NTNU
Det skapende universitet

Utvidet taleaktivitetsdeteksjon

Henrik Lødrup Parnemann

Elektronisk systemdesign og innovasjon
Innlevert: juni 2014
Hovedveileder: Torbjørn Svendsen, IET
Medveileder: Øystein Birkenes, Cisco Systems Norway AS

Norges teknisk-naturvitenskapelige universitet
Institutt for elektronikk og telekommunikasjon

Sammendrag

I denne rapporten er det undersøkt og utviklet et system for deteksjon av taleaktivitet i et vilkårlig lydsignal. Dette er utvidet med mulighet for å også detektere korte, brå, energirike lyder, såkalte *transientlyder*. Transientdeteksjon er ønsket brukt i lydnivåutjevnings-system for å kunne redusere uønsket langvarig dempning av tale som umiddelbart etterfølger transientlyder. I denne perioden med stor dempning reduseres talenivået såpass mye at innholdet kan oppleves som uhørlig.

Systemet som er implementert er basert på rammevis, glattet, logaritmisk delbånds-energi som deteksjonsegenskap [1]. Deteksjonen gjøres individuelt per delbånd, og avgjøres for rammen basert på et simpelt, vektet flertall. En statistisk energifordelingsmodell ligger til grunn, der signalet deles i to modellkomponenter: støy/ikke-tale og tale. Det antaes at inngangssignalet til en hver tid består av en støykomponent, samt en tidvis talekomponent. Energifordelingen for hver av disse to lydklassene modelleres som normalfordelinger med parametersett gitt av fire parametre for hver av de to modellkomponentene. Dermed kan det beregnes hvor sannsynlig det er at en gitt lydramme hører til disse to fordelingene. Modellparametrene estimeres initielt med forventningsmaksimering, og oppdateres etter hvert som systemet kjøres. Transientdeteksjonsegenskapen som benyttes er basert på sannsynlighetsmålet for klassetilhørighet, der rammen klassifiseres som transientlyd om denne sannsynligheten er tilstrekkelig lav for de to modellkomponentene [2].

Transientdeteksjonsnøyaktigheten har tidligere lidd ved lengre perioder uten talepåtrykk, ettersom modellens talekomponent oppdateres fortlø-

pende med støydata som grunnlag (problemet omtales som *driving*). For etterfølgende, reell taleaktivitet vil sannsynligheten for klassetilhørighet en periode være lav for begge modellkomponentene, og denne taleaktiviteten vil klassifiseres som transientaktivitet. Det er foreslått noen endringer for deteksjonsalgoritmen. Viktigst av disse er innføringen av oppdateringsstopp for modellparametrene ved fravær av taleaktivitet. Det foreslåes å benytte dobbelt parametersett, der det ene oppdateres som normalt, samt danner grunnlag for oppdateringsbetingelsen, mens talekomponentens middelverdiparameter i det andre parametersettet fryses ved indikasjon på fravær av taleaktivitet. Det er forsøkt flere metoder for å avgjøre når modellen burde fryses og ikke, og det er endt på en metode som benytter et vektet snitt av den empiriske komponentsannsynligheten over alle delbånd.

Den foreslåtte deteksjonsalgoritmen er implementert i MATLAB fra bunnen av, med sanntidsvennlige løsninger. Det er gjennomført forskjellige tester av algoritmen med og uten de foreslåtte endringene. Resultatene er sammenlignet med andre deteksjonsalgoritmer, og den utvidede taledetektoren oppnår større deteksjonsnøyaktighet i de fleste tilfeller. Transientdeteksjon skiller seg positivt ut med sterkt forbedrede resultater.

Abstract

This thesis investigates a system for detection of voice activity in an arbitrary sound signal. The system is implemented with an extention enables the system to detect short, abrupt, energy-intensive sounds, named *transient sounds*. Transient detection is a desired feature in systems for automatic gain control, as it enables the system to reduce the effects of long-term attenuation of speech consecutively following transient sounds. For this consecutive period, the attenuation may be so severe that the content may be unintelligible.

The system implemented is based on frame-wise, smoothed, logarithmic sub-band energy as detection feature [1]. Detection is done independently per sub-band, and the final frame decision is determined on the basis of a simple, weighted majority vote. A statistical model of the energy distribution is the underlying basis for the system, where the signal is considered to be comprised of two model components: noise/non-speech and speech. It is assumed that the input signal consists of an ever-present noise component, in addition to a occasionally present speech component. The energy distribution for each of these sound classes is modelled as Gaussian distributions with parameter sets given by four parameters for each of the two model components. Thus it is possible to calculate the likelihood of an observed frame to belong to these two distributions. The model parameters is estimated initially using expectation maximization, and updated sequentially as the system is active. The transient detection feature that is used, is based on the likelihood values for the current frame being speech or non-speech. The frame is classified as transient if the sum of these likelihoods are sufficiently low [2].

The accuracy of transient detection has in earlier work suffered if longer periods of time with voice activity absent have occurred. This is due to that the speech component of the model is updated regardless of whether it is speech and noise, or only noise, that are being presented to the algorithm (this problem is denoted *drifting*). For actual speech consecutive to a long, silent period, the likelihood for both components will be low, and the speech frame will erroneously be classified as transient. A few alterations are proposed for the detection algorithm. Of these, the most important is the method not updating the speech mean of the model parameters in the absence of voice activity. It is proposed to utilize double parameter sets, so that one may be used for voice activity detection and would be updated regardless of signal content. The other is used for transient detection, and requires voice activity for the speech mean of the parameter set to be updated. Several methods for determining the presence of speech is investigated, and a method utilizing the empirical class occurrence likelihood weighted over all sub-bands is developed.

