet.

Masteroppgaven omhandler HMM-basert talesyntese, og det er blitt sett på hvordan en kan forbedre naturligheten til en basisversjon, norsk HMM-basert talesyntese. Oppgaven er en avsluttende oppgave ved Master i Kommunikasjonsteknologi.

Jeg ønsker å rette en stor takk til veileder Dyre Meen. I tillegg vil jeg takke faglærer Torbjørn Svensen for god veiledning gjennom oppgaven.

Trondheim, 19. juni 2006

_____

Lene Mølmen

Sammendrag

I denne masteroppgaven har en sett på hvordan en kan forbedre naturligheten i en norsk stemme basert på skjulte Markovmodeller. HMMmodeller har en bra egenskap til å modellere godt de variasjoner som finnes i et talesignal. Det er blitt tatt utgangspunkt i en basisversjon av norsk HMM-basert talesyntese hvor en kan syntetisere norsk tale fra trente HMMer.

Det er gitt en teoretisk beskrivelse av et system for HMM-basert talesyntese (HTS-system). I treningsdelen blir spektrum og eksitasjonsparametere trukket ut fra taledatabasen og modellert av kontekstavhengige HMMer. I syntesedelen skjøtes kontekstavhengige HMMer sammen i henhold til den teksten som skal syntetiseres. Taleparametrene inneholdt i HMMmodellene brukes til å styre en signalkilde og et MLSA-filter som syntetiserer tale i henhold til parametersekvensen.

Kvaliteten på den norske stemmen generert med basisversjonen av HTS-systemet, har en ”vokodet” klang. En årsak til denne klangen kan være at det brukes en svært enkel signalkilde, som enten genererer stemt lyd eller ustemt lyd når talesignalet genereres. Stemmen i basisversjonen hadde og en unaturlig setningsmelodi som det var ønskelig å forbedre.

Norsk er et tonespråk. Det vil si at tonen varieres slik at samme ord får ulik betydning alt ettersom hvilken ordtone (tonem) ordet uttales med. Et eksempel på ordpar med tonemkontraster er: bade - badet. Denne karakteristiske egenskapen er implementert i systemet i den hensikt å forbedre naturligheten i talen.

To blandede eksitasjonsmodeller er blitt studert, Harmonic plus Noise Model (HNM) og STRAIGHT, i den hensikt å redusere den ”vokodete” klangen på talen. STRAIGHT er modellen som er implementert i denne masteroppgaven. STRAIGHT ekstraherer kontinuerlige og jevne fundamentalfrekvenskurver fra taledatabasen. Systemet bruker en pitsj-adaptiv metode i spektralanalysen og oppnår et glattet spektrogram uten spor av signalperiodisitet. Disse metodene gjør at STRAIGHT kan resyntetisere svært naturlig og forståelig tale.

To norske HTS-stemmer med forbedret naturlighet er blitt konfigurert i det generelle tekst-til-tale systemet Festival. Festival gjør det mulig å syntetisere en hvilken som helst norsk setning. Disse setningene er syntetisert med den gamle signalkilden, men det er blitt lagt til tonelag i begge stemmene, og den ene er trent med f0-kurver ekstrahert fra STRAIGHT. Arbeidet med å lage en HTS-stemme basert på taleparametere fra STRAIGHT-modellen, førte ikke frem da tiden ikke strakk til. Stemmen er blitt trent av HTS-systemet og det er blitt generert parametersekvenser STRAIGHT kan lese inn for syntese. Det som gjenstår er selve syntetiseringen.

Evaluering av stemmene med hensyn på naturlighet, viser at HTS-stemmen trent med f_0-kurver fra STRAIGHT og tonelag, er den stemmen som oppnådde størst naturlighet.

HTS-systemet er et svært fleksibelt system som har lite ”fotavtrykk” og er attraktiv i anvendelser av små enheter som har begrenset med lagringsplass og beregningskraft, som f.eks mobiltelefoner og PDAer.

Innhold

Figurer

Tabeller

Forkortelser

ap	Aperiodiske parametere
C/N	Carrier-to-noise
f_0	grunnfrekvensen til vibrasjonen fra stemmebåndet, tonen
HMM	Hidden Markov Modell
HNM	Harmonic plus Noise Model
HTK	Hidden Markov Model Toolkit
HTS	HMM-basert talesyntesesystem
LPC	Linear Predictive Coding
mcep	Mel cepstrum coefficients (cepstrum er et anagram av det engelske ordet spectrum)
MELP	Mixed Exitation Linear Predictive Vocoder Algorithm
MDL	Minimum Description Lenght
MFCC	Mel Frequency Cepstrum Coefficient
ML	Maximum-Likelihood
MLSA	Mel Log Spectrum Approximation
PSOLA	Pitch Synchronous Overlap Add
SPTK	Speech Signal Processing Toolkit
STFT	Korttids Fouriertransform
STRAIGHT	Speech Transformation and Representation using Adaptive Interpolation of weighted spectrum
TTS	Text-to-Speech

1 Innledning

Menneskelig tale er kompleks av natur ved at den kan varieres avhengig av talestilen og følelsene til den som snakker. Dette er vanskelig å imitere ved kunstig tale. Det finnes i dag flere tekst-til-talesystemer (TTS), som kan generere svært forståelig og naturlig tale. Datadrevet skjøtesyntese er en utbredt metode som gir svært høy kvalitet på den syntetiske talen. Synteseteknikken baseres på antagelsen om at en kan generere vilkårlig tale ved å skjøte sammen basisenheter av en lagret talemengde. Denne teknikken er lite fleksibel ved at det er vanskelig å variere stemmekarakteristikk som personlighet, talestil og følelser, til dette trengs det store mengder med data fra ulike talere og med ulik talestil. Innsamling av slike data er en tidkrevende og kostbar prosess.

En annen metode, HMM-basert talesyntese system (HTS), er blitt utviklet og implementert av HTS-working group, en gruppe tilknyttet The Nagoya Institute of Technology i Japan [1]. Teknikken bruker parametrene i skjulte Markovmodeller (HMM) til å generere tale. Modellene er godt kjent fra talegjenkjenning. De er svært fleksible og det er mulig å endre egenskapene til den syntetiske stemmen ved å omforme HMM parametrene på en passende måte. Kontekstavhengige HMMer (trifoner) brukes for å få tak i de kontekstuelle faktorene som påvirker spektrum og som definerer de ulike lingvistiske og prosodiske faktorene i et språk. Disse kontekstuelle faktorene gjør at systemet er enkelt å utvide til nye språk [2]. En implementering av et HMM-basert system for talesyntese (HTS) har en liten syntesemotor, kun i størrelsesorden 1 MB inkludert akustiske modeller. Disse akustiske modellene inneholder all nødvendig informasjon for å syntetisere tale. Teknikken er derfor svært aktuell i små enheter som mobiltelefoner eller PDAer. Systemet kan enkelt endre talerkarakteristikk ved bruk av adapsjonsteknikker [3] og interpoleringsteknikker [4] utviklet for talegjenkjenning. Med talerkarakteristikk menes egenskaper som kjønn, alder, sinnsstemning og tonefall.

I tidligere arbeid [5], har en kunnet syntetisere norsk tale fra trente HMMer med til dels god kvalitet og naturlighet. Likevel har den syntetiske stemmen hatt en litt robotaktig klang. En av grunnene til dette kan være at i basisversjonen av systemet, brukes en tradisjonell eksitasjonsmodell som enten generer et periodisk pulstog eller hvit støy. I denne masteroppgaven har en sett på hvilke forbedringer som oppnås ved å erstatte eksitasjonsmodellen med en mer nøyaktig modell. To modeller er blitt studert. En høykvalitets vokoder kalt STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weighted spectrum), utviklet av Kawahara et al.[6], og Harmonic plus Noise Model (HNM) utviklet av Stylianou [7].

STRAIGHT systemet bruker f_0-adaptive metoder. Spektralanalysen gjøres i kombinasjon med en overflaterekonstruksjonsmetode i tid -og frekvensdomenet for å fjerne signalperiodisitet og det brukes en eksitasjonskilde basert på fasemanipulasjon. Dessuten kan STRAIGHT manipulere taleparametere som pitsj, ansatsrørets lengde og taleraten uten at den høye kvaliteten på reproduksjonen av tale synker.

I den andre modellen, HNM, er talesignalet representert med harmoniske av f_0 samt en frekvens for maksimum stemte frekvens. Over denne frekvensen representeres talen ved støy. Under har vi de harmoniske komponentene. Det er bare de harmoniske komponentene som modifiseres, for ustemt støy vil det opprinnelige talesignalet kopieres. Syntetisk tale med disse talesyntesene høres jevn ut og kvaliteten er konsekvent.

I masteroppgaven er det valgt å implementere STRAIGHT til det norske HTS-systemet. Dette fordi verktøyene for STRAIGHT-systemet lå til rette og det er vist i [8] at en implementasjon av STRAIGHT i det japanske HTS-systemet har gitt gode resultater for å bedre kvaliteten og naturligheten på talen.

Setningsmelodi eller intonasjon er et fenomen som spiller en viktig rolle for hvordan mennesket oppfatter hvor naturlig en syntetisk stemme er. En karakteristisk egenskap ved intonasjonen i mange norske dialekter er forskjellen mellom to ulike tonelag eller tonemer. Det finnes to tonelag, dvs. ordmelodier, i norsk, som vi vanligvis kaller *tonelag 1* og *tonelag 2*. Det som karakteriserer det vi kan kalle et *tonelagspar*, er at språklydene i de to ordene er de samme, slik at betydningsforskjellen helt og holdent er knyttet til det at de har ulike melodier [9]. Tonelaget er og en viktig dialektmarkør. Det er i denne masteroppgaven tatt med denne karakteristikken for å bedre naturligheten i den norske HTS-stemmen.

Det vil i denne rapporten først bli gitt en oversikt over basisversjonen av et HMM-basert talesyntesesystem, før nyere utvikling av systemet, STRAIGHT-basert vokoding blir beskrevet.

Det antas at leseren har kjennskap til fundamentale teknikker innenfor taleteknologi.

Rapporten er delt inn som følger:

Kapittel 2 introduserer begrepet talesyntese og de viktigste syntese teknikkene er presentert.

Kapittel 3 inneholder en teoretisk beskrivelse av basisversjonen av HMM-basert talesyntese.

Kapittel 4 presenterer ulike eksitasjonsmodeller før STRAIGHT-systemet brukt i denne masteroppgaven blir beskrevet.

Kapittel 5 beskriver arbeidet med å implementere STRAIGHT-systemet til det norske HTS-systemet, samt evaluering av stemmene som ble utviklet.

Kapittel 6 presenterer konklusjoner trukket fra oppgaven og videre arbeid.

2 Introduksjon til talesyntese

Tale er det hjelpemiddelet som brukes av de fleste mennesker for å kommunisere. Videre kan tale formidle annen type informasjon slik som følelser, holdninger og talerindividualitet. En kan derfor si at tale er et naturlig og nyttig hjelpemiddel for mennesker når en skal kommunisere. I løpet av de siste årene har datamaskin-teknologien utviklet seg enormt, og datamaskiner har blitt noe de fleste mennesker benytter i hverdagen. Taleteknologi kan være et viktig bidrag for å forenkle menneske-maskin interaksjon, og på denne måten gjøre datamaskiner ytterlige allment tilgjengelig. Talesyntese og talegjenkjenning har de siste årene blitt mer vanlig som grensesnitt, og i de kommende årene er det forventet at bruken av denne typen grensesnitt vil øke. Talegjenkjenning konverterer tale til skrevet tekst. Talesyntese er en teknikk som gjør den motsatte prosessen av talegjenkjenning, det vil si den gir ut informasjon i form av kunstig tale. I slike systemer er en og interessert i andre typer for informasjon, slik som talerindividualitet og følelser som er nødvendig for å kunne realisere god kommunikasjon.

Dette kapittelet forteller kort om historien til talesyntese og beskriver hva som menes med et tekst-til-tale system (TTS), før de viktigste talesyntesemetodene blir beskrevet.

2.1 Historikk

Idéen om at en maskin kan generere tale har eksistert en god stund, men selve realiseringen av en slik maskin har bare vært praktisk mulig de siste 50 årene. Det er først inntil nylig, det vil si de siste 20 årene, at vi har kunnet se praktiske eksempler av tekst-til-talesystemer som kan si en gitt tekst. Noen med til dels bra kvalitet, det vil si stemmer som er forståelige, men fortsatt er lydkvaliteten og naturligheten i talen et gjenstående problem [10].

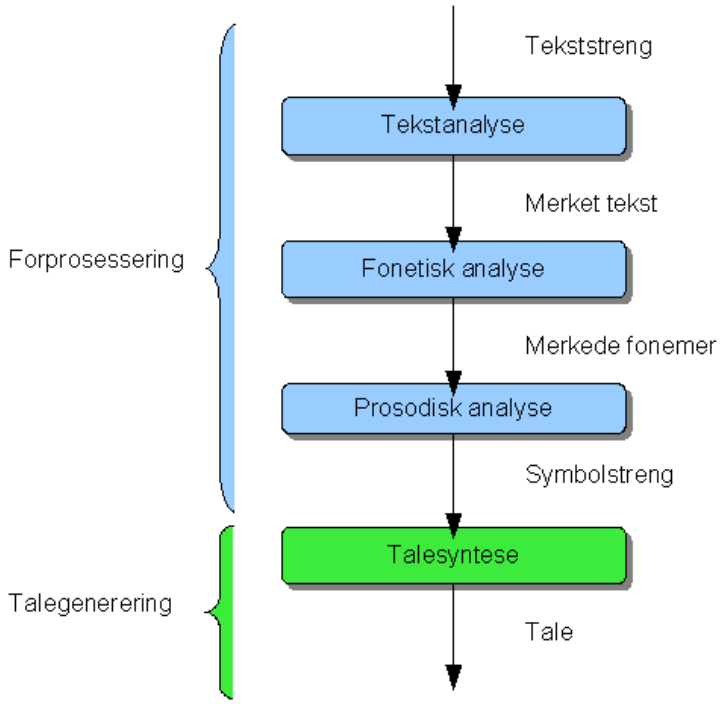
Allerede i 1939 ble det på Bell Laboratories utviklet og demonstrert et mekanisk system som kunne si hele ytringer. Systemet, kalt Voder [11], ble styrt ved at en person manøvrerte ulike tangenter og pedaler for å styre et kunstig taleorgan. Senere kom elektroniske systemer som lagde ulike talelyder ved å la et talesignal fra et kunstig stemmebånd passere gjennom ett antall filtre som etterlignet den menneskelige strupen.

Ettersom datamaskinene de siste årene er blitt raskere, billigere og har tilgang på større lagringsplass, skjer det meste av utviklingen av talesyntese i dag på datamaskiner. Maskinen bruker ulike avanserte teknikker og modeller for å etterligne måten mennesker snakker på. Disse systemene kan forandre en vilkårlig tekst til en sekvens med parameterverdier som beskriver talelyden og prosodien. Denne sekvensen blir tilslutt sendt gjennom et kunstig taleorgan, en såkalt syntetisator.

2.2 Systemoversikt

Tekst-til-talesyntese (TTS), er en teknikk for å lage talesignal fra en vilkårlig gitt tekst i den hensikt å overføre informasjon fra en maskin til et menneske ved kunstig tale. For å kunne overføre den fullstendige informasjonen i et talesignal må et TTS system ha muligheten til å generere naturlig tale med forskjellige stemmekarakteristikker og forskjellig talerstil [12]. Eksempler på stemmekarakteristikker er alder, kjønn, grov/myk stemme etc. Eksempler på talerstiler er glad, sint, deprimert, nøytral.

Et TTS system består av to hoveddeler. Den første delen er forprosessoren, hvor teksten som kommer inn analyseres og det produseres en streng med lydsymboler (typisk fonemer eller difoner) med tilhørende merking. Merkingen spesifiserer hvordan lydenheten skal realiseres med hensyn på grunntone, trykk og varighet, dette kalles prosodi. I den andre delen, talesyntesedelen, blir symbolstrengen omformet til syntetisk tale, hvor den akustiske lyden blir kontrollert av den fonetiske og prosodiske informasjonen [13]. Et forenklet blokkdiagram av tekst-til-talesystem med tekst som inndata er vist i figur 1.



Figur 1: Oversikt over et tekst-til-talesystem [14].

I denne figuren tilsvarer de tre første blokkene forprosessoren. Den siste blokken, talesynteseblokken, er selve talegenereringen.
Kvaliteten til et TTS system er avhengig av både den lingvistiske prosesseringen og selve syntetiseringen av talen. Grad av kvalitet på den syntetiske talen kan måles i oppfattet naturlighet og forståelighet. Den perfekte talesyntetisator vil tilfredsstille Turing-testen, da vil en person ikke kunne angi om det er et menneske eller en maskin det kommuniserer med [14]. De ulike modulene i et TTS system blir forklart nærmere i påfølgende underkapitler.

2.2.1 Tekstanalyse

I tekstanalysen spiller den kontekstavhengige informasjonen en viktig rolle for kvaliteten og forståligheten til den syntetiske talen. Den grunnleggende lingvistiske prosesseringen forsikrer at tall og forkortelser blir riktig tolket og uttalt, og er dermed en viktig faktor for den totale kvaliteten på talen.

Tekstanalysen inneholder tre prosesser [14]:

- **Dokumentstruktur**

Skrevet tekst kan bygges opp på ulike måter, for eksempel i flere kolonner og sider som en vanlig avis artikkel. Dette kan skape problemer for maskinen som leser inn teksten. Avsnitt og setningsoppdeling er viktig å påvise da det kan ha direkte konsekvenser for prosodien.

- **Tekst normalisering**
  I dette trinnet konverteres symboler, tall, forkortelser og andre ikke-ortografiske enheter av tekst til vanlig ortografisk transkripsjon. For eksempel vil tallet *"234"* konverters til *"to hundre og tretti fire."* Forkortelser må skrives ut til fullstendige ord, uttalt slik det skrives eller uttalt ord-for-ord. Spesielt for forkortelser oppstår det kontekstuelle problemer, f.eks St. kan bety helgen eller gate på engelsk (street). Noen ganger vil den naboliggende informasjonene være til hjelp for å finne rett betydning.

- **Lingvistisk analyse**
  Den lingvistiske analysen gjenvinner syntaktisk gruppe og de semantiske egenskapene til ordene, frasene og setningene. Lingvistisk analyse er viktig med tanke på uttale og prosodi i senere prosesser.

2.2.2 Fonetisk analyse

Neste trinn i systemet er å konvertere ordene til en fonemsekvens ved hjelp av leksikon, grafem-til-fonem regler og morfologisk analyse [14]. Korrekt uttale for forskjellige kontekster i en tekst. Noen ord, såkalte *homografer*, gjør ting vanskelig for TTS systemer. Homografer er ord som staves likt men har forskjellig betydning og veldig ofte uttales forskjellig. Et eksempel er *man* (pronomen) og *man* (på hest). Uttalen på forskjellige ord kan og være forskjellig sett i forhold til hvilken kontekst ordet befinner seg i. På norsk har vi ofte tonelagsforskjell mellom verb og substantiv. Ordet *"hoppet"* er et eksempel på det. Morfologisk analyse løser dette problemet, som omhandler ordbøying og orddannelse. Tilslutt gjøres en bokstav-til-lyd konvertering og oppslag i en ordbok for å finne riktig uttale for vilkårlige ord.

2.2.3 Prosodisk analyse

Å finne riktig intonasjon, trykk og varighet fra skrevet tekst er en utfordrende oppgave. Disse egenskapene kalles prosodiske egenskaper. Prosodi er et komplekst nett av fysiske og fonetiske effekter som bevisst eller ubevisst anvendes for å uttrykke blant annet holdninger, antagelser og type humør gjennom tale [14]. Syntaks og semantikk spiller en viktig rolle i prediksjonen av prosodi. Pitsjkonturen er avhengig av meningen i ytringen. Prosodi dekker alle de egenskapene til uttale som ikke direkte går på definisjonen av vokaler og konsonanter. Disse egenskapene kan være pauser, forandringer i tonefall, volum, stemmekvalitet og talehastighet. Momenter som kjønn og aldre er også en del av prosodien. I talesyntese skjer vanligvis genereringen av prosodi i to steg:

Lingvistisk nivå: På det lingvistiske nivået utledes en abstrakt beskrivelse av setningsprosodien. Denne beskrivelsen består av nivå for ordfremheving og frasestruktur.
Akustisk nivå: På det akustiske nivået beregnes de fysiske parametrene for den akustiske realiseringen ut fra informasjonen utledet på det lingvistiske nivået.

I noen TTS systemer, spesielt engelske TTS systemer, blir ToBI (Tone and Break Indices) [15] merkelapper predikert. ToBI er en standard for merking av pitsjaksent (som

angir prominensen til en stavelse) og tonalitet (som angir grenser mellom prosodiske fraser). Modellen representerer intonasjonen som en sekvens med H (high) og L (lav) toner, og kombinasjoner av disse. På norsk har vi en egen modell for intonasjonsmodellering, *Trondheimsmodellen* [9]. I Trondheimsmodellen er stavelsen den minste prosodiske enheten. Stavelse er byggestein for tonale og rytmiske enheter og "bærer" av fenomener som tonelag, trykk og fraseaksent i det norske språket.

2.2.4 Talesyntese, generering av talebølgeformen

Talesyntese kan produseres av flere ulike metoder. Disse metodene kan bli delt inn i tre grupper [16]:

- Artikulatorisk syntese, har til hensikt å modellere det menneskelige taleproduksjonssystemet.
- Formantsyntese, modellerer overføringsfunksjonen til vokaltrakten basert på en kilde-filter modell.
- Skjøtesyntese, setter sammen segmenter av innspilte data fra naturlig tale.

Det vil nedenfor bli gitt en mer utfyllende beskrivelse av hver metode. I tillegg vil HMM-basert talesyntese bli introdusert.

2.2.5 Artikulatorisk syntese

Artikulatorisk syntese er en syntesemetode som skaper en fullstendig modell av taleren. Strupehodet, strupe, tunge, tenner og lepper blir modellert. Forskjellige lyder blir produsert ved å simulere hvordan en luftstrøm beveger seg og får musklene i talerøret til å trekke seg sammen slik at artikulatorene flytter på seg og endrer formen til talerøret. Metoden er svært kompleks og lite beregningsmessig effektiv til bruk i kommersielle systemer [17].
Denne metoden kan produsere naturlig tale, men den krever for mye beregnings kraft til å bli implementert i praktiske systemer [11]. Artikulatorisk syntese er den eneste av de syntesemetodene som blir presentert i oppgaven som modellerer taleapparatet.

2.2.6 Formantsyntese

Formantsyntese er basert på en kilde-filtermodell for taleproduksjon, hvor et sett av parametere (formantfrekvenser, båndbredder, amplitude, eksitasjonstype, grunntone) styrer lydgenereringen samt regler for overgang mellom lydene. Den menneskelige stemmen har ofte tydelige resonansfrekvenser, såkalte formanter, som avbildes når lyden går fra strupehodet, gjennom strupen, svelget og munnhulen. Dette modellerer man ved å la lyden fra et kunstig strupehode styres gjennom et antall filtre, og dermed skape formanter som er nær oppmot formantene til et snakkende menneske. Parametersekvenser utledet fra regler gir et signalspekter med finstruktur og formantstruktur som endrer seg i samsvar med ønsket fonemsekvens og prosodi. Tale fra en formantsyntetisator får en robotaktig, men klar stemme. Det er to typer formantsyntese med ulike egenskaper:

- Parallell formantsyntese: er bra for å modellere nasaler, frikativer og stopp-konsonanter.
- Serie formantsyntese: er bra for ikke-nasale stemte lyder.

Det er vanlig at disse to typene brukes i kombinasjon for å få best mulig ytelse [16].

2.2.7 Skjøtesyntese

Utgangspunket for skjøtesyntese er en database med innspilt tale, som er delt opp i segmenter og er fonetisk og prosodisk merket. I korte trekk handler skjøtesyntese om at segmenter (dvs. fonemer, ord eller liknende) fra denne databasen blir satt sammen til ytringer. Dette er en metode som er enkel og som ofte gir en naturlig tale. Forutsetningen er at segmentene passer godt sammen og at skjøtingen er god.
Det finnes flere ulemper med skjøtesyntese. Systemet er avhengig av å ha et komplett forråd av segmenter, noe som kan være plasskrevende og praktisk vanskelig å oppnå. En annen ulempe er at dersom ett segment mangler må det erstattes med et annet segment som likner, eller med stillhet. Dette vil gi diskontinuitet i ytring og kvaliteten blir således redusert. Videre blir talesegmenter i høy grad påvirket av koartikulasjon (dvs. endring i fonemartikulasjon og akustisk realisering pga. innflytelse fra andre lyder i samme utsagn). Når to formanter ikke passer sammen i skjøtepunktet, oppstår spektrale diskontinuiteter. Prosodiske diskontinuiteter inntreffer når pitsjen i skjøtepunktene ikke passer sammen [14]. Syntetisering av en vilkårlig tekst er fortsatt en stor utfordring for slike systemer [18]. Systemet er også vanligvis begrenset til en taler og en stemme; det er vanskelig å endre karakteristika ved stemmen.

Skjøtesyntese kan deles inn i tre kategorier; datadrevet skjøtesyntese, difonsyntese og domene syntese. Difonsyntese og domenesyntese kan sees på som spesialtilfeller av datadrevet skjøtesyntese.

2.2.7.1 Datadrevet skjøtesyntese (unit selection synthesis)

Denne syntesemetoden bruker store taledatabaser som inneholder et variert utvalg av enheter. Enhetene kan være blant annet foner, difoner, setninger, morfem, ord og stavelser. Enhetene i databasen organiseres. Dette kan gjøres på flere måter; en måte kan være å indeksere segmentene mht. for eksempel pitsj, tilstøtende fonem, faktisk fonem, posisjon i stavelse eller lignende. Ved syntetiseringen bestemmes hvilke enheter som skal brukes fra databasen utifra et gitt sett med kostnadsfunksjoner.

Datadrevet skjøtesyntese gir ofte en naturlig tale, da det er lite behov for signalbehandling av bølgeformen. Signalbehandling av bølgeformen er en stor kilde til forringelse av kvalitet på tale [13] [17]. Enkelte systemer bruker dog signalbehandling for å glatte ut bølgeformen ved skjøtingen, en av flere teknikker er Pitch Synchronous Overlap Add PSOLA (PSOLA) [16]. Teknikken ble utviklet av France Telecom (CNET), og er ikke en selvstendig talesyntesemetode, men en metode som glatter ut fundamentalfrekvensen i skjøtepunktene når to talesegmenter skjøtes sammen. PSOLA kan og gjøre prosodiske modifikasjoner som for eksempel endre amplitude, varighet eller pitsj til et talesegment.

2.2.7.2 Difonsyntese

Difonsyntese klarer seg med en liten taledatabase, da denne metoden bare bruker en eller få representasjoner av hvert difon som forkommer i et gitt språk. Antall slike enheter er begrenset, det er cirka 1500 til 2000 i et språk [16]. Et difon består av siste halvdel av et fonem etterfulgt av første halvdel av etterfølgende fonem. Dette tar vare på koartikulasjonseffekten mellom fonemene. Midtpunktet av et fonem er relativt stasjonært, og det egner seg derfor til skjøting. Hvert difon blir lagret som en bølgeformsrepresentasjon av en virkelig, uttalt realisering av difonet. I selve syntesen blir så difonene skjøtet sammen, etter at de er blitt modifisert til å ha en grunntone og varighet som samsvarer med tekstanalysen gjort i forprosessoren [13].

2.2.7.3 Domenesyntese

Domenesyntese er i praksis en liten utgave av datadrevet skjøtesyntese som beskrevet over, bare spesialisert innenfor ett felt [18]. Et eksempel vil være ”frøken ur”. Databasen behøver bare å inneholde setningen ”klokken er” og tallene 1 til 60. Domenesyntese gir mulighet for svært naturtro syntetisering.

2.3 HMM-basert talesyntese

En relativ ny metode for syntetisering av tale er blitt utviklet av et forskningsteam ved Institute of Technology i Japan [19]. Her brukes en modell godt kjent fra talegjenkjenning, til å modellere tale. Modellene som brukes er skjulte Markovmodeller (HMM). I dette systemet, som kalles HMM-basert talesyntese system (HTS), brukes de samme HMMene som brukes i talegjenkjenning til å generere tale [20]. Taleparametrene blir generert på en slik måte at den mest sannsynlige observasjonssekvensen blir maksimert. Dette systemet er blitt videreutviklet i løpet av de siste årene og er tilpasset bl.a. japansk og engelsk språk. I den siste versjonen av gratisprogrammet Festival (1.95beta) er det lagt inn støtte for HMM-basert talesyntese [21].

Det er dette systemet vi skal se nærmere på i de neste kapitelene.

3 Et HMM-basert talesyntese system

Som allerede nevnt i innledningen er et HMM-basert talesyntese system (HTS) blitt utviklet og implementert av et forskningsteam ved The Nagoya Institute of Technology i Japan [19]. Denne teknikken er allerede blitt tilpasset til flere språk, blant annet til engelsk, portugisisk og svensk [22].

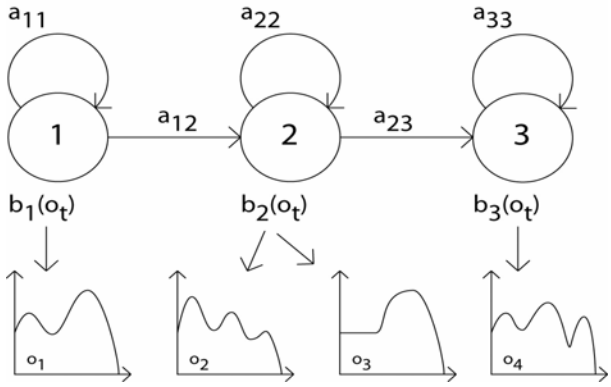
I HMM-basert talesyntese genereres fundamentalfrekvens, spektrum og fonemvarighet direkte fra trente HMMer. Dette gjøres fra desisjonstrær basert på en kontekstklyngings teknikk [23] [24]. F_0 modelleres av en utvidet HMMmodell kalt Multi-Space probability distributions HMM (MSD-HMM) [25]. Varigheten modelleres med en enkel gaussisk tetthetsfordeling hvor hver dimensjon viser varigheten til hver tilstand i HMMen. Mel-kepstrum parametrene modelleres av enten multi-dimensjonale gaussiske fordelings HMMer eller multi-dimensjonale gaussiske mixture fordelings HMMer. For hver egenskap konstrueres et desisjonstre. Desisjonstre for f_0 og for Mel-kepstrum konstrueres i hver tilstand i HMMmodellen. For varighet konstrueres bare et desisjonstre. Selve treningsprosedyren er automatisk. I syntesen blir den glattede parameterkurven, bestående av de statiske parametrene, generert fra HMMene ved å maksimere likelihood-kriteriet med hensyn på de dynamiske egenskapene til talesignalet [24].

Dette kapittelet introduserer de teorier og modeller som til sammen danner grunnlaget for et HMM-basert talesyntese system. Basisversjonen av HMM-basert talesyntesesystem vil bli beskrevet her, mens det utvidede systemet vil bli beskrevet i kapittel 5.

3.1 Definisjon av skjulte Markovmodeller

Skjulte Markovmodeller har lenge vært i bruk innenfor talegjenkjenning på grunn av deres fleksibilitet og evne til å modellere godt de variasjoner som er i et talesignal [26].

En skjult Markovmodell (HMM) er en stokastisk prosess som genererer en sekvens med diskrete tidsobservasjoner fra en rekke med tilstander som er koblet sammen [26]. HMMene oppdaterer hver tilstand ved et visst tidsintervall. Tilstandene endres i forhold til transisjonssannsynlighetene til hver tilstand, og genererer så en observasjons vektor o_t ved tiden t i forhold til sannsynlighetsfordelingen til den aktuelle tilstanden.
Et eksempel på en skjult Markovmodell som genererer et sett med observasjonsvektorer er vist i figur 2.



Figur 2: Et eksempel på en tre-tilstands venstre-mot-høyre HMM med sannsynlighetsfordelinger over de ulike utfallene.

Transisjonssannsynligheten a_{ij} er den betingede sannsynligheten for at en modell i tilstand i vil endre seg til tilstand j. En antar at transisjonssannsynligheten er konstant og ikke endrer seg over tid. Sannsynlighetsfordelingen $b_j(o_t)$ beskriver fordelingen av observasjonene som genereres fra tilstand j. Den gir sannsynligheten for at tilstand j har generert observasjonen o_t. Observasjonssekvensene er synlige, men det er ikke mulig å si hvilken tilstand som har generert hvilken observasjonssekvens. Vi sier at tilstandssekvensen er skjult, derav navnet skjulte Markovmodeller (Hidden Markov models).

Det er vanlig å bruke en venstre-til-høyre HMM for å modellere sekvenser med taleparametere, dette fordi talesignalets egenskaper endrer seg etter hverandre på denne måten. Modellen brukes ofte til å modellere en kort språklyd, for eksempel et fonem (den minste meningsskillende enheten i et språk). Hvis en da har tre tilstander som vist i figuren over, vil den første tilstanden modellere begynnelsen av fonemet, den midterste tilstanden vil modellere midten og den siste tilstanden vil modellere slutten av fonemet.

I HTS brukes skjulte Markovmodeller som består syv tilstander. De fem midterste tilstandene genererer en observasjonsvektor, mens den første tilstanden er en start-tilstand og den siste en stopp-tilstand.

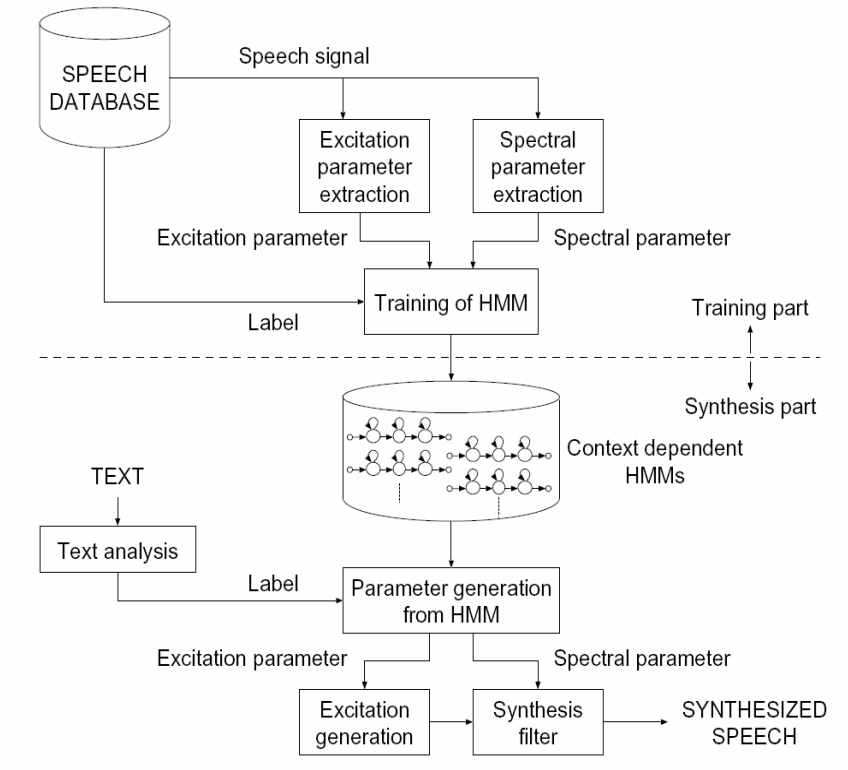
HMMer kan deles inn i forskjellige typer HMMer: diskrete HMMer og kontinuerlige HMMer, som kan modellere henholdsvis sekvenser med diskrete symboler og kontinuerlige vektorer. Observasjonssekvensen til den siste HMMen realiseres i praksis som en gaussisk multimiks fordeling.

3.2 Skjulte Markovmodeller til å generere tale.

I talegjenkjenning brukes HMMer til å modellere spektret av talesignalet. Spekteret inneholder informasjon om vokaler og konsonanter. Oppgaven til en talegjenkjenner er å finne den modellen som best beskriver den observerte ytringen. HMMene brukes til å generere en observasjon, dette kan utnyttes og brukes i talesyntese. Den produserte egenskapsvektorsekvensen som tilsvarer den observerte ytringen, kan brukes til å produsere tale fra en gitt enhetssekvens (fonemsekvens). Det er da viktig å tenke på at egenskapsvektoren er tilpasset taleproduksjon. Siden HMMmodellene enten er kontinuerlige eller diskrete, må det gjøres noen endringer av modellen. Nå ønsker en ikke bare å få tak i informasjon som tilsvarer spektrumet, men en vil og ha informasjon om egenskapene til kilden, som grunntonen f_0 og om kilden er stemt eller ustemt. I tillegg er det ønskelig å styre lengden på talelyden. Disse parametrene er med på å bestemme prosodien i en uttale og deres egenskaper kan være pauser, forandringer i tonefall, volum, stemmekvalitet og talehastighet.

3.3 HTS- systemoversikt

HTS-systemet bruker parametrene inneholdt i HMMmodeller til å generere tale. Systemet består av to deler: en treningsdel og en syntesedel. Figur 3 viser en oversikt over systemet.



Figur 3: Oversikt over HTS systemet [2].

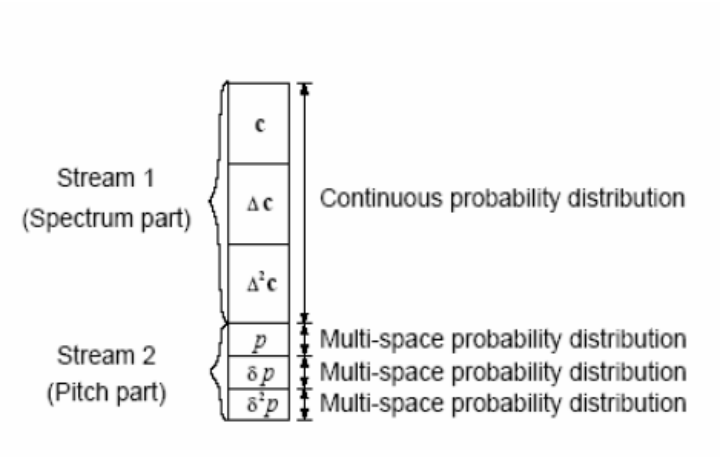
I de neste underkapitlene beskrives hva som skjer i de to delene av systemet.

3.3.1 Treningsdelen

Formålet med trening er å generere en database av akustiske HMMmodeller. Disse modellene vil bli brukt i syntesen for å generere tale.

3.3.1.1 Egenskapsvektor

Egenskapsvektoren til HMMene består av to deler, en spektrumsdel og en eksitasjonsdel, som vist i figur 4. Spektrumsdelen inneholder vektorer av Mel-kepstrum koeffisienter sammen med dens null-koeffisient og deres Δ -og ΔΔ-koeffisienter. Eksitasjonsdelen består av logaritmen til fundamentalfrekvensen f_0 og dens Δ -og ΔΔ-koeffisienter. Deltakoeffisientene inneholder informasjon om hastighet og akselerasjon hos parametrene og er med på å utjevne parameterkurvene slik at en oppnår en mer naturlig, kunstig tale.