The suggested detection algorithm is implemented from scratch in MATLAB, with real-time friendly solutions. Various tests are conducted, testing the algorithm with and without the suggested changes. The results are compared to a few other detection algorithms, and the extended voice activity detector is seen to achieve greater detection accuracy in most tests. Transient detection is showing greatly improved results.

Forord

Jeg ønsker å rette en stor takk til mine to veildere: fagveileder Torbjørn Svendsen ved IET og ekstern veileder Øystein Birkenes ved Cisco Systems Norway. Deres hjelp og innspill har vært viktige bidrag i arbeidet, og især har det vært nyttig å få innblikk i de praktiske utfordringene ved problemfri lyd i videokonferansesamtaler.

Rapporten er forsøkt skrevet så lettfattelig som mulig, innenfor fagets begrensninger. Ikke minst er det forsøkt å være raus med kommentarer i MATLAB-koden (vedlegg A), da jeg selv har erfart hvordan det er å skulle sette seg inn i en stor mengde mer eller mindre udokumentert datakode.

Innhold

Figurer

Vedlegg:

Tabeller

Kapittel 1

Introduksjon

I mange sammenhenger vil det være nyttig å kunne automatisk avgjøre når tale er tilstede i et lydsignal. Dette er det som menes med *taleaktivitetsdeteksjon* (eng.: *Voice Activity Detection, VAD*), eller bare *taledeteksjon*. Det benyttes oftest en signalmodell der et lydsignal antaes å bestå av to deler: en støykomponent N og en støyfri talekomponent S. Dette er illustrert i (1.1).

$$X_{k+1} = \begin{cases} N & \text{hvis } k+1 \text{ er ikke-tale} \\ N+S & \text{hvis } k+1 \text{ er tale} \end{cases}, \tag{1.1}$$

med andre ord er $S = 0$ om tale er fraværende.

Ettersom det antaes en todelt signalmodell, er en taledetektor (eng.: *Voice Activity Detector, VAD*) en binær klassifiserer, der klassene benevnes *tale* og *ikke-tale/stillhet/støy*. Signalmodellen antar kontinuerlig tilstedeværelse av støy, om enn av varierende intensitet, og periodevis taleaktivitet. Dette medfører at målet med en taledetektor blir å detektere $S \neq 0$, en oppgave som blir vanskeligere dess lavere signal-støy-forhold (*Signal-to-Noise Ratio, SNR*).

Om et talebehandlingssystem har kjennskap til signalets innhold, kan dette ofte utnyttes for å gjøre systemet bedre eller mer effektivt. Taledeteksjon

åpner for at signalsegmenter som inneholder tale kan behandles annerledes enn de delene der tale er fraværende. Blant annet benyttes dette i diverse metoder for talekomprimering med variabel bitrate [3]. En slik talekodek kan benyttes for å redusere bitraten brukt på segmenter som klassifiseres som ikke-tale ved for eksempel å unnlate å kode bølgeformen i disse delene på enkodersiden, og heller erstatte segmentet med passende, auto-generert støy på dekodersiden. Slik senkes den gjennomsnittlige bitraten, men dette er helt avhengig av en god taledetektor.

Forskjellige bruksområder fordrer forskjellige krav til en taledetektor. I talekodek-eksempelet som er nevnt ovenfor, vil det eksempelvis være viktig å ikke feilklassifisere talesegmenter som ikke-tale, da disse i så fall vill klippes vekk i det kodede signalet. Dermed kreves det at en talekoder som skal benyttes til et slikt formål har en lav andel feilklassifiseringer av talesegmenter. I og med at en taledetektor er en binær klassifiserer, kan dette enkelt oppnås på bekostning av feilklassifiseringsraten av ikke-talesegmenter. Alternativet er å utvikle en taledetektor som presterer generelt bedre, men dette er igjen en større utfordring.

Andre eksempler på bruk av taledeteksjon finnes i for eksempel talegjennkjenningsapplikasjoner (eng.: *Automatic Speech Recognition, ASR*) og taleforbedringssystemer [1]. Taledeteksjon kan også bygges inn i systemer for utjevning av lydnivået i lydsignaler. Dette benytter Cisco i sine videokonferansesystemer, noe som er bakteppet for dette arbeidet.

1.1 Talekarakteristika

For å kunne detektere taleaktivitet i et lydsignal kan det taes utgangspunkt i en rekke egenskaper som kan utledes fra lydsignalet. I kombinasjon med kjennskap til hvordan disse egenskapene er karakteristiske for talesignaler, kan signalegenskapene benyttes til å fatte en automatisert avgjørelse om hvorvidt taleaktivitet er tilstede i et inngangssignal eller ikke. Egenskaper som kan utnyttes er for eksempel stemthet, energinivå eller forskjellige frekvensmål – for å nevne noen få. Det finnes mange slike egenskaper, basert på objektive, kvantitative mål som i større eller mindre grad er enkle å finne. Mange av signalegenskapene tar utgangspunkt i hvordan signalener-

gien fordeler seg i frekvensspekteret, ettersom tale her har karakteristiske mønstre. Dette skyldes de fysiologiske begrensningene taleproduksjonsorganene er underlagt, og gir mer eller mindre definerte rammer for hvilke lyder som er mulige å produsere for disse organene. Dette bidrar til å gjøre deteksjonsoppgaven enklere, ettersom disse begrensingene kan benyttes til å skille tale fra lyder som ikke er underlagt samme restriksjoner og mønstre.