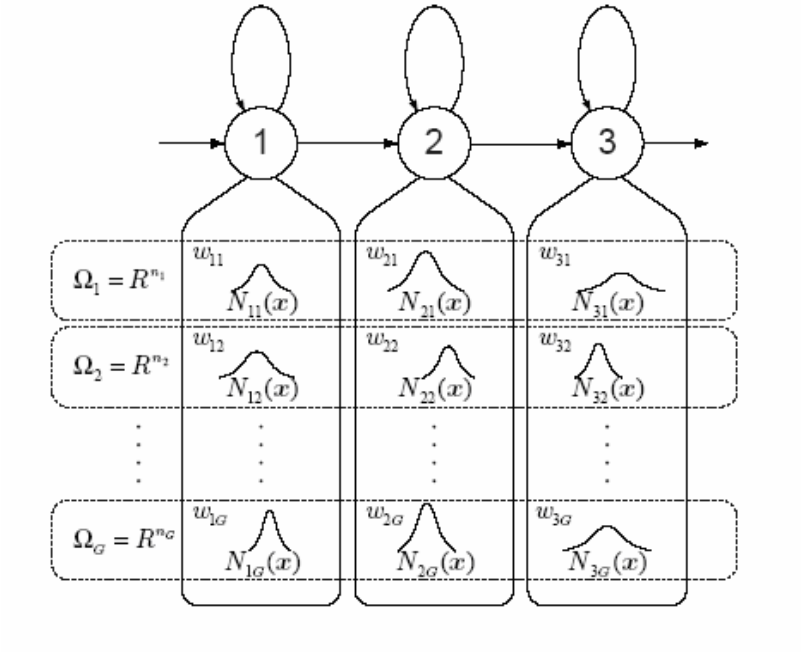
Figur 4: Egenskapsvektor, hvor strøm 1 er spektrumsdelen og strøm 2 er eksitasjonsdelen [23].

3.3.1.2 Modellering av spektrum

Modelleringen av spektrum skjer på samme måte som i talegjenkjenning, bortsett fra at der brukes ofte Mel-frekvens kepstralkoeffisienter (Mel frequency cepstral coefficients - MFCC), mens i HTS brukes Mel-kepstrum (mcep) som parametere for spektrum. Sekvenser av disse parametervektorene hentes fra taledatabasen ved hjelp av en Mel-kepstral analyseteknikk [27] som generer et perseptuelt vektet sett av parametere for hver ramme. En fordel med mcep-koeffisientene sammenlignet med MFCC er at en kan syntetisere tale fra parametrene ved bruk av et Mel Log Spectral Approximation (MLSA) filter [27].

3.3.1.3 Modellering av fundamentalfrekvens (f_0)

For å modellere f_0 kan en ikke bruke de vanlige diskrete eller kontinuerlige HMMene, dette fordi verdier for f_0 ikke er definert i de ustemte delene av talesignalet. Observasjonssekvensen til f_0 består av endimensjonale kontinuerlige verdier og diskrete verdier som representerer de ustemte områdene i talesignalet. Løsningen på problemet er gitt i [25]. Her har man utvidet HMMen slik at modellen er i stand til å modellere en sekvens med observasjonsvektorer med ulike dimensjoner inkludert nulldimensjonsvektoren, det vil si diskrete symboler. Denne utvidete HMM modellen, kalt Multi-Space probability distributions HMM (MSD-HMM) modellerer spektrale parametere og fundamentalfrekvensen i et enhetlig rammeverk av HMM. Figur 5 viser en oversikt over en MSD-HMM.



Figur 5: En oversikt over MSD-HMM [25].

MSD-HMM inneholder både diskrete HMMer og kontinuerlige multimiks HMMer som spesialtilfeller. Dette kommer av at multi-space sannsynlighetsfordeling tar for seg både de diskrete og de kontinuerlige fordelingene.

3.3.1.4 Modellering av varighet

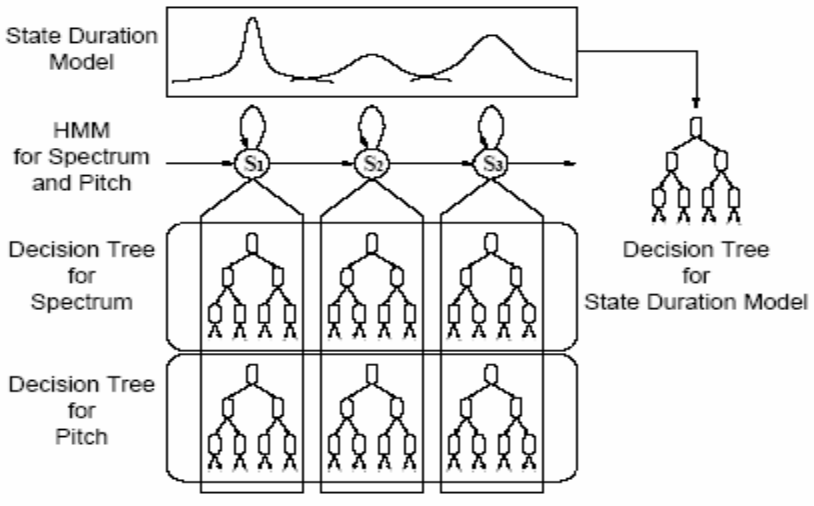
Virkemåten til HMMen fører til en indirekte varighetsmodellering som en får på grunn av transisjonssannsynlighetene. Fonemvarigheten får en eksponensiell fordeling. Dette stemmer ikke overens med hvordan varighet ser ut i den menneskelige talen. I HTS brukes eksplisitt en enkel gaussisk tetthetsfordeling til å modellere tilstandsvarighetstettheter [23]. Hver tilstand inneholder like mange tetthetsfordelinger som det er tilstander i HMMmodellen. Modellene i HTS vil da modelleres av 5-mixture HMMer hvor sannsynlighetstetthets-funksjonen består av fem Gaussiske tetthetsfunksjoner. Denne tilnærmingen har den fordelen at taleraten på den syntetiske stemmen lett kan varieres. Det er ikke nødvendig med merking av grenser når passende definerte modeller er tilgjengelige, siden tilstandsvarighetstettheten er estimert i den innlagte treningsdelen av fonem HMMene.

3.3.2 Kontekstavhengige modeller

I kontinuerlig tale kan en parametersekvens for en spesiell taleenhet (for eksempel fonem) variere i forhold til fonetisk og lingvistiske sammenhenger. For å modellere disse variasjoner for spektrum, f_0 og varighet, blir fonetiske, prosodiske og lingvistiske kontekstuelle faktorer som fonemidentitet, trykk og plassering i en stavelse tatt i betraktning. Ut fra denne informasjonen lages kontekstavhengige HMMer, som for eksempel trifoner som er et fonem som er avhengig av hvilket fonem som står foran og bak. Disse kontekstuelle faktorene varierer for hvert enkelt språk.

3.3.2.1 Kontekstklynging basert på desisjonstre

Ett stort problem med all datadrevet talesyntese er vanskeligheten med å samle inn tilstrekkelig med data, spesielt når kontekstavhengige modeller brukes. Etter hvert som antallet kontekstuelle faktorer øker, vil også kombinasjonen av de øke eksponentielt [23]. Problemet er at en ikke kan forberede databasen på absolutt alle kombinasjoner av kontekstuelle faktorer, derfor må en bruke treningsdataene mest mulig effektivt. Til dette brukes en teknikk kalt kontekstklynging. Ved å bruke informasjon fra en annen talelyd med lignende kontekstuelle egenskaper, blir det mulig å få en akseptabel tilnærming av den talelyden som mangler i treningsdatabasen. Denne grupperingen av modeller med liknende egenskaper, gjøres med desisjonstrær, der hver modell tilhører ett visst fonem som sorteres i en hierarkisk trestruktur ved hjelp av ja/nei spørsmål. Siden mcep, f_0 og varighet har ulike egenskaper, blir det laget forskjellige trær for hver av disse, se figur 6.



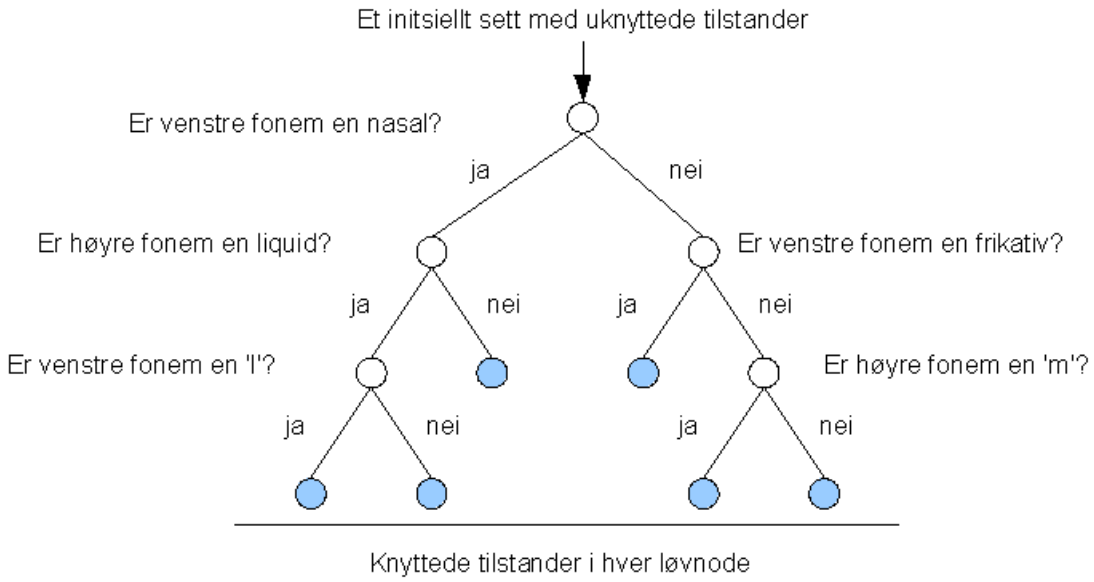
Figur 6: Desisjonstre for kontekstklynging [23].

I neste avsnitt blir det forklart hvordan et slikt tre er bygd opp.

3.3.2.2 Konstruksjon av desisjonstre

Et fonetisk desisjonstre er et binært tre hvor det er knyttet et spørsmål til hver enkelt node [28]. I systemet beskrevet her, er hvert av disse spørsmålene knyttet til den fonetiske konteksten, dvs. det nærmeste fonemet på høyre og venstre side. For eksempel, i figur 7 er spørsmålet ”er fonemet til venstre for det aktuelle fonemet nasalt?” assosiert med rot noden til treet. Et tre er konstruert for hver tilstand av hvert fonem for å klynge alle de samsvarende tilstandene til de assosierte trifonene. For eksempel, treet vist i figur 7 vil dele sine tilstander inn i seks delklynger som samsvarer med de seks endenodene. Tilstanden til hver delklynge er knyttet for å forme en enkelt tilstand, og spørsmålene og trestrukturen er valgt for å maksimere sannsynligheten for at treningsdataene gir disse

knyttede tilstandene. Samtidig forsikres det at det er tilstrekkelig data assosiert med hver enkelt knyttet tilstand for å estimere parameterne for en multimiks gaussisk blandingsfordeling. Når alle slike trær er blitt konstruert, kan trifoner som ikke finnes i treningsdatabasen bli syntetisert ved å finne de passende endenodene for konteksten til det trifonet, og så bruke dets knyttede tilstander assosiert med disse nodene for å konstruere trifonet.



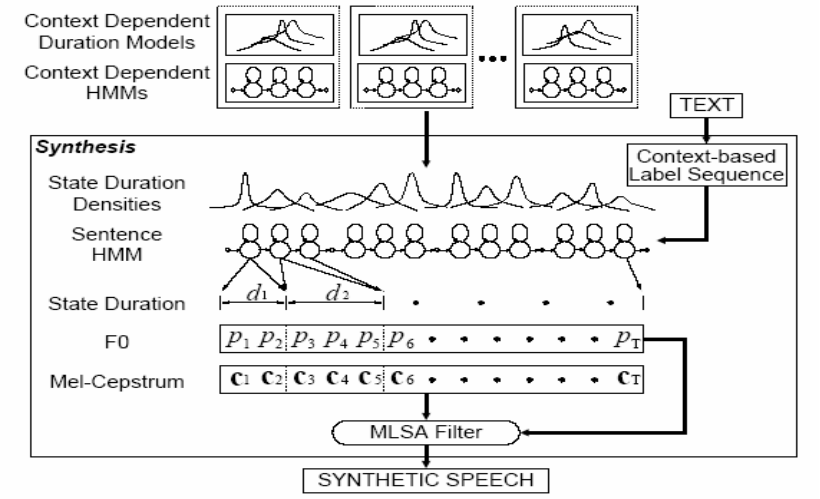
Figur 7: Eksempel på et fonetisk desisjonstre [28].

For å finne det spørsmålet som best deler en gruppe, brukes et delingskriterium. For å kunne bestemme hvor like et antall medlemmer i klynge er, trengs et avstandsmål. I talegjenkjenning har man tradisjonelt brukt Maximum Likelihood (ML) –kriteriet. Ulempen med dette kriteriet er at det må velges et separat stopp-kriterie for å begrense størrelsen på treet. Dette kriteriet kan ikke optimeres med noen kjent metode og er ofte empirisk bestemt og varierer fra gang til gang. I HTS brukes i stedet Minimum Description Length (MDL) –kriteriet [29]. MDL beregner antall informasjonsbærende symboler som beskriver samtlige medlemmer i klyngen. MDL-kriteriet er effektivt til å velge det spørsmålet som vil komprimere treet mest mulig. I tillegg definerer MDL, i form av en grenseverdi, når en skal stoppe oppdelingen, noe som har vist seg å gi gode resultater uten manuell justering.

3.3.3 Syntese

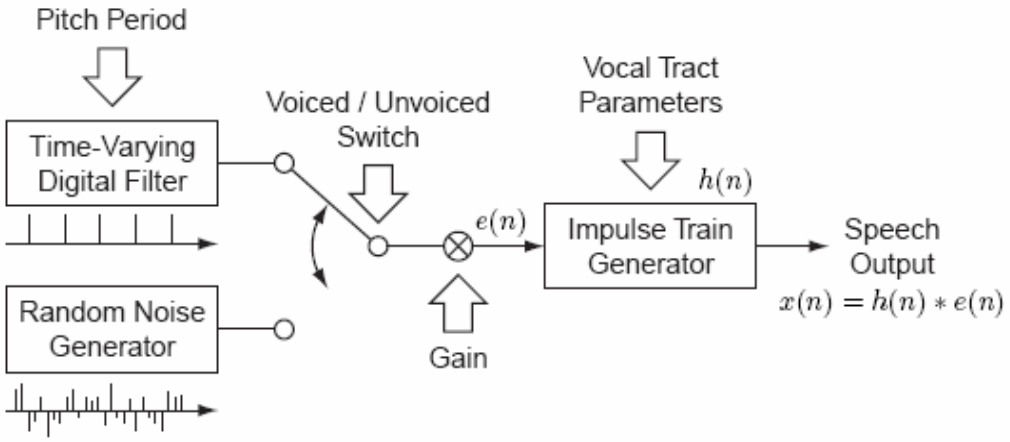
HTS systemet inkluderer ikke en egen tekstanalysator, så HTS bruker det generelle tekst-til-talesystemet Festival for å syntetisere HTS-stemmene. Først konverterer Festival en vilkårlig tekst til en kontekstavhengig merket sekvens [2]. I henhold til den merkede sekvensen blir en setnings-HMM laget ved å skjøte sammen kontekstavhengige fonem HMMer. Denne HMM-setningen tilsvarer den teksten som er skrevet inn til systemet, og det er denne som syntetiseres. For å kunne generere en tilstandssekvens med tilhørende parametersekvens ut i fra et antall HMMer, og i tillegg kunne dra nytte av de dynamiske koeffisientene (Δ -og ΔΔ koeffisienter), brukes en algoritme som maksimaliserer sannsynligheten for denne sekvensen av HMMer [24]. Denne algoritmen er iterativ og en fullstendig beskrivelse av den finnes i [12]. De dynamiske koeffisientene vil legge en viktig begrensing for hva en har muligheten til å generere. Koeffisientene forteller noe

om utviklingen i tid og brukes for å få en mest mulig glatt kurve. Med glatt menes at de tilfredsstiller kontinuitetsbetingelser som ligger i naturlig tale. Gangen i syntesen vises i figur 8.



Figur 8: HTS, syntesedelen av systemet [23].

Det siste steget for å generere syntetisk tale er å la de genererte parametrene styre en syntetisator. I HTS brukes en enkel signalkilde, *Excite*, som styres av f_0 og stemte/ustemte parametere sammen med et filter, *Mel Log Spectrum Approximation Filter* (MLSA). Excite genererer fra en pitsjperiode et pulstog for stemt tale (vibrasjon av stemmebåndet) eller for ustemt tale (turbulente luftstrømmer som oppstår ved innsnevring i ansatsrøret), en sekvens med gaussisk støy [30]. Signalet fra *Excite* filtreres gjennom MLSA-filteret [13] som styres av parametrene til spektret. Produktet vi får ut er syntetisk tale. Figur 9 viser kilde-filterkonseptet.



Figur 9: Kilde-filter modell for taleproduksjon [12].

Den HTS-syntetiserte talen kjennetegnes ved at den av og til har en ”vokodet” klang, en av grunnene til dette kan i ifølge [31] komme av at kilde-filtermodellen som brukes genererer signalene enten som stemte eller ustemte. For å forbedre kvaliteten kan man enten gjøre lyden klarere ved å kjøre signalene gjennom et filter som framhever og forsterker formantamplituden (formant emphasis filter), eller en kan bruke en sammensatt eksitasjonsmodell, MELP. Denne eksitasjonsmodellen er blitt implementert i HTS av Yoshimura, Tokuda et al. i [31], men den er ikke blitt implementert i basisversjonen av HTS-systemet.

I det neste kapittelet skal en se nærmere på ulike eksitasjonsmodeller og hvordan de modellerer menneskets taleorgan. En slik modell må kunne generere både periodiske signaler fra stemmebåndet og støy (som oppstår pga. turbulens i trange områder i talerøret, for eksempel svelget eller når tungen er i nærheten av gommen eller tennene, eller turbulente strømmer som oppstår etter at luften stoppes etter at vi har uttalt plosiver).
Disse signalene kan også overlappe hverandre, men da kreves en mer sofistikert modell som tillater såkalt blandet eksitasjon (mixed excitation).

4 Modeller for blandet (mixed) eksitasjon

I følge akustisk teori, består taleproduksjonsprosessen av en kildekomponent og et filter. Et av målene i taleanalyse er å studere karakteristikken til kilden og filteret ved å prosessere talesignalet. Tradisjonelt modelleres kilden som enten stemt eller ustemt. Kilden representerer luftstrømmer fra stemmebåndet, og filteret representerer resonansene fra ansatsrøret som endrer seg over tid [14]. Men den stemte delen i naturlig tale består også av uregelmessige komponenter; såkalte aperiodiske komponenter. Aperiodiske komponenter er tydelig i stemte frikativer (dvs. /v/, /z/), og pustende vokaler (dvs. høye vokaler i kontekst med ustemte konsonanter). Den aperiodiske komponenten finnes også i vanlige vokaler på grunn av at det oppstår turbulente luftstrømmer rundt øyeblikket når stemmebåndslappene lukkes, noe som gir økning av aspirasjonsstøy [14] s. 833.