I taledeteksjon er det tidligere brukt mange forskjellige egenskaper for deteksjon, blant annet null-krysningsrate [4, 3], energinivå [4, 1, 3] og lineære prediksjonskoeffisienter [3, 5, 6]. Ofte benyttes en kombinasjon av disse og andre [3], slik at klassifiseringen skjer på et mest mulig robust grunnlag. Signalenergi, både direkte og avledede egenskaper, kan kanskje sies å være det mest avslørende med tanke på taleaktivitet. Meget sjelden vil taleaktivitet forekomme i et signal uten at det også er en økning i energinivået for de aktuelle talesegmentene [7]. Dermed er energinivået en relativt god indikator på taleaktivitet, eksempelvis kan en enkel taledetektor i prinsippet realiseres ved å klassifisere alle signalsegmenter som har et energinivå over en gitt terskel som tale.

Mange algoritmer for taledeteksjon har da også benyttet mer eller mindre heuristisk bestemte terskelverdier for forskjellige signalegenskaper, noe som ofte kuliminerer i varierende resultater for forskjellige testdata. Dette kan fungere bra i gitte sammenhenger, gjerne der systemene skal benyttes under kontrollerte forhold. Det er da nærliggende å anta at et deteksjonssystem som tilpasser seg de omgivelsene det måtte utsettes for, vil prestere bedre enn en statisk deteksjonsmetode gjør for varierende omgivelser. I [1, 8] er det tatt en mer statistisk tilnærming til deteksjonsproblemet, og det er utviklet taledetektorer som mer eller mindre kontinuerlig tilpasser seg forholdene de opererer under.

Det er imidlertid fortsatt de samme signalegenskapene som ligger til grunn. Med et bedre rammeverk og bedre egenskapsmodeller, kan dynamiske deteksjonsalgoritmer ofte gi økt nøyaktighet sammenlignet med statiske detektorer, selv om disse tar flere egenskaper i betraktning [1, 8]. Altså er det ikke nødvendigvis den algoritmen som undersøker flest mulige signalegenskaper som presterer best – det kan være vel så viktig å kombinere og modellere få på en fornuftig måte.

1.2 Lydnivåutjevning

Automatisk lydnivåutjevning (eng.: *Automatic Gain Control, AGC*) er en betegnelse på teknikker for å sørge for at lydnivået i et signal varierer jevnt og innenfor gitte grenser. Utjevningen foregår ved å dempe/forsterke lydnivåer over/under en gitt grense, med parametrene *angrepstid* (eng.: *attack*) og *falltid* (eng.: *release*) [9]. Angrepstiden styrer hvor raskt dempning/forsterkning skal tre i kraft, mens falltiden kontrollerer hvor raskt dempningen/forsterkningen skal avta. Normalt er disse parametrene satt slik at angrepstiden er meget rask (typisk noen få millisekunder), mens falltiden er lengre (kan være noen tusen millisekunder). Forholdsvis lang falltid er nødvendig for at dempnings-/forsterkningssprangene ikke skal avta for brått. Brå endringer i dempning/forsterkning vil bidra til, i stedet for å hjelpe mot, problemet lydnivåutjevningen er ment å skulle redusere effekten av, ettersom endringene i dempning/forsterkning tydelig høres som markante sprang i lydnivå.

Lydnivåutjevning benyttes i videokonferansesystemer fra blant andre Cisco, og bidrar til at det er et jevnt lydnivå på samtalen, uavhengig av avstand til mikrofon o.l. Om det er store variasjoner i effektnivået under en samtale, vil dette kunne oppfattes som slitsomt for deltagerne i mottagerenden. AGC løser dette problemet rimelig tilfredsstillende, men med lydnivåutjevningsmetodene følger det noen utfordringer. Én av disse utfordringene gir seg til kjenne ved forekomster av brå, energirike lyder av kort varighet i et talesignal. En slik plutselig endring i energinivået[1] fører til at AGC-systemet innfører dempning av utgangssignalet, slik at lydnivået oppleves rimelig jevnt i forhold til normalnivået for samtalen. Denne dempningen innføres stort sett såpass raskt at signalet blir tilstrekkelig dempet, og videosamtalens andre deltagere vil forhåpentligvis ikke sjeneres stort av spranget i energinivå hos avsender. Imidlertid vil den store dempningen som nå er påført inngangssignalet vedvare en stund (gitt av falltiden), uavhengig av varigheten til den energirike lyden. Dermed vil etterfølgende taleaktivitet med normalt effektnivå dempes like mye som

[1] «Energi» brukes i denne oppgaven litt omtrentlig, men stort sett kan energien i et signalsegment sees på som summen av de kvadrerte punktprøvene i dette segmentet. Om noe annet menes, er dette spesifisert.

den energirike lyden, og talen vil som følge av dette oppfattes som uhørlig på grunn av lavt lydnivå. Dempningen vil normalisere seg innen en viss tid, men effekten er såpass merkbar at Cisco ønsker å ha mulighet for å detektere slike brå, energirike lyder, for å kunne tilpasse lydnivåutjevningen slik at disse ikke ødelegger for umiddelbart etterfølgende tale. For å muliggjøre slik deteksjon, er det tenkt å benytte en utvidet form for taledeteksjon med mulighet for deteksjon av korte, brå, energirike lyder, såkalte *transientlyder*.

1.3 Transientlyder

I en hvilken som helst ikke-kontrollert situasjon der lyd fanges opp via mikrofon, er det en viss risiko for at brå, energirike lyder skal opptre. Som nevnt i avsnitt 1.2 kan slike lyder medføre redusert taleforståelighet, ettersom umiddelbart etterfølgende tale dempes for mye. Disse lydene, som her benevnes *transientlyder* (eller bare *transienter*), er ikke entydig definert. Det er den raske økningen i energinivå, den korte varigheten og den forholdsvis høye energien i lydene som er de viktigste egenskapene disse lydene har, og som gir forklarer hvorfor «transienter» er valgt som benevning.

Et av de store problemene når det kommer til å definere transientlyder, er lydenes varighet. Det som omtales her som transientlyder har ofte varighet mellom 100 ms til 1000 ms. Delvis skyldes den lange varigheten av lydene lydutbredelsens natur, og i kombinasjon med omgivelser som reflekterer lydene, vil varigheten nødvendigvis strekke ut i tid utover varigheten til hva som normalt vil menes med «transienter». Nettopp *etterklangstiden* i et rom vil bidra til at transientlydene strekker seg ut i tid, med avtagende energinivå, slik at det som initielt var en energirik lyd med brå ansats, etter hvert vil ha egenskaper (i hovedsak energinivå) som er vanskelig å skille fra talesignalet. Dette bidrar til å gjøre varighetsbestemmelse utfordrende. Om varigheten skal kunne avgjøres automatisk er det avgjørende å utarbeide klare definisjoner og metoder, for eksempel ved å definere at en transientlyd er over når energinivået har falt et gitt antall desibel fra toppnivået.

For typiske videokonferansesituasjoner og lignende, kan transientlyder ek-

sempelvis være følgende: dørsmell, aggressiv lukking av bok/pc nær mikrofon, slag/kopp i bordet eller voldsom kremting i umiddelbar nærhet til mikrofon – for å nevne noen få. Akkurat smelling av dør stikker seg ut som en særlig kort, energirik lyd som relativt ofte kan forekomme i en typisk møteromssituasjon. Det samme gjelder ufrivillig dunking av kopp ned i bordet – en lyd som kan ha et meget høyt energinivå i forhold til normalt talenivå om det gjøres i umiddelbar nærhet til mikrofonen. Alle disse eksemplene illustrerer korte lyder, men de har ikke noen bestemt eller entydig varighet.

Energinivået i transientlyden vil normalt avta gradvis fra ansats, men denne falltiden vil variere fra lydkilde til lydkilde, og er i tillegg veldig avhengig av de akustiske omgivelsene (i hovedsak etterklangstid o.l.). Møterom, som videokonferansesystemer gjerne er plassert i, vil normalt sett ha rimelig gode akustiske forutsetninger for taleforståelighet, og følgelig vil romakustikk ikke nødvendigvis medføre de helt store problemene. Møterom er imidlertid oftest langt fra ekkofrie, og NS 8175:2012 tillater opptil 480 ms etterklangstid for et møterom med 3 m takhøyde bygget etter klasse C [10]. Av dette sees det at også møterom som følger normal byggestandard vil bidra til å strekke transientene ut i tid med avtagende energinivå.

Transientlydene som er benyttet i dette arbeidet er beskrevet i avsnitt 3.1.3, og en metode for deteksjon av slike står omtalt i avsnitt 2.3.1.

1.4 Om arbeidet

For å kunne håndtere problemene som oppstår i lydnivåutjevningssystemer ved transientlyd i tale, er det ønskelig å oppnå to ting: for det første er det nødvendig å ha en metode for å oppdage transientlyder, og da spesielt transienter som opptrer i talesegmenter, der de forårsaker mest problemer. Dermed kommer også behovet for en taleaktivitetsdetektor som, på en god måte, evner å detektere taleaktivitet. Det finnes mange taledeteksjonsalgoritmer allerede, og metoden som er undersøkt i dette arbeidet baserer seg på [1, 2]. Taleaktivitetsdeteksjon med mulighet for å detektere transientlyder kalles *utvidet taleaktivitetsdeteksjon* (eng.: *Extended Voice Activity Detection, EVAD*).

Det er derfor i arbeidet med denne rapporten undersøkt forskjellige endringer i denne deteksjonsmetoden, med fokus på forbedring av transientdeteksjonen under vanskelige forhold. Ved lengre tids fravær av taleaktivitet i inngangssignalet, vil transientdeteksjonsmetoden som er foreslått i [2] prestere særdeles dårlig. Denne problematikken utdypes i avsnitt 2.3.2.

Tidligere implementasjoner av deteksjonsalgoritmen som er beskrevet i [1] har ikke vært særlig sanntidsvennlige, så kjørbar MATLAB-kode har i stor grad blitt implementert på nytt. Dette har utgjort en stor del av arbeidet som er utført for denne rapporten, inkludert mye testing for å forsikre om at metodene fungerer korrekt. Logiske feil og ukloke valg i implementasjonen kamufleres godt i sluttresultatene, så slike problemer er vanskelig å avdekke. Videre er det lagt vekt på å utvikle en god og fleksibel løsning for generering av varierte testdata, slik at evalueringen av deteksjonsmetodene blir gjort på et så virkelighetstro grunnlag som mulig. Løsningen er laget for en viss grad av fleksibilitet, slik at testdata av forskjellig karakter kan genereres med samme rutine.

Sluttproduktet som skal stå igjen etter dette arbeidet er forhåpentligvis et system som løser de utfordringene som er beskrevet i dette kapittelet. I rapporten presenteres bakgrunnsmateriale for taledeteksjonsproblemet, modellrammeverk for en taledetektor, fremgangsmåte for implementasjonen og testrutinen. Resultatene fra testene som gjøres legges også frem, samt en enkel analyse av disse. Til slutt foreslåes det videre muligheter for utvikling av metoden.