Studiet av den aperiodiske komponenten av eksitasjonen er viktig i taleanalyse. Denne komponenten kan være til hjelp for å karakterisere stemmeegenskaper som pusting og uregelmessigheter. Pusting er assosiert med lufttap i stemmebåndet og turbulent støy. Uregelmessigheten defineres som nærværet av en lavfrekvent støykomponent. Ved å inkludere aperiodisk komponent i stemt eksitasjon kan en produsere en naturlig syntetisk tale [32]. Utførlig karakterisering av kilden kan ha betydning når en ønsker å generere syntetisk tale med ønsket stemmekarakteristikk.

Det meste en kjenner til om de tidsmessige egenskapene til talespekteret er anvendt i klassiske syntetisatorer som kontrollerer resonanser og kildeegenskaper, men kvaliteten på den syntetiske talen er svært unaturlig. Ekstrahering av et stabilt spektrum fra tale, har vært et av hovedmålene for ulike digitale signalprosesseringsalgoritmer i årenes løp. Flere klassiske metoder som kanal vokoderen (VOCODER) [33] og den moderne varianten av den, Linear Predictive Coding (LPC) [14] s. 353 er gode eksempler på slike metoder. En svakhet ved disse metodene er at den reproduserte talen høres svært unaturlig ut og lider av at stemmen har en ”robotaktig” klang. Flere metoder er blitt foreslått til å separere et talesignal i en periodisk og en aperiodisk komponent, slik at en enklere kan manipulere kildesignalene i den hensikt å redusere den syntetiske klangen på talen i disse *Vokoder*-metodene: de er basert på sinusformet modellering, på harmonisk plus støy modellering, flerbånds-eksitasjons vokoder (multiband excitation vocoder) [34] eller pitsj-adaptive metoder som brukes i STRAIGHT-modellen.

Det er i denne oppgaven blitt sett på to av metodene nevnt over: HNM og STRAIGHT. I de neste underkapitlene presenteres disse og i tillegg den tradisjonelle LPC-vokoderen. Det blir lagt spesiell vekt på STRAIGHT-systemet. STRAIGHT er modellen som er benyttet i denne masteroppgaven for å forbedre naturligheten i den norske HTS-stemmen.

4.1 Linear Prediction based Methods (LPC)

Lineære prediksjonsmetoder er opprinnelig designet for talekodingssystemer, men kan og brukes i talesyntese. Faktisk var den første talesyntetisatoren utviklet fra talekodere. På samme måte som for formantsyntese, så er det grunnleggende i LPC basert på kilde-filtermodell konseptet. Filterkoeffisientene estimeres automatisk fra hver ramme i talesignalet.

Det essensielle i lineær prediksjon er at et talesampel kan predikeres fra et endelig antall forgående p sampler $y(n-1)$ til $y(n-k)$ ved linear kombinasjon med en liten feil $e(n)$ som kalles restsignal (residual). Dermed

$$y(n) = e(n) + \sum_{k=1}^{p} a(k)y(n-k), \tag{4.1}$$

og

$$e(n) = y(n) - \sum_{k=1}^{p} a(k)y(n-k) = y(n) - \tilde{y(n)} \tag{4.2}$$

hvor $\tilde{y(n)}$ er den predikerte verdien, p er den lineære predikator ordenen, og $a(k)$ er den lineære prediksjonskoeffisienten som en finner ved å minimere summen av den kvadrerte feilen over en ramme. To metoder, kovariansmetoden og autokorrelasjonsmetoden er vanlige metoder å bruke for å beregne disse koeffisientene. Men bare med autokorrelasjonsmetoden garanteres et stabilt filter [14] s. 290.

I syntesen tilnærmes eksitasjonen som et pulstog for stemt lyd og hvit støy for ustemt lyd. Eksitasjonssignalet blir så filtrert med et filter hvor koeffisientene er $a(k)$. Filterordenen er vanligvis mellom 10 og 12 for en samplingsrate ved 8 kHz, men for høyere kvalitet ved 22 kHz samplingsrate, må ordenen være mellom 20 og 24. Koeffisientene oppdateres hvert 5-10ms.

Den største mangelen ved denne metoden er at den representerer en allpol-modell. Det betyr at fonemer som inneholder antiformanter som nasaler og nasaliserte vokaler blir dårlig modellert. Kvaliteten er også dårlig for korte plosiver fordi tidsrammen på lyden kan være kortere enn rammestørrelsen brukt i analysen. Med disse manglene betraktes kvaliteten på den syntetiske talen ved bruk av LPC-metoden som dårlig, men med noen modifikasjoner og utvidelser av basismetoden kan kvaliteten øke. Flere variasjoner av lineær prediksjon er blitt utviklet for å øke kvaliteten på basismetoden [35]. Med disse metodene er eksitasjonssignalet som brukes forskjellig fra den vanlige LP-metoden, og kilde og filter er ikke lenger separert.

4.2 HNM (Harmonic plus Noise Model)

Stylianou har utviklet en eksitasjonsmodell kalt Harmonic plus Noise Model (HNM) [7], hvor pitsjsynkron analyse brukes for harmonisk dekomponering. I HNM representeres talesignalene som en tidsvarierende harmonisk komponent pluss en modulert støykomponent. Den *harmoniske komponenten* fremstiller den kvasi-periodiske strukturen til talesignalet, mens *støykomponenten* fremstiller den ikke-periodiske strukturen til talesignalet, slik som frikativer, pusting eller periode-til-periode variasjoner av stemmebåndseksitasjon, osv. De to komponentene er separert i frekvensdomenet av en tidsvarierende parameter; frekvensen til den høysete stemte frekvensen, F_m. Over denne frekvensen representeres talen ved støy, under denne frekvensen består talesignalet av harmoniske komponenter med sakte varierende amplituder og frekvenser. Denne fremstillingen av talesignalet er ikke gyldig hvis en ser på teorien som omhandler taleproduksjon. Stemt talesignal er kvasi-periodisk, dvs. at lave frekvenser også

inneholder en støykomponent, og høye frekvenser inneholder både støy og kvasi-periodiske komponenter [7]. Til tross for dette gir denne enkle talemodellen perseptuelt en høy kvalitet på den syntetiske talen.

Analysen i HNM er basert på ramme-for-ramme prinsippet. Parametrene estimeres på grunnlag av om rammen er stemt eller ustemt. Er rammen stemt, estimeres pitsj, maksimum stemt frekvens, amplituder og faser av harmoniske til fundamentalfrekvensen. Hvis rammen er ustemt, estimeres energienvelopen og LPC koeffisientene (tilsvarer LPC-allpol filteret som brukes i støydelen) [7]. Analysen og syntesen i HNM er pitsjsynkron, derfor er det nødvendig å estimere stemmebåndslukkingene (Glottal closure instants, GCI) nøyaktig.

Parametrene oppnådd i analysen brukes så i syntesen. I syntesen brukes en pitsjsynkron ”overlap and add”-metode. Den harmoniske delen syntetiseres direkte i tidsdomenet som en sum av harmoniske. Støydelen oppnås ved å filtrere enhetsvarians av hvit, gaussisk støy gjennom et normalisert allpol-filter. Hvis rammen er stemt, filtreres støydelen med et høypass filter med en beskjæringsfrekvens lik F_m. Deretter moduleres den med en envelope i tidsdomenet synkronisert med pitsjperioden. Denne moduleringen av støydelen viser seg å være nødvendig for å sikre naturlighet i noen talelyder som f.eks. stemte frikativer [36]. Ved å legge sammen den syntetiserte harmoniske med den syntetiserte støydelen, fås syntetisert tale.

Det finnes fire ulike teknikker [37] for generering av det harmoniske signalet i HNM. Mer enn 80 % av kjøretiden i HNM syntesemodulen, brukes for å generere det harmoniske signalet. Disse er teknikkene er forklart i [37], og vil ikke bli beskrevet her.

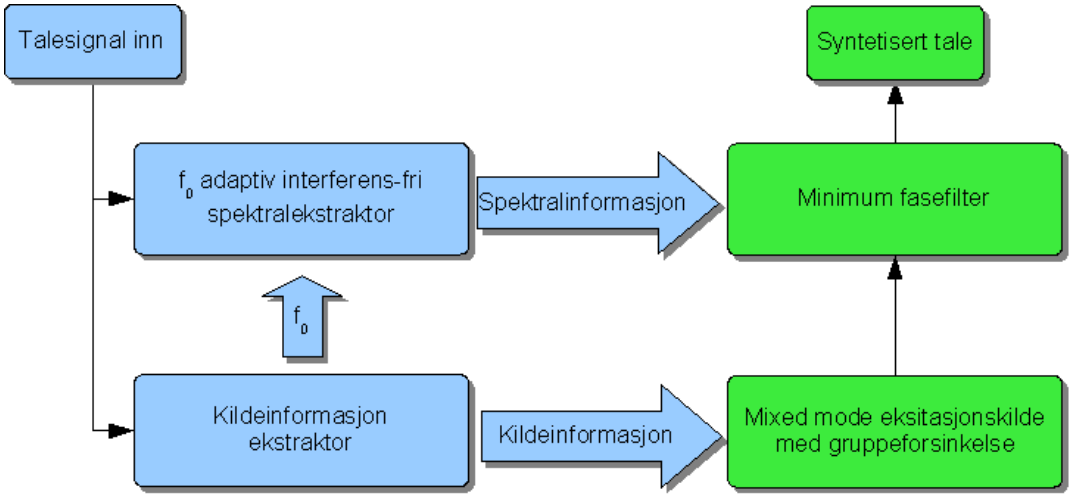
Denne parametriske modellen til Stylianou gjør det mulig å endre talerate, intonasjonskurve og talerkarakteristikk. HNM er blitt implementert i skjøtesyntese og har vist å gi god forståelighet og naturlighet på den syntetiske stemmen [7]. Den parametriske representasjonen av tale gjør det enkelt å kunne glatte ut diskontinuiteter i skjøtepunktet mellom to lydenheter. Men modellen inneholder ikke hele transformasjonen av kildesignalet til talen. Bare fundamentalfrekvensen blir endret. De høye harmoniske frekvensene separeres fra f_0 i analysetrinnet, og det er ingen forskjell mellom dem og filterkomponentene (formantene). De blir reprodusert uten noen form for transformasjon. Dette kan være en ulempe, siden høye harmoniske av pitsjen kan være svært fremtredene i talesignalet, og deres frekvenser burde også bli transformert sammen med fundamentalfrekvensen.

4.3 STRAIGHT

Kawahara, Masuda-Kasuse, og de Cheveigné har utviklet en høykvalitets vokoder kalt STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weighted spectrum) [6], som bruker den klassiske kilde-filter teorien til Dudley [33] for å separere kilde- og filterinformasjon fra hverandre. STRAIGHT består av tre hoveddeler:

- ekstrahering av kildeinformasjon basert på fastpunktsanalyse
- f_0-adaptiv glatting av spekteret som fjerner periodisk interferens forårsaket av signalperiodisitet i spektrogrammet
- resyntese ved bruk av manipulasjon av gruppeforsinkelse for å kontrollere tidsmessige strukturer i en eksiterende kilde

En oversikt over STRAIGHT-systemet er vist i figuren under.



Figur 10: Oversikt over STRAIGHT-systemet.

Den største forskjellen i forhold til tidligere metoder er at i STRAIGHT legger en til mer informasjon istedenfor å redusere informasjonen. Dette gir stor mulighet for manipulasjon [6]. Vi skal i de neste underkapitlene se nærmere på de tre hoveddelene STRAIGHT består av.

4.3.1 Ekstrahering av kildeinformasjon

Kildeinformasjonen i STRAIGHT består av fundamentalfrekvens (f_0) og aperiodiske komponenter. Begge disse parametrene er avhengig av en ekstraheringsmetode kalt fastpunktsanalyse [38].

I de neste underkapitlene gis det en nærmere beskrivelse av hvordan disse parametrene trekkes ut.

4.3.1.1 Ekstrahering av fundamentalfrekvens, f_0

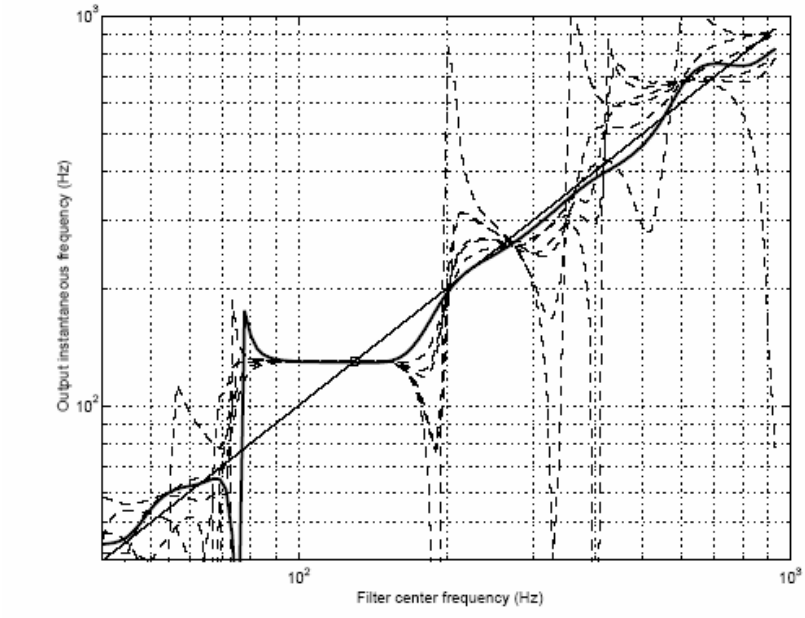
Analyse av naturlig tale er ingen enkel oppgave. Fundamentalfrekvensen i talesignaler endres hele tiden, og hver repetisjon er aldri lik forrige periode på grunn av de vekslende bølgene fra kilden. Tradisjonelle f_0-estimeringsmetoder er basert på intervall målinger som gir oppstykkede f_0-kurver. Disse oppstykkede kurvene gir periodisitet til kilden og fundamentalfrekvensen vil da være mer skjør ovenfor modifikasjoner [6]. Det en ønsker er en metode som gir kontinuerlige f_0-kurver.

STRAIGHT har en f_0-estimeringsmetode som antar at talesignalet har en tilnærmet harmonisk struktur [6],

$$x(t) = \sum_{k=1}^{N} a_k(t) \cos\left(\int_0^t k w_0(\tau) + w_k(\tau) d\tau + \phi_k(0) \right), \qquad (4.3)$$

hvor a_k representerer sakte endrende momentanamplitude, $w_k(\tau)$ representerer sakte endrende forstyrrelser fra den k'te ordens harmoniske komponent. I denne representasjonen er f_0 momentanfrekvensen til fundamentalkomponenten hvor k=1. I uttrekningen av f_0 brukes momentanfrekvenser av andre harmoniske komponenter for å forbedre f_0 estimater. Ved å bruke et båndpass filter som sentrerer rundt f_0 og ekskluderer andre komponenter, får man en kontinuerlig og høytoppløselig f_0 kurve. For å kunne velge fundamentalkomponenten kreves det imidlertid kjennskap til fundamentalfrekvensen på forhånd. Et konsept kalt "fundamentalhet" løser dette problemet uten å bruke a priori kjennskap om fundamentalfrekvensen. Båndpassfiltre med lik form på log-frekvensaksen brukes til å ekstrahere fastpunkter [38] fra filtersenterfrekvensen til momentanfrekvensen på utgangen av filteret. Disse fastpunktene evalueres i forhold til estimerte carrier-to-noise rater (C/N), for å velge et fastpunkt som tilsvarer fundamentalfrekvensen. "Fundamentalheten" er designet slik at den vil ha en maksimumsverdi bare når en harmonisk komponent oppholder seg i passbåndet av et båndpassfilter [6]. Det vil si at den ønskede fundamentalfrekvensen oppnås som momentanfrekvensen til høyeste "fundamental" filterutgang. Filteret som brukes er laget fra en isometrisk Gabor-funksjon foldet med en kardinal B-spline funksjon. Denne kombinasjonen gir filteret mulighet til å gi ulike egenskaper til fastpunktene, som gir anvendelige C/N-rate estimater til de korresponderende komponenter.

Figuren under illustrer hvordan det log-lineære filteret er ordnet som gir sammenhengen mellom fundamentalkomponenten og fastpunktet. En ser av figuren at det bare er i nærheten av fundamentalfrekvensen en får stabil og flat mapping.



Figur 11: Illustrerer sammenhengen mellom fundamentalkomponenter og fastpunkter [38].

Momentanfrekvensen gir en god beskrivelse av signaler som endrer frekvens med tiden. Men til tross for dette, brukes det i de fleste sinusformede metoder "peak picking" over spektrumet uten noen referanse til momentanfrekvensen. I "peak picking" teknikken dukker det opp mange uekte topper i søket etter de høyeste toppene, dette gjelder spesielt for analyse basert på fastintervall målinger [39]. I [39] refereres det til Abe et al. som

beskriver et momentanfrekvens-amplitudespektrum. Dette spektrumet viser klarere de harmoniske strukturene som finnes i kvasi-periodiske signaler, som for eksempel tale, i forhold til tradisjonelle spektrogram som for eksempel kortids Fouriertransform (STFT). Det har og vist seg at ved å bruke momentanfrekvenser oppnås en robust og nøyaktig estimeringen av f_0-kurver [38].

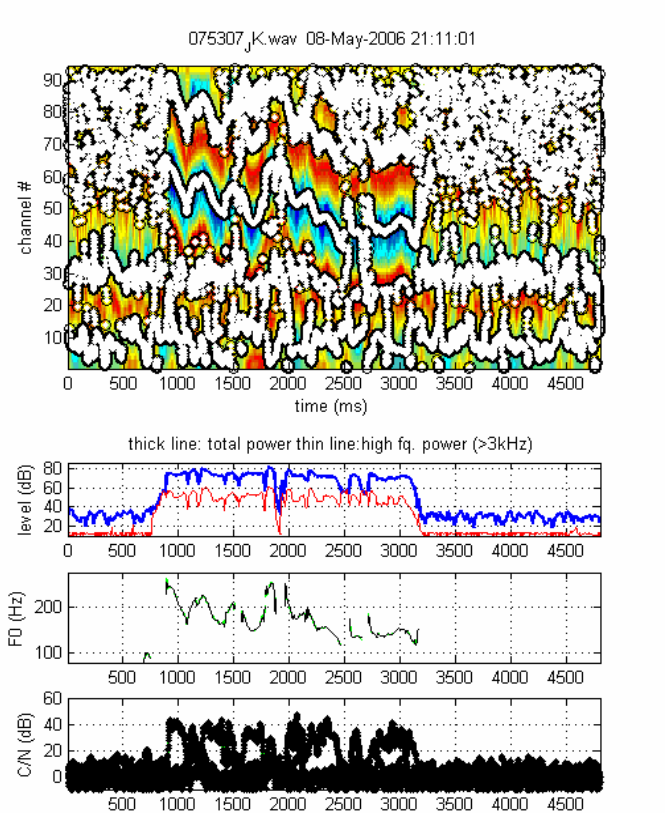
4.3.1.2 Aperiodiske verdier

Metoden som ekstraherer aperiodiske komponenter i STRAIGHT, er basert på en analyse utført i både tidsdomenet og i frekvensdomenet ved bruk av momentanfrekvenser og gruppeforsinkelse. Signalrepresentasjonen består av aperiodiske mål i frekvensdomenet og energikonsentrasjoner i tidsdomenet som representerer kildeegenskapene [40].
Aperiodiske mål i frekvensdomenet er definert som raten mellom nedre og øvre glattede spektralenvelope, som representere den relative energifordelingen til de aperiodiske komponentene. Målene i tidsdomenet er definert som varigheten til den aperiodiske komponenten og varigheten til hvordan en hendelse utvikler seg [41]. En hendelse kan for eksempel være når stemmebåndslappene lukkes, dette vises som høy energikonsentrasjon spesielt ved høye frekvenser. Disse aperiodiske komponentene og f_0 som tidsfunksjoner brukes til å generere kildesignalet til den syntetiske talen. Dette gjøres ved å kontrollere det relative støynivået og tidsenvelopen til støykomponenten til det sammensatte eksitasjonssignalet, sammen med timing og amplitudefluktuasjoner.

Den aperiodiske indeksen introdusert her har likheter med en dekomponeringsmetode av talesignalet i periodiske og aperiodiske komponenter presentert i [32].

4.3.1.3 Kildeinformasjon i STRAIGHT

I figuren under vises ekstrahert kildeinformasjon for den norske setningen ”*Fire ganger i året kommer denne essaysamlingen*.”



Figur 12: Kildeinformasjon fra en norsk setning ekstrahert av STRAIGHT.

Det øverste bildet representerer kandidater av f_0 og deres pålitelighet på betingelse av C/N-raten. Prikkene på bildet er f_0 kandidater ekstrahert fra fastpunktsanalysen. Fargene i bildet representerer C/N-raten. C/N-raten står for carrier-to-noise raten, hvor carrier tilsvarer f_0-komponenten. Den blå streken tilsvarer høy C/N-verdi og rødfargen tilsvarer lav C/N-verdi. Den kandidaten som har høyest C/N-verdi, er den beste f_0-kandidaten i hver ramme.

Neste graf viser den totale energikurven (blå strek) og energi i de høye frekvensområdene (rød strek). Frekvensområdet høyere enn 3 kHz er definert som det høyeste frekvensområdet i STRAIGHT implementasjonen. Denne informasjonen brukes for å bestemme stemthet/ustemthet i kildeinformasjonen.

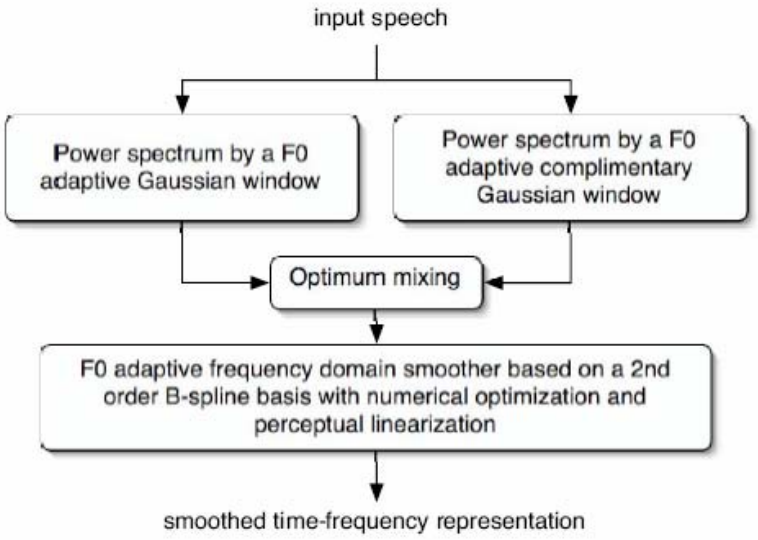
I den nest nederste grafen vises den ekstraherte f_0-kurven. Det gjøres først et initsiellt estimat av f_0 basert på den første avbildningen. Deretter korrigeres f_0-kurven basert på de tre laveste harmoniske komponentene.

Siste grafen representerer C/N-rater for f_0-kandidater.

4.3.2 F_0-adaptivt glattet spektrum

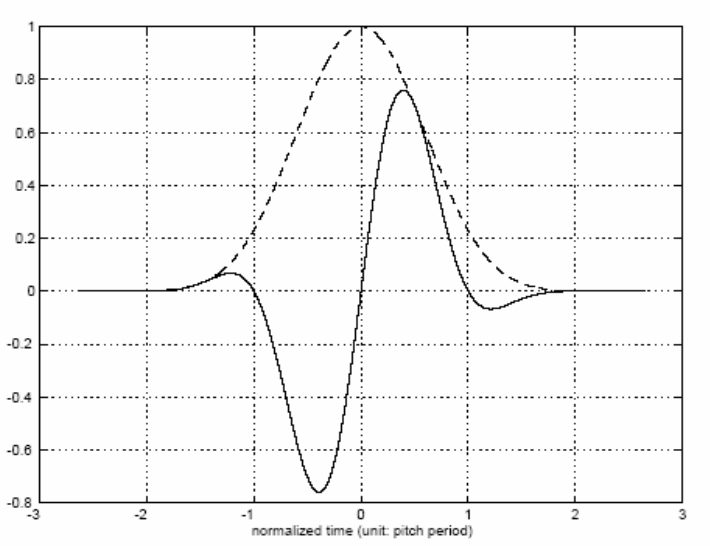
Ved å bruke f_0, utfører STRAIGHT en pitsj-adaptiv spektralanalyse i kombinasjon med en overflate-rekonstruksjonsmetode. Denne analysen utføres i både tidsdomenet og frekvensdomenet for å fjerne signalperiodisitet. Det grunnleggende prinsippet i STRAIGHT-metoden er å lage et spektrogram som ikke inneholder regelmessige strukturer i tidsdomenet, slik at problemet med å rekonstruere overflaten i tids- og frekvensdomenet reduseres til å bli et problem for å eliminere regelmessige strukturer i frekvensdomenet [6]. Tids- og frekvensrepresentasjonen er i grunnen en glattet versjon av det tradisjonelle talespektrogrammet.

Talesignalet er ikke periodisk og endres over tid. Dynamiske endringer er essensielle for å overbringe lingvistisk informasjon og for hvordan ord skal sies, dvs. volum, intonasjon, hastighet osv. I STRAIGHT brukes et kompensasjonsvindu sammen med et adaptivt gaussisk vindu. Disse to vinduene rekonstruerer til sammen en glatt representasjon av spekteret i tid og frekvens, uten spor av interferens fremkalt av signalperiodisitet under vindusprosessen. Figuren under illustrerer hvordan det glattede spektrogrammet konstrueres.



Figur 13: Interferensfritt spektrum implementert i STRAIGHT [42]

Kompensasjonsvinduet er på en slik form at det vil lage maksimum der det originale vinduet lager hull. Vindusformen er vist i figuren under.



Figur 14: Kompansasjonstidsvindu (heltrukket linje) og originalt tidsvindu (striplet linje) [6].

Effektspektrumet som bruker dette tidsvinduet reduserer faseinterferensen. Et slikt effektspektrogram $P_r(w,t)$ er en representasjon av en vektet kvadratisk sum av effektspektrene $P_c(w,t)$ og $P_o(w,t)$. Disse effektspektrene er blitt til ved bruk av henholdsvis kompensasjonsvinduet og det originale tidsvinduet. Likning (4.4) viser det totale effektspekteret:

$$P_r(w,t) = \sqrt{P_o^2(w,t) + \xi P_c^2(w,t)} \qquad (4.4)$$

hvor ξ minimerer tidsmessige variasjoner i det resulterende spektrogrammet.

Neste trinn er å eliminere spektralinterferens forårsaket av signalperiodisitet. Tidligere metoder for å estimere spektralenvelopen til det periodiske signalet, er ofte basert på å estimere parametere basert på parametrisk spektralmodell. Disse modellene er vanligvis følsomme mot unøyaktighet mellom modell og virkelig tale, og feil i estimeringen av f_0. I STRAIGHT brukes en ikke-parametrisk modell som er motstandsdyktig mot f_0-feil. Metoden bruker en kardinal B-spline glattingsfunksjon som eliminerer periodiske effekter [6]. I implementasjonen brukes en 2. ordens kardinal B-spline basis $h_t(w)$:

$$h_t(w) = 1 - \left| \frac{w}{w_0(t)} \right| \qquad (4.5)$$

hvor $w_0(t) = 2\pi f_0(t)$ og $-w_0(t) \le w \le w_0(t)$. Siden den fundamentale vinkelfrekvensen $w_0(t)$ er en funksjon av tid, så er glattingsfunksjonen adaptiv til fundamentalfrekvensen. Det glattede spektrogrammet ved tid t beregnes ved å bruke denne glattingsfunksjonen og et effektspektrum med redusert faseinterferens.

I det glattede spektrogrammet:

$$S(w,t) = \sqrt{g^{-1}(\int_D h_t(\lambda) g(|P_r(w-\lambda,t)|^2) d\lambda} \qquad (4.6)$$

representerer D støtten fra glattingsfunksjonen $h_t(w)$. Denne operasjonen gjøres både i tidsdomenet og i frekvensdomenet, slik at prosessen blir mindre sensitiv mot f_0-feil.

Et problem med denne algoritmen, er overglatting. Dette oppstår på grunn av usikkerhet mellom tid og frekvens, slik at glattingsfunksjonen $h_t(w)$ må operere på den allerede glattede spektralrepresentasjonen. Hvis en betrakter den foreslåtte glattingsfunksjonen til å være 2. ordens kardinal B-spline med lokalisert normalisering er det mulig å lage en glattingsfunksjon som kan kompensere for overglatting. Dette reduserer problemet til være et inverst filtrerings problem, det vil si å gjenvinne den originale impulsen fra den glattede impulsen.

4.3.3 Resyntese

STRAIGHT er i stand til å resyntetisere ulike stemmekvaliteter fra et sett med parametriske representasjoner; glattet spektrogram, f_0 og aperiodiske indekser representert i tid og frekvens. Talekvaliteten er avhengig av disse parametrene.

Under syntesen lager STRAIGHT en sammensatt strøm som inneholder en vektet sum av pulstog med fasemanipulasjon og gaussisk støy. Vektingsprosessen utføres i frekvensdomenet og det er her de aperiodiske indeksene blir brukt. Det variable filteret er implementert ved å bruke minimum faseimpulsrespons og en ”overlap and add”-metode (PSOLA). I kildegenereringen anvendes det allpass-filtre for å redusere den summende klangen som kommer fra tradisjonell pulseksitasjon [6].
Minimum faserespons brukes fordi hørselen til mennesket er sensitivt ovenfor spesielle typer av fasekarakteristikk som tidsmessig asymmetri. Ved å implementere allpass-filtre vil en kontrollere disse tidsmessige strukturene godt. I konstruksjonen av allpass-filtrene brukes gruppeforsinkelse som bedre kontrollerer de tidmessige strukturene i talesignalet enn ved å bruke fasekarakteristikkene direkte. Gruppeforsinkelse representerer tidsenergien i massesentrumet (temporal energy centroid) for hver harmoniske frekvens. Den syntetiserte talekvaliteten i STRAIGHT er svært høy [6].

5 HMM-basert talesyntese basert på STRAIGHT-vokoding

I denne masteroppgaven er det sett på hvordan en kan oppnå en mer naturlig, syntetisk tale basert på HMM. I en basisversjon av norsk, HMM-basert talesyntese [5] anvendes en enkel signalkilde som enten generer støy eller en bølgeform som oppstår når stemmebåndslappene vibrerer i strupehodet. Dette er en veldig forenklet modell av vårt taleorgan. Ved å bestemme om det er enten stemt eller ustemt tale, oppstår det feil, spesielt for talen som inneholder støy og for stemte frikativer. Blandede eksitasjonsmodeller som setter sammen stemt og ustemt tale til et signal, gir en mer presis representasjon av talesignalet. STRAIGHT er en modell som har vist å gi høy naturlighet på den resyntetiserte talen. En implementasjon av denne modellen er gjort i et japansk HTS-system med gode resultater [8]. Det er denne modellen som er implementert i denne oppgaven i den hensikt å forbedre naturligheten til den norske HTS-stemmen.

Alle språk bruker intonasjon for å uttrykke følelser, talehandlinger (utsagn, spørsmål, bekreftelse), holdning eller andre slike nyanser. Men ikke alle språk bruker toner for å skille leksikalsk betydning av et ord. Når dette oppstår, er toner fonemer, akkurat som vokaler og konsonanter, disse tonene kalles tonelag eller tonemer. I basisversjonen av norsk HMM-basert talesyntese er ikke tonelag, en karakteristisk egenskap ved det norske språket, blitt tatt med. Tonelag er tatt med i denne implementasjonen av norsk HTS-stemme.

Analyse/syntese systemet STRAIGHT [6] brukes til å ekstrahere taleparametrene: fundamentalfrekvens, aperiodiske indekser og spektrumsparametere, som brukes i treningen av HTS-systemet. Verktøyet Speech Signal Processing Toolkit (SPTK) [30] brukes til å lage egenskapsvektor med taleparametrene. HMM-based Synthesis System Toolkit (HTS) [19], en modifisert versjon av HTK, brukes for trening av selve systemet. STRAIGHT-systemet brukes deretter til selve syntetiseringen av stemmen, hvor den bruker blandet eksitasjon til å syntetisere tale. En steg for steg oppskrift for hvordan implementere STRAIGHT til HTS-systemt er beskrevet i appendiks A.1. De komponentene som utgjør den forbedrede norske HTS-stemmen er inkludert på medfølgende CDer i appendiks A.6.

5.1 Forstudie

For å få en bedre forståelse for hva som er karakteristisk ved det norske språket og hvilke egenskaper som er viktige og har betydning for naturligheten i den norske talen, ble det brukt tid på å sette seg inn i teorien rundt emnet. Deretter ble disse egenskapene implementert i HTS-stemmen.
For å finne en modell for blandet eksitasjon, ble to analyse/syntese systemer studert. HNM-koder (Harmonics plus Noise Model) [42] har vist å gi god lydkvalitet til NextGen-systemet. I tillegg tillater HNM å utføre modifikasjon og koding med lav kompleksitet. STRAIGHT, det andre systemet som er studert, er en vokoder som har vist å gi høy naturlighet til den resyntetiserte talen. Denne metoden eliminerer periodisk interferens i naturlig tale og den ekstraherer jevne og kontinuerlige f_0-kurver. I denne masteroppgaven ble STRAIGHT valgt som den modellen som skulle implementeres i norsk, HMM-basert talesyntese. Grunnen til dette valget var at verktøyene som brukes i

STRAIGHT-modellen lå til rette og som nevnt er STRAIGHT allerede blitt implementert i et japansk HTS-system [8].

5.2 Verktøy brukt i den norske HTS-syntesen

HTS benytter seg av fire ulike verktøy for å trene og syntetisere en ny, norsk HTS-stemme.

STRAIGHT

STRAIGHT er et system som kan gjøre taleanalyse, modifisere talesignaler og resyntetisere tale. Systemet er implementert i MatLab, og fungerer på flere ulike operativsystemer som for eksempel Mac OS X, Windows (2000, XP) og Unix.
STRAIGHT inneholder fire hovedfunksjoner som alle er skrevet i MatLabkode. Disse funksjonene er: exstraightsource, exstraightspec, exstraightsynth og exstraightAPind. Ved å bruke disse funksjonene kan en ekstrahere taleparametere som kan brukes til å trene skjulte Markovmodeller i HTS-systemet, og tilslutt syntetisere tale fra parametrene inneholdt i HMMene.

HTK (Hidden Markov Modell Toolkit)

Hidden Markov Modell Toolkit er en integrert pakke med verktøy for å kunne bygge og manipulere skjulte Markovmodeller (HMM). Programmet består av ett sett med biblioteker og ett sett med flere enn 20 verktøy moduler [43].

HTK ble i utgangspunktet laget til bruk i talegjenkjenning, men den kan og brukes til talesyntese ved å gjøre noen endringer. HTS bruker HTK for trening av HMM-modeller sammen med SPTK (Speech Signal Processing Toolkit, se avsnitt under). I HTS er det gjort følgende endringer i HTK [19]:

- Kontekstklynging basert på MDL kriterium i stedet for ML kriterium.
- Strømavhengig kontekstklynging
- MSD-HMMer for modellering av f_0
- Modellering og klynging av tilstandsvarigheter

SPTK (Speech Signal Processing Toolkit)

Speech Signal Processing Toolkit er en samling verktøy for prosessering av talesignaler. SPTK er gratis programvare og fungerer på unix-baserte operativsystemer. HTS-systemet bruker verktøy fra SPTK til å klargjøre taleparametere for trening av HMMer og ekstraherer spektrumsparametere ved å bruke verktøyet *mcep* [30].

Festival

HTS systemet er ikke et komplett TTS system. Systemet er avhengig av en ekstern enhet som kan omforme skrevet tekst til en fonemsekvens og markere prosodien i teksten. I den siste versjonen av Festival (1.95beta) er det lagt inn støtte for HMM-basert

talesyntese. Festival [44] er et frittstående program og tilbyr en flyttbar, språkuavhengig talesyntesemotor for ulike plattformer og under forskjellige APIer. Festival bruker regler skrevet i programmeringsspråket Scheme. Disse reglene brukes til å beskrive enkle syntaktiske og semantiske regler, og for mer komplekse substitusjoner som involverer grammatisk kunnskap.

For at Festival skal kunne bruke informasjonen som finnes i taledatabasen må det bygges en ytringsstruktur for hver ytring som finnes i databasen. Ytringsstrukturen inneholder relasjoner til ulike elementer som ord, stavelser og fraser. Denne ytringsstrukturen brukes i selve prosessen fra tekst til tale. Ytringen inneholder en enkel streng av tegn som konverteres steg for steg og fyller ut ytringsstrukturen med informasjon, helt til talebølgeformen er bygd. I prosjektoppgaven brukes et norsk TTS system, Talsmann [45] til å lage ytringer (forprosesseringen) som syntetiseres av Festival.

5.3 Egenutviklede verktøy

I dette arbeidet er det blitt utviklet et antall verktøy for å automatisere uttrekningen av taleparametere fra taledatabasen. I tillegg er det laget enklere verktøy for å konvertere datafiler mellom ulike formater. Verktøyene blir beskrevet under.

5.3.1 Verktøy som ekstraherer taleparametere fra taledatabasen ved bruk av STRAIGHT

Et enkelt skript er blitt laget, *run_straight.sh*, som kaller en kode i MatLab; *lene_make_straightparam.m* og verktøyet *mcep_dyre*. Dette skriptet automatiserer prosessen med å trekke ut parametere fra taledatabasen ved bruk av verktøyene i STRAIGHT og SPTK. MatLabkoden *lene_make_straightparam* leser inn wav-filer fra FonDat1-taledatabasen, ekstraherer taleparametrene f_0, aperiodiske parametere (ap) og spektrogram, og skriver de til fil. Disse filene lagres med en samplingsfrekvens på 16 kHz og rammeskiftsperiode på 1ms.

Mel-kepstral koeffisienter
STRAIGHT lager et glattet spektrogram basert på pitsj-adaptiv metode. I HTS-systemet lages HMMmodeller fra Mel-kepstral koeffisienter som trekkes ut fra et talespektrogram. Verktøyet *mcep_dyre* er en modifisert utgave av SPTK verktøyet *mcep*. *mcep_dyre* brukes til å trekke ut Mel-kepstral koeffisienter fra STRAIGHT-spektrogrammet og lagrer de til fil. I denne oppgaven er det brukt 40 Mel-kepstral koeffisienter inkludert den 0te koeffisienten.

Aperiodiske indekser
Aperiodiske mål i frekvensdomenet, basert på raten mellom øvre og nedre glattede spektralenvelope, representerer den relative energifordelingen til den aperiodiske komponenten. Disse parametrene ekstraheres i *lene_make_straightparam.m*. De aperiodiske parametrene er representert som en 513 x (lengde på fil) matrise. Ved å kjøre MatLabkoden *ap_fivefreq_band,* tas gjennomsnittet av de aperiodiske parametrene over fem frekvensbånd og datamengden reduseres.