Det er i denne rapporten brukt noen typografiske virkemidler for å lettere skille forskjellig typer tekst. **Fet skrift** brukes om vektorer, *Kursiv* om matematiske variabler og `Teletype` for MATLAB-variabler og -rutiner. Sistnevnte type er også tidvis benyttet om tallmerkene som beskriver lydinnholdet i testdataene. Standard bruk av subskript for matematiske variabler er $V_{k,l,z}$, der V er variabelnavn, $k \in \ltimes$ er rammenummer, $l \in [1, \ldots, N]$ gir delbåndsnummer og $z \in \{0, 1\}$ signaliserer lydklassen (ikke-tale/tale). Om én eller flere av disse parametrene er irrelevant for sammenhengen de står i, tillates det at enkelte subskript droppes for enkelhets skyld.

Kapittel 2

Teori & løsninger

2.1 Referanse-algoritmer

I dette arbeidet er det laget en tale- og transientdetektor (omtalt som *HLP EVAD*) basert på taledetektoren presentert i [1, 2] (benevnes heretter *GMM EVAD*), og dennes prestasjoner i forhold til taledetektoren beskrevet i standarden *ITU G.729* [3] (*ITU VAD*) og metoden som er foreslått i [8] (*Sohn VAD*). Det er også sett på to transientdeteksjonsalgoritmer basert på GMM-rammeverket. Transientdeteksjon er implementert for både GMM EVAD og HLP EVAD. ITU VAD og Sohn VAD er uten transientdeteksjon, og er med i testene som referanser for de to taledeteksjonsmetodene. Transientdeteksjonsresultatene fra GMM EVAD og HLP EVAD kan kun sammenlignes med hverandre, da det ikke har lyktes å finne noen sammenlignbar referansedetektor.

2.1.1 ITU G.729

I [3] foreslår *Den internasjonale telekommunikasjonsunion* (eng.: *International Telecommunications Union*, *ITU*) en taledetektor som en del av kommunikasjonsstandarden ITU G.729, der denne kan benyttes for mer effektiv talekoding ved å effektivt kode stille partier i talestrømmen. Det

er kommet forbedringer av denne taledetektoren i senere tid, og den versjonen som er benyttet her er versjonen som ble lansert i 2005: G.729 Vedlegg II [11]. Denne taledetektoren benevnes *ITU VAD* i denne rapporten. Detektoren benytter seg av et 30 ms vindu, delt inn i 15 ms forutgående lyddata, nåværende ramme à 10 ms og 5 ms med fremoverskuende data. Dette vinduet flyttes 10 ms frem for hver nye ramme. Analysen som gjøres i dette vinduet består i å hente ut noen egenskaper ved signalet, blant annet: korttidsenergi (både helbånds- og lav-delbåndsenergi brukes), nullkrysningsrate og linjespektralpar. Det sees i hovedsak på forskjellen mellom de forskjellige egenskapene og et glidende gjennomsnitt av hver enkelt egenskap over ikke-tale segmentene allerede detektert og klassifisert. Slik sammenlignes hele tiden aktuelle egenskapers avvik fra tilsvarende egenskaper for ikke-tale-segmenter, og algoritmen er således i stand til å skille mellom tale og ikke-tale. Implementasjon for MATLAB er hentet fra [12], og ingen endringer er gjort i denne implementasjonen.

2.1.2 Sohn VAD

Taleaktivitetsdetektoren som er beskrevet i [8] (benevnes her *Sohn VAD*), er basert på en sannsynlighetsmodell for tilstedeværelse av tale i inngangssignalet. Mer konkret benyttes forholdet mellom de to sannsynlighetene for klassetilhørighet som egenskap for klassifiseringsavgjørelsen. Denne metodikken er relativt lik fremgangsmåten som benyttes i GMM EVAD, og vil i så måte antaes gi resultater som er rimelig sammenlignbare med denne.

I tillegg benyttes det en glattingsmetode i Sohn VAD som baserer seg på overgangssannsynligheten mellom de to lydklassene. Denne varianten av intra-ramme-glatting av klassifiseringsavgjørelsen er litt annerledes enn den som benyttes i ITU VAD og de GMM-baserte taledetektorene, da disse baserer seg på en noe enklere metode for begrensning av overgang fra taleklassifisering til ikke-tale-klassifisering (se avsnitt 2.2.6).

Metoden er inkludert i dette arbeidet som en sammenlignbar referanse for taledeteksjonsmetodene som undersøkes. Implementasjonen for MATLAB er hentet fra lydbehandlingsbiblioteket *VOICEBOX* [13], og er umodifisert.

2.2 Taledetektor basert på gaussiske blandingsmodeller

Metodikken som er benyttet i denne oppgaven baserer seg på taledetektoren foreslått i [1], og MATLAB-koden er også opprinnelig skrevet av forfatterne av [1], med videre modifikasjoner og feilkorrigeringer beskrevet i [2]. Metoden er imidlertid implementert på ny i MATLAB, blant annet for å være godt tilpasset en eventuell sanntidsimplementasjon. Mer om dette i avsnitt 3.2.

2.2.1 Signalegenskap for taleaktivitetsdeteksjon

For å kunne analysere inngangssignalet med mål om å detektere forskjellige typer signalaktivitet, må det bestemmes hva slags signalegenskaper som skal ligge til grunn for analysen. Disse egenskapene omtales ofte som en *egenskapsvektor*, som kan være av variabel lengde, fra én og oppover. I dette arbeidet er *aktivitetsdeteksjonen* (begrepet dekker både tale- og transientdeteksjon) i all hovedsak basert på kun én egenskap, nemlig *glattet, logaritmisk ramme-energinivå*. Dette er en egenskap som er en funksjon av delbåndsenergien i en bestemt ramme, og som estimerer det logaritmiske energi-innholdet i hvert delbånd for hver ramme, og dette ene målet forteller mye om signalinnholdet (under normale verdier for signal-støy-forhold). Ved å undersøke dette energimålet, og sammenligne det observerte nivået med forhåndsdefinerte nivåmodeller for både tale og ikke-tale, kan algoritmen gi et sannsynlighetsmål på klassetilhørighet. Som en bimodal deteksjonsalgoritme, vil detektoren klassifisere rammene som enten den ene eller den andre klassen, avhengig av om det observerte energimålet ligger over eller under en *desisjonsgrense* $\grave{\theta}$ (se avsnitt 2.2.6).