Tilslutt må disse taleparametrene konvertere rammeskiftsperioden fra 1ms til 5ms for at det skal være tilpasset rammeskiftsperioden HTS-systemet bruker. Denne konverteringen

gjøres ved å bruke verktøyet *ch_track* fra festival, *x2x* verktøyet fra SPTK og et utviklet skript skrevet i python. En mer utførlig beskrivelse hvordan dette gjøres er beskrevet i appendiks A.1.

5.3.2 Modifisert treningsskript for norsk HTS-stemme

I programpakken HTS versjon 1.1.1 følger det med skript som er nødvendig for trening og syntese. Treningsskriptet for trening av HMMmodeller er modifisert slik at det er tilpasset taleparametrene som blir laget fra STRAIGHT.

Prototyp HMMmodellen ble endret slik at den kan ta inn aperiodiske komponenter i tillegg til mcep og f_0, for å modellere tale. I tillegg måtte det legges til rette for klynging og konstruksjon av desisjonstrær for aperiodiske komponenter. Endringen som er gjort er markert med farge i en tekstfil av treningsskriptet. Dette ligger vedlagt i CD i appendiks A.6.

HMGenS er et verktøy i SPTK, spesielt laget for HTS-systemet. Verktøyet brukes til å generere parametersekvenser fra trente HMMer. Dette verktøyet ble modifisert slik at aperiodiske parametersekvenser genereres i tillegg til fundamentalfrekvens og mcep parametere. Testsetninger fra taledatabasen blir generert med dette verktøyet.

5.3.3 Tonelag i HTS-stemmen

I skriptet *utt2lab.sh* som følger med i programpakken HTS versjon 1.1.1, konverteres ytringsfiler fra Festival til kontekstavhengige merkefiler og kontekstavhengige segment merkefiler for HTS. Toneme er lagt til i *utt2lab.sh* på stavelsesnivå for å bedre naturligheten i norsk HTS-stemme. Stavelser er ”bærere” av fenomenet tonelag, eller såkalte tonemer som det også kalles.

I tillegg er det laget spørsmål i spørsmålsfilen *questions_qst001.hed* relatert til kontekstklynging av tonelag.

5.4 Fonema referansedatabase (FonDat1)

Formålet med FonDat1 databasen er at dens innhold skal brukes til å eksperimentere med verktøy for automatisk segmentering og prosodisk merking av tale for datadrevet skjøtesyntese. Databasen inneholder 2092 forskjellige setninger som er innlest av to talere, en mann og en kvinne [46]. De har lest inn de samme setningene. Kvinnestemmen, JK, er brukt i denne masteroppgaven. I databasen finnes 200 setninger som er manuelt korrigert. Disse 200 setningene ble brukt for å lage HTS-stemme med taleparametere ekstrahert med STRAIGHT.

5.5 Språkspesifikke tilpasninger

HTS systemet er allerede tilpasset flere språk som japansk, portugisisk, svensk og engelsk. Arbeidet med å trene en ny stemme er tilnærmet automatisk og det kreves små endringer for å tilpasse HTS systemet til nye språk.

5.5.1 Fonemoppsett

I de fleste språk vil ikke en skrevet tekst tilsvare hvordan en uttaler teksten, så for å kunne beskrive riktig uttale trenger man en form for symbolsk lydrepresentasjon. Hvert språk har forskjellig fonetisk alfabet og forskjellig sett med mulige fonemer og kombinasjoner av disse. Et fonemsett kan bli definert som et minimum antall av symboler som trengs for å beskrive hvert ord som er mulig i et språk. Tabell 1 viser et alfabet for norsk, som er tilpasset for å fungere med HTK og Festival. På norsk har vi i alt 51 fonemer inkludert symboler for stillhet og kort pause.

Korte vokaler	A, o, e, ae, oe, ou, i, u, y
Lange vokaler	A:, o:, e:, ae:, oe:, ou:, i:, u:, y:
Plosiver	p, b, t, d, k, g
Frikativer	f, v, s, Sh, Ch, j, h
Sonoranter	M, n, Ng, l, r
Viktige allofoner	eh, aei, oey, oui, Ai, aeu, ui, Oy, rt, rd, rn, rl, rL_
Andre symboler	Sil (stillhet), sp (kort pause)

Tabell 1: Symboler i det fonetiske alfabetet ntnu_no med HTK/Festival ”safe” symboler.

5.5.2 Kontekstavhengige merkefiler

HTS bruker såkalte kontekstavhengige merkefiler i trening av HMMene og i selve syntesen. Disse merkefilene inneholder informasjon om konteksten til hvert fonem i tillegg til transkripsjon. Merkefilene har en struktur som vist under i det norske språket:

Fonem – Stavelse – Ord – Frase – Ytring.

Det har vist seg i både talegjenkjenning og talesyntese at resultatet vil bli bedre om hver modell lagrer informasjon om kontekstuelle sammenhenger til fonemet. Som beskrevet i avsnitt 3.3.2 er det viktig å gruppere disse kontekstuelle faktorene på en slik måte at treningsdataene blir utnyttet mest mulig effektivt. I masteroppgaven ble det tatt med følgende kontekstuelle faktorer:

- Fonem
  - navn på {forrige foregående, foregående, aktuelle, etterfølgende, neste etterfølgende} fonem
  - posisjon til aktuell fonem i aktuelle stavelse (fremre og bakre)

- Stavelse

- Trykk eller ikke trykk på {forgående, aktuell, etterfølgende} stavelse
- Aksentuert eller ikke aksentuert {foregående, aktuell, etterfølgende}stavelse
- Antall fonem i {foregående, aktuell, etterfølgende}stavelse
- Posisjon av aktuell stavelse {fremre, bakre} i aktuelt ord
- Posisjon av aktuell stavelse {fremre, bakre}i aktuell frase
- Antall trykkbelagte stavelser {før, etter} aktuell stavelse i aktuell frase
- Antall aksentuerte stavelser {før, etter} aktuell stavelse i aktuell frase
- Antall stavelser fra {forrige trykkbelagte, forrige aksentuerte} stavelse til aktuell stavelse
- Antall stavelser fra aktuell stavelse til {neste trykkbelagte, neste aksentuerte} stavelse
- Navn på vokal av aktuell stavelse
- Antall stavelser i {forrige, aktuell, neste }frase
- Antall stavelser i en ytring
- Toneme (Tonelag) på {foregående, aktuell, etterfølgende} stavelse (gjelder kun første stavelse i foten)

- Ord
  - Ordklasse til {forrige, aktuelt, neste}ord
  - Antall stavelser i {forrige, aktuelt, neste}ord
  - Posisjon av aktuelt ord i aktuell frase {fremre, bakre}
  - Antall innholds ord {før, etter} aktuelt ord i aktuell frase
  - Antall ord fra forrige innholdsord til aktuelt ord
  - Antall ord fra aktuelt ord til neste innholdsord
  - Antall ord i {forrige, aktuelle, neste}frase
  - Antall ord i en ytring

- Frase
  - Posisjon av aktuell frase i ytring {fremre, bakre}
  - Antall fraser i en ytring

Fonemene ble klassifisert ved ulike særtrekk som:

- Kort eller lang pause
- Stemt, kontinuerlig, ikke kontinuerlig
- Vokal: foregående, midten, etterfølgende, høy, middels, ikke-rund, lukket, halv-åpen, åpen, redusert, oral, nasal
- Halv vokal: oral, nasal
- Konsonant: stopp, frikativ

Det er i denne masteroppgaven tatt med en karakteristisk egenskap ved intonasjonen i det norske språket, forskjellen mellom to tonale betoninger eller såkalte tonemer. Definisjonen på tonem er: en type fonem i et språk som varierer tonen for å fremkalle ulike forskjeller i den leksikalske betydningen [47].
Uansett hvilket språk vi snakker, varierer vi tonehøyden når vi snakker. Dette kaller vi *setningsmelodi*, eller *intonasjon*, og vi bruker denne variasjonsmuligheten systematisk, blant annet for å formidle hva vi vil den vi snakker med spesielt skal legge merke til, til å

formidle følelser, og i mange språk også for å skille mellom spørsmål og vanlige utsagn [47].
Mange språk har i tillegg det vi kan kalle *ordmelodier*, eller tonelag, der variasjon i tonehøyden brukes til å skille mellom ord med ulik betydning. Blant europeiske språk finnes det bare noen få der forskjellige ordmelodier alene kan fortelle oss hvilket ord vi hører. Norsk og svensk hører til i denne lille gruppen, sammen med for eksempel serbisk, kroatisk, litauisk og noen få tyske og nederlandske dialekter. I andre verdensdeler, for eksempel i Øst-Asia og Afrika sør for Sahara, er dette fenomenet svært alminnelig.
På norsk kalles dette fenomenet for *tonelag*. Det finnes to tonelag, dvs. ordmelodier, i norsk som vi vanligvis kaller *tonelag 1* og *tonelag 2*. Det som karakteriserer det vi kan kalle et *tonelagspar*, er at språklydene i de to ordene er de samme, slik at betydningsforskjellen helt og holdent er knyttet til det at de har ulike melodier [9]. Et eksempel på dette er ordet *badet* som vil ha betydningen *"å bade"* (verb) og *badet* (subjektiv).

5.6 Tekst-til-tale system med blandet eksitasjon

I denne masteroppgaven ble analyse/syntese systemet STRAIGHT brukt til å trekke ut taleparametere som modelleres av HMMer i HTS-systemet. STRAIGHT består av tre hovedkomponenter:

- ekstrahering av f_0 og aperiodisk analyse
- spektralanalyse
- syntetisering av tale

5.6.1 Egenskapsvektoren

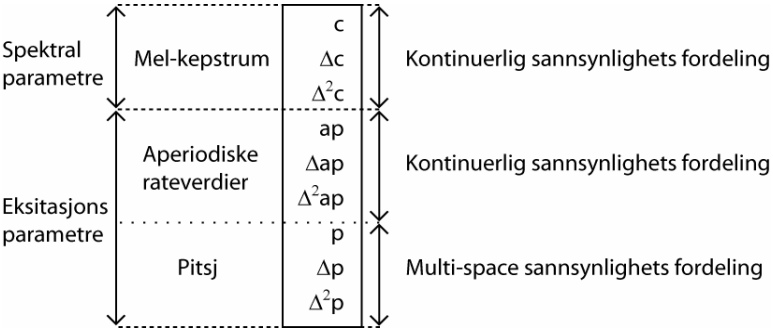
Fundamentalfrekvens, aperiodiske parametere og spektralparametere trekkes ut fra taledatabasen ved bruk av STRAIGHT. Parametrene er representert som en vektor og matriser av reelle tall. Denne enkle representasjonen gjør manipulasjon av parametrene enkelt og direkte. Disse parametrene blir så satt sammen til en egenskapsvektor som HTS-systemet kan lese inn og lage HMMmodeller av. Parametrene fra STRAIGHT har standard rammeskiftsperiode på 1 ms, mens HTS-systemet bruker 5 ms. For at parametersekvensen generert fra STRAIGHT skal være kompatibel med HTS, må rammeskiftsperioden endres til 5 ms.

Gangen i hvordan egenskapsvektoren blir til:

- STRAIGHT ekstraherer fundamentalfrekvens og aperiodiske parametere fra taledatabasen.
- Så blir talespektrumet konstruert ved å bruke informasjon om fundamentalfrekvensen som ble laget i forrige trinn. STRAIGHT utfører en pitsj-adaptiv spektralanalyse i kombinasjon med en overflate-rekonstruksjonsmetode i tid og frekvens for å fjerne signalperiodisitet.

- Deretter blir 40 Mel-kepstral parametere, inkludert den 0te koeffisienten, ekstrahert fra det logaritmiske STRAIGHT-spektrumet. Hvordan Mel-kepstral koeffisienter lages er forklart i appendiks A.3.
- Gjennomsnittet av de aperiodiske parametrene beregnes over fem frekvensbånd.
- Parametersekvensen konverteres så fra 1 ms til 5 ms.
- f_0 konverteres til logaritmisk f_0 (log f_0).
- Dynamiske koeffisienter legges til spektrum, f_0 og aperiodiske parametere.
- En sekvens med egenskapsvektorer blir konstruert med en HTK header slik at HTS-systemet kan lese inn sekvensen med egenskapsvektorer for trening av HMM modeller.

Figuren under viser strukturen til egenskapsvektoren.



Figur 15: Strukturen til egenskapsvektoren med parametere fra STRAIGHT.

Funksjoner og verktøy som er brukt til å ekstrahere taleparametere fra STRAIGHT, er beskrevet i appendiks A.5.

5.7 Trening av taledatabasen

HTS-systemet ble trent på 200 setninger lest av en kvinne. Opptakene var lagret med16 kHz punktprøvningsrate og 16 bits oppløsning.

Treningsdelen består av tre deler:

1. **Taleparametere**
   STRAIGHT systemet brukes for å ekstrahere fundamentalfrekvens og aperiodiske parametere fra taledatabasen. HTS bruker Mel-kepstrum parametere som representasjon av det spektrale spekteret. STRAIGHT-systemet konstruerer et glattet pitsj-adaptiv spektrum. Fra spektrumet ekstraheres en sekvens med 40 mcep parametere.

2. **Kontekstuelle merkelappfiler**
   Ytringsinformasjon fra treningsdatabasen blir konvertert til merkelappfiler som inneholder informasjon om språklige sammenhenger. Disse brukes i modelleringen av spektrum, f_0, aperiodiske komponenter og varighet.

3. **HMMmodellene**
   HMMmodellene modellerer pitsj, aperiodiske parametere, mcep parametere og varighet samtidig. Hver modell inneholder tilstandsvarighetstettheter og observasjonsvektorer som består av tre strømmer:
   - Mel-kepstrum koeffisienter og deres tilhørende Δ -og ΔΔ-koeffisienter
   - Aperiodiske parametere (ap) gjennomsnittelig fordelt over 5 frekvensbånd og deres tilhørende Δ -og ΔΔ-koeffisienter.
   - Logaritmen til fundamentalfrekvensen (f_0) og dens Δ -og ΔΔ-koeffisienter.

For hver av ytringene som brukes i treningsprosessen er det nødvendig med to filer som inneholder informasjon om:

- Start og stopp tid for hvert fonem i ytringen.
- Kontekstuelle faktorer i ytringen, se avsnitt 5.5.2. Fra disse faktorene dannes desisjonstrærne.

5.7.1 Trening av HMM modellene

Trening av modellene skjer ved hjelp av programpakken HTK og et treningsskript kjører HTK-kommandoene for HTS. For nærmere beskrivelse av treningsskript se appendiks A.6.

Det ble brukt en venstre-til-høyre HMM modell med 5 tilstander. Fra hver tilstand genereres det en sannsynlighetsfordeling som består av 5 strømmer:

- en strøm med Mel-kepstral koeffisienter (spektrum) med tilhørende Δ- og ΔΔ-koeffisienter
- en med aperiodiske verdier med tilhørende Δ- og ΔΔ-koeffisienter.
- en strøm med log f_0 parametere
- en strøm for Δ log f_0
- en strøm for ΔΔ log f_0

Spektrum og aperiodiske verdier modelleres med enkle gaussiske fordelinger med diagonal kovariansmatrise. Log f_0, Δ log f_0 og ΔΔ log f_0 strømmene modelleres med Multi-Space probability distributions (MSD) bestående av enkle gaussiske fordelinger med diagonal kovariansmatrise (stemt rom) og enkle diskrete fordelinger som genererer et symbol (ustemt rom). Hver tilstandsvarighets-sannsynlighetsfordeling modelleres med enkle gaussiske fordelinger, med dimensjon lik antall tilstander i HMMmodellen.

Under kjøringen av treningsskriptet lages fire forskjellige desisjonstrær, et for mcep, f_0, aperiodiske verdier og varighet. Dette ved hjelp fra det modifiserte HTK-verktøyet HHEd. HHEd er et desisjonstrebasert klyngingsverktøy [43]. Modifiseringen består i at det brukes et MDL-kriterium for å optimere beslutningstrærne istedenfor et ML-kriterium som normalt brukes i HTK.

Hva som utføres under treningen:

- Definisjon av en prototyp HMM og treningsdata leses inn.
- Beregning av global varians (flat start) for å få en optimal skala for parametrene for varians.
- Initialisering av kontekstuavhengige HMMer. K-means algoritmen trener modellene og de verdiene en oppnår her blir startverdiene i neste oppgave.
- Parameterverdiene blir re-estimert av Baum Welch-algoritmen. Den finner sannsynligheten for å være i en tilstand gitt en tid ved bruk av Forover-bakover algoritmen. Denne sannsynligheten blir så brukt til å forme et vektet gjennomsnitt for HMM parametrene.
- Kontekstuavhengige HMMer kopieres til kontekstavhengige HMMer.
- Innebygd re-estimering (embedded re-estimation) av kontekstavhengige HMMer. Baum Welch-algoritmen brukes også her.
- Strømavhengig klynging basert på desisjonstrær utføres for spektrum. Klyngingen foregår ved å velge ut passende særegne egenskaper fra listen med spørsmål og HMMene grupperes deretter. Optimeringen av desisjonstreet skjer ved bruk av MDL-kriteriet.
- På samme måte klynges modellene for aperiodiske verdier og deretter for f_0.
- Innebygd re-estimering av klyngede HMMer, der noen av tilstandene deler parametere med andre modeller med samme tilstand. Dette for å minske antall tilstander som må lagres. Ved siste iterasjon genereres modeller for tilstandsvarighet.
- Desisjonstrebasert klynging av varighet.
- Macromodell-filene (MMF) som blir brukt av HTK konverteres til det binære formatet som hts-engine benytter.
- Det genereres usette kontekstavhengige HMMer fra desisjonstrærne. Hts_engine traverereserer desisjonstrærne og finner de korresponderende løvnodene. Ved å bruke statistikk på de nodene som er funnet, genereres mcep, ap og f0 sekvenser og syntetiserer en bølgeform.

5.8 Syntese av en norsk HTS-stemme

Gangen i syntetiseringen er som følger:

- Forprosessering utføres på den ønskede ytringen (teksten som er skrevet inn til systemet) sammen med den kontekstuelle informasjonen beskrevet i avsnitt 5.5.2. Denne transkripsjonen genereres av et skript som bruker det norske TTS-systemet Talsmann for å omforme tekst til en fonetisk transkripsjon.
- Den kontekstavhengige HMMen som best beskriver det valgte segmentet og den aktuelle konteksten velges. Hvis denne modellen ikke eksisterer blant treningsmodellene, skapes en modell av parameterverdiene fra nærliggende modeller i desisjonstreet.
- Fra de valgte og genererte modellene gjenskapes mcep og f_0. Disse parametrene brukes til å styre en enkel signalkilde og et MLSA-filter [13] som generer syntetisk tale.

Da tiden ikke strakk til, fikk en dessverre ikke syntetisert tale fra parametrene ekstrahert fra STRAIGHT. Disse parametrene er blitt trent av HTS-systemet. Parametrene ligger til rette for å skape en syntetisk stemme ved å bruke STRAIGHT-systemet. STRAIGHT krever f_0, aperiodiske verdier og effekt-spektrum for å generere tale. Det som gjenstår er å konvertere Mel-kepstrum til effekt-spektrum, siden det er det STRAIGHT trenger for å lage tale. Deretter kan funksjonen i STRAIGHT som genererer tale brukes. Denne er kort beskrevet i appendiks A.5.

6 Evaluering av naturlighet i norsk HTS-stemme

6.1 Kvalitet

Hvor bra en kunstig stemme oppfattes bestemmes av flere faktorer. Det er særlig to egenskaper som må tas i betraktning; grad av tydelighet og naturlighet. Disse egenskapene beskrives nærmere her:

- **Tydelighet**
  Hvor tydelig en kunstig stemme oppfattes henger sammen med hvor stor del av de syntetiserte ordene og setningen som er klare og forståelige for lytteren. En syntetisert stemme kan ha en robotaktig klang, men vil likevel være forståelig og tydelig selv om det høres godt at den er syntetisk.

- **Naturlighet**
  Med naturlighet menes i hvor stor grad den syntetiske stemmen har en setningsmelodi som tilsvarer naturlig tale fra et menneske. Naturlighet henger ofte sammen med prosodi, det vil si om den syntetiske stemmen inneholder pauser, endrer tonefall, volum, stemmekvalitet og hastighet når den snakker. Tonefall er den faktoren som er den mest uttrykksfulle av de ulike prosodiske fenomenene. Mennesket endrer systematisk fundamentalfrekvensen for å uttrykke følelser når det snakker, eller for å vekke oppmerksomheten til lytteren.

I et tekst-til-tale system er det ikke bare de akustiske karakteristikkene som er viktige, men også forprosesseringen av tekst og lingvistisk realisering bestemmer den endelige talekvaliteten. I evaluering av tale er det vanlig å fokusere på konsonanter, siden de er vanskeligere å syntetisere godt i forhold til vokaler. Spesielt nasaler (/m/ /n/ /ng/) er problematiske.
Disse egenskapene ble brukt til å vurdere kvaliteten til den norske stemmen og resultatet er gitt i neste underkapittel.

6.2 Resultat og vurderinger

I denne masteroppgaven har hensikten vært å forbedre naturligheten til en allerede eksisterende norsk HTS-stemme. Den kontekstuelle faktoren *tonelag*, en karakteristisk egenskap ved det norske språket, ble lagt til basisversjonen. Det var av interesse å se om denne egenskapen ville forbedre naturligheten i den syntetiske stemmen.

Da tiden ikke strakk til, er det ikke utført en formell lyttetest for vurdering av stemmene oppnådd i oppgaven. Kvalitet og naturlighet på stemmene er blitt vurdert subjektivt av forfatteren, og grunnet dette vil det ikke bli trukket noen bastante konklusjoner.

De tre stemmene 'basisversjon av norsk HTS-stemme', 'basisversjon med toneme' og 'basisversjon trent med f0-kurver fra STRAIGHT og toneme', vil for lesbarheten heretter refereres til som B, BT og BTSf0.

B, BT og BTSf0 er konfigurert og testet i Festival og de er trent fra en database med 2092 setninger.

Tabellen under gir en oppsummering av vurderingene:

	(B) Basisversjon, norsk HTS-stemme	(BT) Basisversjon, norsk HTS-stemme med tonelag	(BTSf0) Basisversjon, norsk HTS-stemme med f0 generert fra STRAIGHTog med tonelag
Tydelighet	Forståelig	Forståelig	Godt forståelig
Naturlighet	Lite setningsmelodi	Mer melodi sammenliknet med (B)	Mindre setningsmelodi enn i (BT)
Stemmekvalitet	Robotaktig klang	Fortsatt robotaktig klang	Fortsatt robotaktig klang. Støy ved "t"ene.
Tonefall	Unaturlig rytme	Mer naturlig rytme, men kan oppfattes som mer "hakkete" enn (B)	Den med mest naturlig rytme sammenlignet med de to andre stemmene
Pauser	Passende pauser ved komma	Greie pauser	Greie pauser
Hastighet	Passe rytme	Litt rask	Passe tempo

Tabell 2: Oversikt over resultatene fra vurdering av naturligheten i stemmene

BTSf0 virker mer naturlig og robust i forhold til de to andre stemmene. Denne stemmen har en god setningsmelodi og det flyter forholdsvis jevnt uten vesentlige diskontinuiteter når stemmen snakker. BT har en helt grei setningsmelodi. I B er setningsmelodien ikke god på lange setninger, og den har enkelte unaturlige tonefall. I både B og BT høres det ut som om det er diskontinuitet i opplesningen.

Både BTSf0 og BT legger trykket mer riktig og gjør mer tydelige pauser ved komma sett i forhold til B.

Vesentlige forskjeller i stemmene høres når de leser opp alfabetet og tallrekker. BTSf0 leser de fleste bokstaver og tall korrekt, mens B og BT sleper på en del lyder. Denne forskjellen mellom BTSf0 og B/BT er også markant ved opplesning av ukedager og måneder. Se for øvrig appendiks A.6 med lydfiler for eksempler.

I alle tre stemmene minner lydkvaliteten om lyden i en dårlig telefon, lyden oppfattes som uklar. Mer presist virker det som om det er susing eller støy i lydbildet. Kjente rekker, som tall og ukedager oppfattes lett, mens ukjente setninger er vanskeligere å oppfatte. Dette kommer antakelig av den enkle signalkilden som brukes. Forståeligheten er best i BTSf0.

BTSf0 og BT sammenliknet med B antyder følgende:

- Naturligheten med hensyn på setningsmelodi er forbedret i BT og BTSf0
- Forståeligheten er ikke i vesentlig grad bedret i BT, men er noe bedret i BTSf0
- Uttalen i BTSf0 er vesentlig bedret

BTSf0-stemmen ble vurdert til å være den beste stemmen av de tre stemmene vurdert i denne masteroppgaven. Denne stemmen er trent med kontinuerlige f_0-kurver ekstrahert fra STRAIGHT. Ut fra dette ser en at kontinuerlige og robuste f_0-kurver er viktig for å oppnå god naturlighet i talen.

Kontekstuelle faktorer har en påvirkning på setningsmelodi og prosodiske faktorer i språket vårt. Dette høres godt ved å sammenligne B-stemmen med BT-stemmen. Setningsmelodien er klart bedre i BT-stemmen etter at tonelag er lagt til her.

Alle tre stemmene har fortsatt en robotaktig klang. Det er blitt forsøkt å fjerne denne klangen ved å bruke STRAIGHT-modellen. Dessverre strakk ikke tiden til å få syntetisert HTS-stemmen, slik at naturligheten i denne stemmen kunne vurderes.

6.3 Feilkilder

Vurderingen av stemmene er gjort subjektivt av forfatteren av denne rapporten. Det er en feilkilde i seg selv, siden den bare viser et menneskets personlige oppfatningen av naturlighet i stemmene. Hvis det hadde blitt utført en formell test ville kanskje stemmene blitt vurdert annerledes.

7 Konklusjon og videre arbeid

7.1 Konklusjon

I denne masteroppgaven er det blitt forsøkt å forbedre naturligheten til en norsk syntetisk stemme basert på HMMmodeller. Det er blitt tatt utgangspunkt i en norsk, basisversjon av HMM-basert talesyntese (HTS), programvarebiblioteker for HMM-basert talegjenkjenning (HTK), Speech SignalProcessing Toolkit (SPTK) og det generelle tekst-til-talesystemet Festival.

Det er gitt en teoretisk beskrivelse av et system for HMM-basert talesyntese, hvor HMMer brukes til å generere tale. I treningsdelen blir spektrum og eksitasjonsparametere trukket ut fra taledatabasen og modellert av kontekstavhengige HMMer. Kontekstavhengige HMMer (trifoner) brukes for å få tak i de kontekstuelle faktorene som påvirker spektrum og som definerer de ulike lingvistiske og prosodiske faktorene i et språk. I syntesedelen skjøtes kontekstavhengige HMMer sammen i henhold til den teksten som skal syntetiseres. Deretter blir parametrene generert fra HMMene ved hjelp av en algoritme [17] for generering av taleparametere. Tilslutt brukes parametrene til å styre en enkel signalkilde og et MLSA-filter som syntetiserer tale i henhold til parametersekvensen.

Det er blitt konfigurert to norske HTS-stemmer med forbedret naturlighet i det generelle tekst-til-talesystemet Festival. Festival gjør det mulig å syntetisere en hvilken som helst norsk setning. I BT-stemmen er det lagt til egenskapen tonelag, BTSf0-stemmen er trent med f_0-kurver ekstrahert fra STRAIGHT i tillegg til å ha med egenskapen tonelag. I vurderingen av oppfattet naturlighet i disse to stemmene sammenlignet med den norske basisversjonen, er det helt klart at disse faktorene har forbedret naturligheten i stemmene. Begge stemmene har en bedre setningsmelodi og kvaliteten er jevnere. BTSf0-stemmen med f_0-kurver fra STRAIGHT skiller seg klart ut fra de to andre ved at den er mer robust og behersker lengre setninger, samtidig som den beholder den jevne kvaliteten. En nøyaktig estimering av fundamentalfrekvensen er tydelig av stor betydning for kvaliteten på den syntetiske talen. Ut fra dette er det rimelig å påstå at metoden i STRAIGHT for å trekke ut fundamentalfrekvens, er bedre enn ”get_f0” fra ESPS pakken til Entropic, brukt i basisversjonen.
Kontekstuelle faktorer som er karakteristisk for det norske språket har og en innvirkning på hvor god stemmen blir. BT-stemme med tonelag har en betraktelig bedre setningsmelodi enn basisversjonen av norsk HTS-stemme.

Fremdeles har stemmene en robotaktig klang. Dette kommer antakelig av den enkle kilden som brukes. Denne genererer enten stemte eller ustemte pulser. Eksitasjonsmodellen STRAIGHT er integrert til HTS-systemet for å redusere denne klangen. Dessverre strakk ikke tiden til å syntetisere taleparametrene ekstrahert fra STRAIGHT og trent av HTS-systemet. Det er derfor ikke mulig å si om dette ville hatt noen forbedring på talekvaliteten. Parametrene ligger til rette for å leses inn i STRAIGHT-systemet for syntetisering. Det er blitt vist at en kan trene parametere ekstrahert fra STRAIGHT i et HMM-basert talesyntese system. Det som gjenstår er å syntetisere tale fra de genererte parametrene fra treningen.

Til tross for at det ikke ble tid til å få syntetisert HTS-stemmen i STRAIGHT, har arbeidet vist at ved å trene stemmen med f_0-kurver generert fra STRAIGHT, og ved å legge til tonelag, har en oppnådd noe bedret naturlighet i norsk HTS-stemmen.
Håper dette dokumentet er en god dokumentasjon på det som er gjort, i tillegg til å være en hjelp for andre som ønsker å forbedre naturligheten i en norsk HTS-stemme.

7.2 Videre arbeid

Denne masteroppgaven ble utført våren 2006 ved institutt for elektronikk og telekommunikasjon ved NTNU. Under arbeidet med å forbedre naturligheten til basisversjonen av en norsk HTS-talesyntese er det dukket opp flere ting som kan være interessant å se på videre:

- **Syntetisere HTS-stemme ved bruk av STRAIGHT**
  Tips til videre arbeid vil være å få syntetisert HTS-stemmen i STRAIGHT og se om denne stemmen vil gi noen forbedring i talekvalitet og naturlighet. I HTS-systemet er det blitt generert filer med fundamentalfrekvens, aperiodiske-parametere og Mel-kepstrum. For å lage tale krever STRAIGHT fundamentalfrekvens, aperiodiske parametere og effekt-spektrum. Det vil si at Mel-kepstrum må konverteres til effektspektrum.
- **Definere HTS-stemmen i Festival**
  Annet videre arbeid kan være å definere denne stemmen som en egen stemme i Festival. I denne masteroppgaven er en HTS-stemme med taleparametere fra STRAIGHT blitt trent på 200 setninger. Det kan være av interesse å se om kvaliteten på stemmen bedres ytteligere med flere setninger.
- **Kvalitetsvurdering i forhold til andre norske syntetiserte stemmer**
  Man kan vurdere kvaliteten på en norsk HTS-stemme mot andre norske, kunstige stemmer. Man kan da sammenligne blant annet oppfattet naturlighet og tydelighet hos de ulike stemmene.
- **HTS-systemet som prosodipredikator for datadrevet skjøtesyntese**
  Prosodien generert i HTS-systemet er nokså naturlig. Det kunne vært interessant å se om HTS-systemet kunne fungere som en prosodipredikator for datadrevet skjøtesyntese.

8 Referanser

1. *HMM-Based Speech Synthesis System (HTS)*. 2006 3. mars [cited 2006 29. mai]; Available from: http://hts.ics.nitech.ac.jp/.
2. Tokuda, K., H. Zen, and A.W. Black. *An HMM-based speech synthesis system applied to english*. in *IEEE Speech Synthesis Workshop*. 2002. Santa Monica, California.
3. Tamura, M., et al., *Speaker Adaption for HMM-based Speech Synthesis System using MLLR.* Proc. of The Third ESCA/COCOSDA workshop on Speech Synthesis, 1998: p. 273-276.
4. Yoshimura, T., et al., *Speaker interpolation in HMM-Based Speech Synthesis System.* Proc. of Eurospeech, 1997. **vol. 5**: p. 2523-2526.
5. Mølmen, L., *HMM-basert talesyntese*, in *Institutt for Teleteknikk, Fakultet for Informasjonsteknologi, Matematikk og Elektronikk*. 2005, Norges Teknisk-Naturvitenskapelige Universitet: Trondheim.
6. Kawahara, H., I. Masuda-Katsuse, and A.d. Cheveigné, *Restructuring speech representations using a pitch-adaptiv time-frequency smoothing and an instantaneous-frequency-based F0 extraction.* Speech Communication 27, 1999(3-4): p. 187-207.
7. Stylianou, Y., *Applying the harmonic plus noise model in concatenative speech synthesis.* IEEE Tran. Speech and Audio Processing, 2001. **Vol. 9**(No. 1): p. 21-29.
8. Zen, H. and T. Toda. *An Overview of Nitech HMM-based Speech Synthesis System for Blizzard Challenge 2005*. in *Eurospeech*. September 2005. Lisbon.
9. Almberg, J., *The 'Melody' in Synthetic Speech: What Kind of Phonetic Knowledge Can Be Used to Improve It?* Telektronikk, 2003. **99**(No 2): p. 30-44.
10. Black, A. and K. Lenzo. *Building Synthetic Voices*. 2003 [cited 2006 29. mai]; Tilgjenglig fra: http://www.festvox.org/festvox/festvox_toc.html].
11. Traunmuller, H. *History of Speech Synthesis, 1770-1970*. 2000 [cited 20. mai 2006]; Tilgjenglig fra: http://www.ling.su.se/straff/hartmut/kemplne.htm].
12. Masuko, T., *HMM-based Speech Synthesis and its Applications*, in *Institute of Technology*. 2002: Tokyo.
13. Amdal, I. and T. Svensen, *State-of-the-art for datadrevet skjøtesyntese*. 2004: Institutt for Elektronikk og Telekommunikasjon: NTNU-Norges Teknisk-Naturvitenskapelige Universitet.
14. Huang, X., A. Acero, and H.W. Hon, *Spoken Language Processing, a guide to theory, algorithm and system developement*, ed. s. ed. 2001, Upper Saddle River: Prentice Hall.
15. Silverman, K. *ToBI: A Standard for Labeling English Prosody*. in *Spoken Language Processing*. 1992. Banff, Canada.
16. Heggtveit, P.O., *An Overview of Text-to-Speech Synthesis.* Telektronikk, 2003. **Volume 99 No. 2**: p. 30-44.
17. *Speech Synthesis*. 24.mai 2006 [cited 2006 29. mai]; http://en.wikipedia.org/wiki/Speech_synthesis].
18. Svensen, T., *Speech Technology: Past, Present And Future.* Telektronikk, 2003. **Volum 99 No. 2**: p. 6-18.
19. group, H.w. *The HMM-based Speech Synthesis System (HTS) version 1.1.1*. 2003 [cited; Available from: http://hts.ics.nitech.ac.jp/README.html.

20. Masuko, T., et al. *Speech synthesis from HMMs using dynamic features* in *Proc. of ICASSP*. 1996.
21. Clark, R. and A. Black. *The Festival Speech Synthesis System*. 2004,14 juli [cited 2006 mai 29]; Available from: http://festvox.org/festival/downloads.html.
22. Lundgren, A., *HMM-baserad talesyntes*. 2004, Department of Speech, Music and Hearing Royal Institute of Technology: Stockholm.
23. Yoshimura, T., et al. *Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-based Speech Synthesis*. in *Proc. of Eurospeech*. 1999.
24. Tokuda, T., et al. *Speech parameter generation algorithms for HMM-based speech synthesis*. in *Proc. ICASSP*. 2000. Istanbul, Turkey.
25. Tokuda, K., et al. *Multi-Space Probability Distribution HMM*. in *IEICE TRANS. INF. & SYST.* 2002.
26. J.Odell, J., *The use of context in large vocabulary speech recognition*. 1995, Ph.D dissertation, Cambridge University.
27. Fukada, T., et al. (1992) *An adaptive algorithm for mel-cepstral analysis of speech*. **Volume**, 137-140
28. Young, S.J., J.J. Odell, and P.C. Woodland, *Tree-based State Tying for High Accuracy Acoustic Modeling*, in *Engineering Department*. 1994, Cambridge University.
29. Shinoda, K. and T. Watanabe. *Acoustic Modeling Based on the MDL Principle for Speech Recognition*. in *Proc. of Eurospeech*. 1997.
30. group, H.w. *Speech Signal Processing Toolkit*. 2002 [cited; Available from: http://kt-lab.ics.nitech.ac.jp/~tokuda/SPTK/.
31. Yoshimura, T., et al., *Mixed Excitation for HMM-based Speech Synthesis.* proc. of Eurospeech, 2001: p. 2263-2266.
32. Yegnanarayana, B. and C. d'Alessandro, *An Iterative Algorithm for Decomposition of Speech Signals into Periodic and Aperiodic Components.* IEEE Tran. Speech and Audio Processing, 1998. **vol. 6**(No. 1): p. 1-11.
33. Dudley, H., *Remaking speech.* J. Acoust. Soc. Am, 1939. **11**: p. 169-177.
34. Griffin, D. and J.S. Lim, *Multiband excitation vocoder.* IEEE Tran. Speech and Audio Processing, 1988. **vol. 36**: p. 1223-1235.
35. Childers, D. and H. Hu, *Speech Synthesis by Glottal Excited Linear Prediction.* Journal of the Acoustical Society of America, 1994. **JASA vol.96**(4): p. 2026-2036.
36. Laroche, L., Y. Stylianou, and E. Moulines, *HNS: Speech modification based on a harmonic + noise model.* Proc. IEEE-ICASSP, 1993: p. 550-553.
37. Stylianou, Y. *On the implemention of the harmonic plus noise model for concatenative speech synthesis*. in *ICASSP 2000*. June 2000. Istanbul, Turkey.
38. Kawahara, H., H. Katayose, and A.d. Cheveigné. *Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity*. in *Proc. Eurospeech'99*. 1999.
39. Zolfahari, P., et al. *Glottal Closure Instant Synchronous Sinusoidal Model for High Quality Speech Analysis/Synthesis*. in *Proc. of Eurospeech*. 2003. Geneva, Switzerland.
40. Kawahara, H. *Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for high quality speech analysis, modification and synthesis system STRAIGHT*. in *in Proc. of 2nd MAVEBA*. 2001. Firenze, Italy.
41. Kawahara, H., Y. Atake, and P.Zolfaghari. *Accurate vocal event detection method based on fixed-point analysis of mapping from time to weighted average group delay*. in *Proc. ICSLP'2000*. 2000. Beijin, China.

42. Stylianou, Y. *Concatenative speech synthesis using a harmonics plus noise model*. in *Third ESCA Speech Synthesis Workshop*. 1998. Jenolan, Australia.
43. team, H. *HTK*. [cited 2006 mai 20.]; Available from: http://htk.eng.cam.ac.uk/.
44. *Festival*. [cited 2006 mai 20.]; Available from: http://www.festvox.org.
45. *Talsmann – Norwegian TTS for Windows*. 2003 februar 12 [cited 2006 mai 20.]; Available from: http://www.telenor.no/fou/prosjekter/taletek/talsmann/tts_more.htm.
46. *Specification of the FONEMA reference database*. [cited; Available from: /u/Fonema/FonDat1.
47. *Tone (linguistics)*. mai 24 [cited 2006 mai 20]; Available from: http://en.wikipedia.org/wiki/Tone_%28linguistics%29.