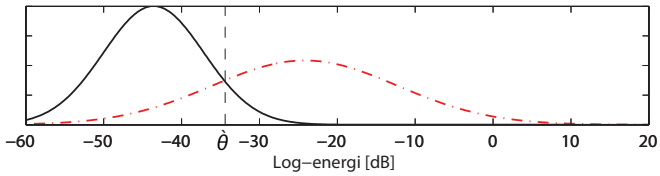
For hver ramme k beregnes det et logaritmisk energinivå $\bar{x}_{k,l}$ per delbånd l, gitt av (2.1):

$$\bar{x}_{k,l} = 10 \log_{10} \left(\frac{1}{f_{l+1} - f_l} \sum_{j=f_l}^{f_{l+1}-1} |Y_{k,j}|^2 \right), \tag{2.1}$$

der f_l er første koeffisientindeks i delbånd l og $|Y_{k,j}|^2$ er den j-te DFT-koeffisienten i ramme k [1]. Det logaritmiske delbånds-energinivået glattes så for å unngå de helt store variasjonene fra ramme til ramme. Se avsnitt 2.3.3 for mer om glattingsmetoden som benyttes.

2.2.2 Modellering av energifordeling

Denne detektoren baserer seg på en generativ modell for diskriminering mellom tale og ikke-tale, der glattet, logaritmisk energi per delbånd modelleres som en gaussisk blandingsmodell (eng.: *Gaussian Mixture Model*, *GMM*) med to komponenter. Altså modelleres både ikke-tale og tale i realiteten som hver sin normalfordeling, hver med de to parametrene middelverdi og varians, samt en vektingsparameter som uttrykker a priori-sannsynlighet for tale/ikke-tale (benevnes også *klassesannsynlighet*). En slik to-deling av energifordelingen omtales også som *bimodal fordeling*, og er illustrert i Figur 2.1. Taledetektoren gjør analysen rammevis, og hver ramme splittes i mel-delbånd (det er benyttet 8 delbånd i forsøkene – antallet kan justeres rimelig enkelt). Modelleringen gjøres individuelt for hvert delbånd, og følgelig vil hele prosessen beskrevet i dette avsnittet være identisk for hvert av disse båndene. Videre beskrivelse gjelder derfor for ett enkelt delbånd, og prosessen gjentaes så for de resterende båndene.



Figur 2.1: Energifordelingsmodell som viser de to komponentene (heltrukken, svart for støy, og stiplet, rød for tale), samt teoretisk desisjonsgrense θ.

De to komponentene antaes å være tilnærmet ukorrelerte, og det trengs følgelig ikke å estimere kovarians, da denne tilnærmes som null. Det logaritmiske energi-innholdet estimeres for hvert delbånd, og dette glattes ved å filtrere et lite antall rammer (se avsnitt 2.3.3). Det glattede energimå-

let er observasjonen som blir lagt til grunn for klassifiseringen, i tillegg til de oppdaterte modellparametrene. Videre gjøres selve klassifiseringen per delbånd, uavhengig av hverandre, ved at det beregnes en desisjonsgrense for energimålet. Det aktuelle delbåndet merkes som ikke-tale eller tale avhengig av om energimålet er henholdsvis. lavere eller høyere enn denne grensen. Det vil også være mulig å se på hvor langt unna denne grensen observasjonen ligger, og gi et sannsynlighetsmål for tilhørighet til hver av de to klassene, noe som gjør det mulig å si noe om hvor god klassifiseringsavgjørelsen er. Dette målet kan uttrykkes som sannsynligheten for at observasjonen hører til klassen, gitt modellparametrene:

$$p\left(x_{k,l}|\lambda\right) = \sum_{z} p\left(x_{k,l}, z|\lambda\right) = \sum_{z} p\left(x_{k,l}|z,\lambda\right) p\left(z\right), \tag{2.2}$$

hvor $x_{k,l}$ er k-te observasjon (energinivå i ramme k) og $z \in \{0,1\}$ benevner hhv. ikke-tale ($z = 0$) og tale ($z = 1$). Modellparametrene er uttrykt ved $\lambda_{k,l} \triangleq \{\mu_{k,l,z}, \kappa_{k,l,z}, w_{k,l,z} | z = 0;1\}$, der $\mu_{k,l,z}$ benevner middelverdi, $\kappa_{k,l,z}$ er varians og $w_{k,l,z}$ brukes om a priori klassesannsynlighet; alt for komponent z, delbånd l og rammenummer $k \in \mathbb{N}$.

Ettersom modellen er basert på normalfordelte komponenter, er sannsynligheten for observasjonen, gitt klassifisering og modellparametre, utrykt ved

$$p\left(x_{k,l}|z,\lambda_{k,l}\right) = \frac{1}{\sqrt{2\pi\kappa_{k,l,z}}} \exp\left\{-\left(x_{k,l} - \mu_{k,l,z}\right)^2 / 2\kappa_{k,l,z}\right\}. \tag{2.3}$$

Modellparametrene $\lambda_{k,l}$ estimeres initielt ved å maksimere (2.4), som beskrevet i avsnitt 2.2.3. Energifordelingsmodellen er illustrert i Figur 2.1, med den teoretiske desisjonsgrensen $\theta_{k,l}$ markert med stiplet, vertikal linje.