A. Appendiks

A.1 Hvordan bygge en norsk HTS-stemme med taleparametere fra STRAIGHT steg-for-steg.

Her gis det en steg-for-steg oppskrift på hvordan man kan gå frem for å lage en ny, norsk stemme i HTS med taleparametere fra STRAIGHT.

Hva man trenger i utgangspunktet.

Det man trenger for å komme i gang med arbeidet med å syntetisere en norsk HTS-stemme er følgende:

- En norsk taledatabase: denne inneholder treningsdata som HTS-treningsskriptet trenes på.
- HTS trenger programmene Hidden Markov Modell Toolkit (HTK) og Speech Signal Processing Toolkit (SPTK)
    - SPTK: verktøy for prosessering av talesignaler.
      Kan lastes ned fra http://kt-lab.ics.nitech.ac.jp/~tokuda/SPTK/
    - HTK: verktøy for å bygge og manipulere skjulte Markovmodeller.
      Kan lastes ned fra http://htk.eng.cam.ac.uk/
- HTS versjon 1.1.1
    - Inneholder alle skriptene som er nødvendig for trening og syntese
      Kan lastes ned fra: http://hts.ics.nitech.ac.jp/
- Festival: tekst-til-tale system som bygger og syntetiserer stemmer.
  Kan lastes ned fra: http://www.festvox.org
- STRAIGHT analyse/syntese system: send forespørsel til forfatteren, Hideki Kawahara, av systemet om å få den fullstendige versjonen. Se nettside http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTtrial/

Steg 1. Lage en mappestruktur for treningsdata

- Lage en katalog HTS-demo_land_taledatabase_taler hvor datafilene som identifiserer treningsdatabasen skal ligge. Se tabell 2 i appendiks A.2 for oversikt over filer og kataloger som må være tilstede i databasekatalogen. Se på katalogstrukturen til den engelske eksempelstemmen og lag en tilsvarende struktur. Kopier nødvendige konfigurasjonsfiler herfra.

- Lag en katalog kalt HTS-demo. I denne katalogen ligger alt som er vesentlig for selve treningen. Se tabell 3 for fullstendig filoversikt i appendiks A.3.

Steg 2. Nødvendige datafiler
Filene må være formatert slik at de tilsvarer den rette endian på den maskinen eksperimentet kjører fra. Kan være enten LSB eller MSB.

Disse filene identifiserer databasestemmene. Fra disse datafilene genereres treningsdata. $DATASET_$SPEAKER_$UID definerer navnet på taledatabase_taler_ytringsid

- Ytringsfiler, utt-filer (”../utt/$DATASET_$SPEAKER_$UID.utt”). Disse filene brukes til å lage merkelappfiler og er ascii (og Festival) formatert. Ytringsfilene kan lages fra taledatabasen ved å bruke skriptet *make_utts.sh* som ligger i Festival-katalogen under *festival/examples*.
- Raw-filer (bølgeformsfiler) (”../raw/$DATASET_$SPEAKER_$UID.raw”)
  Disse filene er LSB eller MSB, RAW og heltalls-verdier. Brukt samplingsfrekvens er 16 kHz.
- Win-filer (”../win/*.win”)
  Disse filene må være formattert i ENDIAN som tilsvarer den maskinen som brukes. Win-filene som brukes fra ”cmu_artic_awb” eksemplet er ”little-endian”, så hvis maskinen som brukes er ”big-endian”, må disse filene formatteres.

I denne masteroppgaven er det blitt laget treningsdata fra analyse/syntese systemet STRAIGHT. STRAIGHT er implementert i MatLab. Det genereres f0, aperiodiske indekser og mcep parametere fra dette systemet. NB! STRAIGHT har default rammeskift på 1ms, mens HTS-systemet bruker en rammeskifts periode på 5ms. Denne parameteren endres til 5 ms i matlab skriptet *defaultparams.m* som kopieres fra *defaultparamsorg.m* som finnes i STRAIGHT mappen. Hvis *defaultparams.m* filen allerede eksisterer, brukes denne. Den vil da overskrive alle originale standard parametere.

En annen måte å konvertere fra 1 ms til 5 ms, er å bruke verktøyene *ch_track* fra Festival, *x2x* fra SPTK og python skriptet *cat_raw.py*. Konverteringen skjer ved å skrive følgende kommando:
Denne kommandoen leser inn for flere filer.

For fn in ../fil_1ms/.mcep; do cat $fn | python ../cat_raw.py –l lengden | ch_track –itype ascii-s 0.001 –S 0.005 –otype ascii –sm 0.01 –smtype mean - | x2x +af > navn_på_fila_den_skriver_til: done*

Parametrene lages på følgende måte:

- Lag en mappe *STRAIGHT* hvor alle filer tilhørende STRAIGHT-systemet lagres.
- I databasekatalogen hvor alle taledataene lagres opprettes en mappe *straight.*
- Kjør skriptet *run_straight.sh* som lager nødvendige datafiler. I skriptet må det henvises til matlab, SPTK, wav-filer inneholdt i databasekatalogen. Sett mceporder og allpass-konstanten α. Etter at skriptet er kjørt, finnes i mappen *straight*: mcep, f0 og ap-filer. Endre så navnet på mcep og f_0-filene slik at de tilsvarer $DATASET_$SPEAKER_$UID.mcep og $DATASET_$SPEAKER_$UID.f0. Disse filene kopieres så til henholdvis *mcep* og *f0* mappene som opprettes i databasekatalogen.
- For aperiodiske verdiene (ap) må det beregnes gjennomsnittelige verdier over fem frekvensbånd. Dette gjøres ved å kjøre matlab skriptet *ap_fivefreq_band.m*. Opprett en mappe som heter *ap_1*, hvor filene fra straight legges. Sett banen i *ap_fivefreq_band.m* slik at koden leser inn filer fra denne katalogen. Kjør så matlab koden. Når dette er gjort må disse filene endre navn slik at de er på formen $DATASET_$SPEAKER_$UID.ap. Kopier så disse filene til mappen *ap* som opprettes i databasekatalogen.

Steg 3. Lage treningsdata (cmp) fra mcep, ap og f0

- Konfigurere Makefile slik at BYTESWAP settes i henhold til maskinen det kjøres på og MCEPORDER settes i forhold til eksperimentet. I tillegg må det henvises til SPTK og Festival.
- Skriptet *mkdata.pl* genererer treningsdata. Dette skriptet er modifisert slik at det genereres en egenskapsvektor bestående av mcep, ap og f_0. Endringer som er blitt gjort i dette skriptet er blitt merket i et word dokument kalt *modifisert_mkdata.doc.* Dette dokumentet ligger i mappen */HTS_STRAIGHT_1/HTS-demo/FonDat1_JK/scripts.*

- Skriv ”make” inne i databasekatalogen og treningsdata blir generert. I tillegg genereres merkefiler som brukes i klyngingen av HMMer under treningen.

Trene HMMer og bygge desisjonstrær

- Gå inn i mappen HTS-demo
- Questions-filen må forandres slik at den er tilpasset det norske språket. Denne filen inneholder spørsmål relatert til desisjonstrærne. I denne filen er det nå lagt til spørsmål relatert til tonelag.
- Konfigurer Makefila som ligger i denne mappen for HTS, ved å henvise til treningsdata og til HTK.
- Skriv ”make” inne i katalogen HTS-demo, da blir treningsskriptet generert.
- I treningsskriptet må det endres henvisning til et verktøy kalt HMGenS_dyre i stedetfor HMGenS som det henvises til i basisversjonen. HMGenS_dyre er et modifisert verktøy av HMGenS tilpasset for å ta inn aperiodiske parametere. Kjør treningsskriptet Training.pl ved å skrive *perl Training.pl* i kommandovinduet. Dette treningsskriptet er en modifisert utgave tilpasset for å ta inn parametere fra STRAIGHT. Se word dokumentet *modifisert_Training_ntnu_fondat1_jk.doc* for endringene som er gjort. Endringene er markert med farge og denne filen er å finne i */HTS-demo/script* i appendiks A.5.

Når treningsskriptet kjøres bygges det separate desisjonstrær for parametrer for spektrum, f0, aperiodiske parametere og varighet ved hjelp av den modifiserte utgaven av HTK-verktøyet HHEd. Helt på slutten genereres parametersekvenser for mcep, f0, ap, varighet og pitch som tilsvarer testsetninger inneholdt i taledatabasen. Disse brukes i STRAIGHT for å syntetiseres stemme.

A.2 Oversikt over filene i HTS-demo_universitet_land_taledatabase_taler

Fil / Katalog	Innhold / Oppgave
../	
Hlist.conf	Konfigurasjonsfil.
lab_format.pdf	Et eksempel på kontekstavhengig labelformat for HTS.
Makefile	Bygger ulike filer for en demonstrasjon av HTS; blant annet treningsdata og labelfiler.
COPYING	
README	
../f0/	
universitet_land_taledatabase_taler_xxxxxxx.fo	Inneholder pitsj-konturer ekstrahert fra taledatabasen av STRAIGHT.
../**mcep**/	
universitet_land_taledatabase_taler_xxxxxxx.mcep	Inneholder 0te til 40te Mel-kepstral koeffisienter ekstrahert fra pitsj-adaptiv STRAIGHT spektrogram.
../**ap**/	
universitet_land_taledatabase_taler_xxxxxxx.ap	Inneholder aperiodiske verdier, gjennomsnittelig fordelt over 5 frekvensbånd fra STRAIGHT.
../labels/	
full.mlf	Genererer / sender filer til ./fullcontext.fem
mono.mlf	Genererer / sender filer til ./monophone.fem
../labels/fullcontext/gen/	Kontekstavhengige labelfiler med format som vist i lab_format-pdf.lab.
../raw/	
universitet_land_taledatabase_taler_xxxxxxx.raw	Ubearbeidet tale, samplet ved 16kHz.
../scripts/	
Delta.pl	Laster regresjonsvindu og legger til delta-koeffisienter. win-filer er inndata.
extra_feats.scm	Inneholder tilleggsegenskaper for demonstrasjon av HTS. Brukes av Festival for egenskapsuttrekking.
Freq2lfreq.pl	Konverterer f0-filer til logf0-filer.
mkdata.in	Modifisert versjon av den som finnes i HTS-demo_CMU_ARTIC_AWB.tar.gz. Lager treningsdata for HTS, genererer mkdata.pl. Har f0-filer og raw-filer som inndata. Lager

	egenskapsvektor av f_0, ap og mcep, som brukes under trening av HMMer.
utt2lab.in	Konverterer Festivals ytringsfiler til kontekstavhengige labelfiler og kontekstuavhengige segment labelfiler for HTS.
../utts/	
universitet_land_taledatabase_taler_xxxxxxx.utt	Ytringsfiler.
../questions/	
questions_qst001.hed	Spørsmålsfil for kontekstklynging basert på desisjonstre.
../win/	
lf0_dyn.win lf0_acc.win mcep_dyn.win mcep_acc.win ap_dyn.win ap_acc.win	Disse filene inneholder regresjonsvindu for beregning av dynamiske egenskaper for henholdsvis f0, mcep og ap.
../test/	Denne mappen inneholder skriptene *cat_raw.py* og *import_f0_straight_clip.* Python skriptet skriver ut binærdata til ascii, og *import_f0_straight_clip* omformer negative verdier til null. Dette er verktøy brukt for å endre på parametrene fra STRAIGHT.

Tabell 3: Filoversikt i HTS-demo_universitet_land_taledatabase_taler.

A.3 Oversikt over de ulike filene i HTS-demo

Fil / Katalog	**Innhold / Oppgave**
../	
README	
Makefile	Bygger ulike filer for en demonstrasjon av HTS, blant annet treningsskriptet.
../configs/	Ulike konfigurasjonsfiler
../edfiles/	Redigerer filer for HTK-kommandoen HHed.
../gen/	Genererer parametere og syntetisert tale.
../hmms/	Trente HMMer.
../logs/	Log-filer.
../proto/	Prototype HMMer.
../scripts/	Ulike skripts som benyttes i arbeidet med HTS-syntesen.
../stats/	Tilstandsfiler

Tabell 4: Filoversikt i HTS-demo

A.4 Mel-kepstral koeffisienter

I Mel-kepstral analysen [27], modelleres overføringsfunksjonen til vokaltrakten $H(z)$ av M'te ordens Mel-kepstral koeffisienter $c = [c(0), c(1), ..., c(M)]^T$ som følger:

$$H(z) = \exp c^T \tilde{z} \tag{A.3.1}$$

$$= \exp \sum_{m=0}^{M} c(m) \tilde{z}^{-m} \tag{A.3.2}$$

hvor $\tilde{z}^{-1} = [1, \tilde{z}^{-1}, ..., \tilde{z}^{-M}]^T$. Systemet $\tilde{z}^{-1}$ er definert som første ordens allpass funksjon:

$$z^{-m} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \qquad |\alpha| < 1 \tag{A.3.3}$$

og den fordreide frekvensskalaen $\beta(w)$ er gitt som dens faserespons:

$$\beta(w) = \tan^{-1} \frac{(1 - \alpha^2) \sin w}{(1 + \alpha^2) \cos w - 2\alpha}, \tag{A.3.4}$$

Faseresponsen $\beta(w)$ gir en god tilnærming til hørselsfrekvensskalaen med et passende valg av α. For samplingsfrekvens lik 16 kHz gir α = 0.42 en god tilnærming til hørselsfrekvensskalaen.

Spektralkriteriet
I den forventningsrettede estimeringsmetoden av log-spektrumet (UELS), er det blitt vist at estimatet av effektspektrumet $|H(e^{jw})|^2$, som er forventingsrettet i forhold til relativ effekt, oppnås på en slik måte at følgende kriterium E minimeres:

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} \{\exp R(w) - R(w) - 1\} dw \tag{A.3.5}$$

hvor

$$R(w) = \log I_N(w) - \log |H(e^{jw})|^2 \tag{A.3.6}$$

og $I_N(w)$ er det modifiserte periodogrammet til en svak stasjonærprosess $x(n)$ gitt av:

$$I_N(w) = \frac{\left|\sum_{n=0}^{N-1} w(n)x(n)e^{-jwn}\right|^2}{\sum_{n=0}^{N-1} w^2(n)} \qquad \text{(A.3.7)}$$

hvor $w(n)$ er vinduet med lengde N. En kan legge merke til at kriteriet i likning (A.3.5) har den samme formen som maksimum-likelihood estimatet til en normal, stasjonær AR-prosess [12].

Siden kriteriet i likning (A.3.5) er utledet uten noen antakelse om en spesifikk spektralmodell, kan den brukes i spektralmodellen i likning (A.3.2). Hvis en tar gevinstfaktoren K utenfor $H(z)$ i likning (A.3.2) får en:

$$H(z) = K \cdot D(z) \qquad \text{(A.3.8)}$$

hvor

$$K = \exp \alpha^T c \qquad \text{(A.3.9)}$$

$$= \exp \sum_{m=0}^{M} (-\alpha)^m c(m) \qquad \text{(A.3.10)}$$

$$D(z) = \exp c_1^T \tilde{z} \qquad \text{(A.3.11)}$$

$$= \exp \sum_{m=1} c_1(m) \tilde{z}^{-m} \qquad \text{(A.3.12)}$$

og

$$\alpha = \left[1, (-\alpha), (-\alpha)^2, ..., (-\alpha)^M\right]^T \qquad \text{(A.3.13)}$$

$$c_1 = \left[c_1(0), c_1(1), ..., c_1(M)\right]^T \qquad \text{(A.3.14)}$$

Forholdet mellom koeffisienten **c** og $\mathbf{c_1}$ er gitt ved:

$$c_1(m) = \begin{cases} c(0) - \alpha^T c, m = 0 \\ c(m), 1 \le m \le M \end{cases} \qquad \text{(A.3.15)}$$

Hvis systemet $H(z)$ betraktes å være et talesyntesefilter, må $D(z)$ være stabilt. Dersom en antar at $D(z)$ er et minimum fasesystem, har en forholdet:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \log\left|H(e^{jw})\right|^2 dw = \log K^2. \qquad \text{(A.3.16)}$$

Ved å bruke likningen over, blir spektralkriteriet i likning (A.3.5):

$$E = \varepsilon / K^2 - \frac{1}{2\pi}\int_{-\pi}^{\pi} \log I_N(w)dw + \log K^2 - 1 \qquad \text{(A.3.17)}$$

hvor

$$\varepsilon = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{I_N(w)}{\left|D(e^{jw})\right|^2} dw. \qquad \text{(A.3.18)}$$

Minimeringen av E med hensyn på c som fører til minimeringen av ε med hensyn på c_1 og minimeringen av E med hensyn på K. Ved å ta den deriverte av E med hensyn på K og sette resultatet lik null, oppnås K som følger:

$$K = \sqrt{\varepsilon_{\min}} \qquad \text{(A.3.19)}$$

hvor $\varepsilon_{\min}$ er den minste verdien av ε.
Det eksisterer bare et minimumspunkt siden E-kriteriet er konvekst med hensyn på c. Av dette følger at minimeringsproblemet av E kan bli løst effektivt med en iterativ algoritme basert på Fast Fourier transform (FFT) og rekursive formler. I tillegg er stabiliteten til løsningsmodellen $H(z)$ alltid garantert [12].

Så kort oppsummert:
I Mel-kepstral analysemetoden, representeres spektrumet med M'te ordens Mel-kepstral koeffisienter og kriteriet som brukes i forventningsrettet estimering av logspekteret er minimert med hensyn på Mel-kepstral koeffisientene. Minimeringsproblemet løses effektivt av en iterativ teknikk ved å bruke FFT, rekursive formler, og en rask algoritme som krever $O(M^2)$ aritmetiske operasjoner. Konvergensen er kvadratisk og det holder med få iterasjoner for å komme til løsningen.

A.5 Benyttede STRAIGHT-komponenter

Merk at STRAIGHT behøver en installert utgave av MatLab. I denne masteroppgaven er versjon 7 av MatLab benyttet.

Fil	Innhold / Oppgave
exstraightsource.m	Trekker ut fundamentalfrekvens og aperiodiske indekser i hvert frekvensbånd fra talesignalet.
exstraightspec.m	Tar inn f_0 og talesignalet. Ut fra dette genereres et glattet f_0-daptivt spektrogram.
exstraightsynth.m	Tar inn f_0, spektrogram og aperiodiske indekser. Bruker disse til å generere syntetisk tale.

Tabell 5: Skript inkludert i STRAIGHT

ap_fivefreq_band.m	Leser inn matrise bestående av aperiodiske verdier med rammeskift 1ms. Tar gjennomsnittsverdien av de aperiodiske verdiene over fem frekvensbånd og skriver til fil. Lineær skala.
run_straight.sh	Kaller lene_make_straightparam.m for hver fil i taledatabasen. Kaller det modifiserte SPTK verktøyet mcep_dyre.
lene_make_straightparam.m	Kaller extraightsource.m og exstraightspec.m. Skriver de genererte parametrene til fil (ap, spektrogram og f_0).
mcep_dyre	Genererer Mel-kepstral koeffisienter ut fra spektrogrammet generert i lene_make_straightparam.m. Koeffisientene skrives til fil. Samplingsfrekvens er lik 16 kHz, sett allpass konstanten α lik 0,42.

Tabell 6: Ekstra skript laget for STRAIGHT

A.6 Implementasjon

Det er lagt ved cd med implementasjonen og lydfiler av de ulike stemmene som er oppnådd i oppgaven.