2.2.3 Trening og initialisering av modell

Energifordelingsmodellen oppdateres etterhvert som nye rammer blir analysert og klassifisert, men modellen må naturligvis initialiseres. Taledetek-

toren i [1] benytter såkalt blind trening (eng.: *unsupervised training*) av modellen. Initielle modellparametre for delbånd l, $\lambda_{0,l} = \{\mu_{0,l,z}, \kappa_{0,l,z}, w_{0,l,z}\}$, $z = \{0, 1\}$, estimeres da ved å se på et lite antall $(M + 1)$ lydrammer (i dette arbeidet er det de 60 første som brukes, jf. [1]), og benytte en forventningsmaksimeringsalgoritme (eng.: *Expectation Maximization*, *EM*) til å finne modellparametre som svarer best til innholdet i disse rammene. Parametrene er ilagt visse begrensninger, som f.eks. at middelverdi for talemodellen må være større enn for støymodellen (da tale også inneholder støy) og at støy antaes å være mer stasjonær enn tale, med følgelig mindre varians.

Da det antaes uavhengige delbåndsfordelinger fra ramme til ramme, kan felles sannsynlighetsfordeling for alle rammer utrykkes som

$$p\left(\mathbf{x}|\lambda_{k,l}\right) = \prod_{k=0}^{M} p\left(x_{k,l}|\lambda_{k,l}\right), \mathbf{x} \triangleq \{x_0, x_1, x_2, \ldots, x_M\}, \tag{2.4}$$

der $M + 1$ er antall rammer som blir benyttet i initialiseringen. Ligning (2.4) maksimeres for å finne de ideelle modellparametrene. I praksis løses dette av forventingsmaksimeringsalgoritmen.

Metodene som benyttes er uendret fra [1], og er ellers beskrevet der.

Formler som benyttes for parameter-re-estimering i EM-algoritmen følger i ligningene (2.5a)–(2.6d) [1].

$$\overline{w}_{0,l,z} = \frac{1}{M+1} \sum_{j=0}^{M} p\left(z|x_{j,l}, \lambda'_{0,l}\right) \tag{2.5a}$$

$$\overline{\mu}_{0,l,z} = \frac{\sum_{j=0}^{M} x_{j,l} p\left(z|x_{j,l}, \lambda'_{0,l}\right)}{(M+1)\,\overline{w}_{0,l,z}} \tag{2.5b}$$

$$\overline{\kappa}_{0,l,z} = \frac{\sum_{j=0}^{M} \left(x_{j,l} - \overline{\mu}_{0,l,z}\right)^2 p\left(z|x_{j,l}, \lambda'_{0,l}\right)}{(M+1)\,\overline{w}_{0,l,z}} \tag{2.5c}$$

$$p\left(z|x_{j,l},\lambda'_{0,l}\right) = \frac{w'_{0,l,z}p\left(x_{j,l}|z,\lambda'_{0,l}\right)}{\sum_z w'_{0,l,z}p\left(x_{j,l}|z,\lambda'_{0,l}\right)}. \tag{2.5d}$$

Også algoritmen som brukes til å initialisere modellparametrene må gies fornuftige startverdier, så initialverdier $\lambda'_{0,l}$ for EM-algoritmen settes som følger:

Middelverdi $\mu'_{0,l}$: Initiell middelverdi estimeres ved å benytte en *k-middel*-algoritme (eng.: *k-means*), som igjen initialiseres ved å anta at den mest energirike halvparten av initialiseringsrammene tilhører talemodellen, og motsvarende halvpart tilhørende modellen for ikke-tale. Algoritmen regrupperer rammene i to grupper, basert på minste euklidske avstand til gruppe-senteret, før dette oppdateres og prosessen gjentaes. Iterasjonene stoppes ved konvergens innenfor et visst avvik, alternativt avsluttes prosedyren om konvergens ikke er oppnådd etter et visst antall regrupperinger.

Varians $\kappa'_{0,l}$: Basert på rammegrupperingen fra k-middel-algoritmen beregnes varians for de to gruppene med et standard varians-estimat gitt ved $Var(X) = \frac{1}{M}\sum_{i=1}^{M+1}(x_{i,l} - \mu'_{0,l})^2$.

Vekting $w'_{0,l}$: Parameteren svarer til a priori-sannsynlighet for hver av modellkomponentene, og ettersom denne er ukjent ved initialisering, settes vekten lik for begge komponentene ($w_{0,l,z} = 0{,}5\ \forall z$).

Normalt vil standard, sekvensiell re-estimering videre gi (2.6a)–(2.6d):

$$w_{k+1,l,z} = \frac{\sum_{j=k-K+1}^{k} p\left(z|x_{j,l},\lambda_{k,l}\right) + p\left(z|x_{k+1,l},\lambda_{k,l}\right)}{K+1}, \tag{2.6a}$$

$$\mu_{k+1,l,z} = \frac{\sum_{j=k-K+1}^{k} x_{j,l}p\left(z|x_{j,l},\lambda_{k,l}\right) + x_{k+1,l}p\left(z|x_{k+1,l},\lambda_{k,l}\right)}{\sum_{j=k-K+1}^{k} p\left(z|x_{j,l},\lambda_{k,l}\right) + p\left(z|x_{k+1,l},\lambda_{k,l}\right)}, \tag{2.6b}$$

$$\kappa_{k+1,l,z} = \frac{\sum_{j=k-K+1}^{k} \left(x_{j,l} - \mu_{k+1,l,z}\right)^2 p\left(z|x_{j,l}, \lambda_{k,l}\right)}{\sum_{j=k-K+1}^{k} p\left(z|x_{j,l}, \lambda_{k,l}\right) + p\left(z|x_{k+1,l}, \lambda_{k,l}\right)} + \frac{\left(x_{k+1,l} - \mu_{k+1,l,z}\right)^2 p\left(z|x_{k+1,l}, \lambda_{k,l}\right)}{\sum_{j=k-K+1}^{k} p\left(z|x_{j,l}, \lambda_{k,l}\right) + p\left(z|x_{k+1,l}, \lambda_{k,l}\right)}, \tag{2.6c}$$

$$p\left(z|x_{k,l}, \lambda_{k,l}\right) = \frac{w_{k,l,z} p\left(x_{k,l}|z, \lambda_{k,l}\right)}{\sum_z w_{k,l,z} p\left(x_{k,l}|z, \lambda_{k,l}\right)}, \tag{2.6d}$$

der $K = M$ i dette tilfellet, og $\lambda'_{0,l}$ og $\overline{\lambda}_{0,l}$ er modellparametrene fra henholdsvis forrige og nåværende estimeringsiterasjon. Algoritmen re-estimerer parametrene til resultatene konvergerer (evt. avsluttes algoritmen om konvergens ikke oppnås innen et visst antall iterasjoner), og $\overline{\lambda}_{0,l} \Rightarrow \lambda_{0,l}$.

Metoden for re-estimeringen som er gitt i (2.6a)–(2.6d) er imidlertid meget ressurskrevende å beregne, slik at det heller benyttes en sekvensiell oppdatering av modellparametrene. Denne metoden er beskrevet i avsnitt 2.2.4.

2.2.4 Oppdatering av modell

For modelloppdatering taes det utgangspunkt i ligningene (2.6a)–(2.6d) for standard, sekvensiell re-estimering av modellparametrene. Det antaes at inngangssignalet til systemet er såpass saktevarierende at $\lambda_k \approx \lambda_{k-1}$, og følgelig vil $\sum_{j=k-K+1}^{k} p\left(z|x_{j,l}, \lambda_{k,l}\right) \approx \sum_{j=k-K+1}^{k} p\left(z|x_{j,l}, \lambda_{k-1,l}\right)$. Dermed kan summen tilnærmes med nulte ordens moment, $\sum_{j=k-K+1}^{k} p\left(z|x_{j,l}, \lambda_{k-1,l}\right) \approx K w_{k,l,z}$. Ved å så benytte at $\lambda_{k,l} \approx \lambda_{k-1,l}$, fåes

$$\sum_{j=k-K+1}^{k} p\left(z|x_{j,l}, \lambda_{k,l}\right) \approx K w_{k,l,z}, \tag{2.7a}$$

som innsatt i (2.6a) gir

$$w_{k+1,l,z} = \frac{K w_{k,l,z} + p\left(z|x_{k+1,l}, \lambda_{k,l}\right)}{K+1}. \tag{2.7b}$$

Ved å innføre forholdet $\alpha = \frac{K}{K+1}$, blir (2.7b) seende slik ut:

$$w_{k+1,l,z} = \alpha w_{k,l,z} + (1-\alpha)\, p\left(z|x_{k+1,l}, \lambda_{k,l}\right), 0 \leq \alpha \leq 1, \tag{2.7c}$$

der $p\left(z|x_{k+1,l}, \lambda_{k,l}\right)$ beregnes med (2.6d).

Tilsvarende kan en første ordens moment-tilnærming gjøres for første ledd i telleren i (2.6b)

$$\sum_{j=k-K+1}^{k} p\left(z|x_{j,l}, \lambda_{k,l}\right) x_{j,l} \approx K w_{k,l,z} \mu_{k,l,z}, \tag{2.7d}$$

som, innsatt i (2.6b), gir

$$\mu_{k+1,l,z} = \frac{\alpha w_{k,l,z} \mu_{k,l,z} + (1-\alpha)\, p\left(z|x_{k+1,l}, \lambda_{k,l}\right) x_{k+1,l}}{w_{k+1,l,z}}. \tag{2.7e}$$

Samme metodikk brukes for å tilnærme det første teller-leddet i (2.6c) med andre ordens moment:

$$\sum_{j=k-K+1}^{k} \left(x_{j,l} - \mu_{k+1,l,z}\right)^2 p\left(z|x_{j,l}, \lambda_{k,l}\right) \approx K w_{k,l,z} \kappa_{k,l,z}. \tag{2.7f}$$

Settes (2.7f) inn i (2.6c), fåes følgende:

$$\kappa_{k+1,l,z} = \frac{\alpha w_{k,l,z} \kappa_{k,l,z} + (1-\alpha)\, p\left(z|x_{k+1,l}, \lambda_{k,l}\right) \left(x_{k+1,l} - \mu_{k+1,l,z}\right)^2}{w_{k+1,l,z}}. \tag{2.7g}$$

Modellen oppdateres rammevis i henhold til ligningene for modelloppdatering gitt i (2.7c), (2.7e), (2.7g) og (2.6d). Oppdateringen korrigerer opprinnelige modellparametrene med nye parameter-estimater, der α kan tolkes som en fortidsvekt, *glemselsparameteren* α, for i hvor stor grad tidligere modellparametre skal beholdes ($\alpha = 0{,}99$ i dette arbeidet). Etter som $\alpha \to 0$ vil tidligere parametersetts bidrag reduseres fort, og mest vekt blir lagt på senere rammers parametre. Ettersom oppdateringen skjer multiplikativt med tidligere parametre, vil *glemselstiden* (tiden fra k til k' før $\lambda_{k,l}$ har tilnærmet null bidrag til $\lambda_{k',l}$) synke eksponentielt, da $\alpha \leqq 1$. Glemselsparameteren α medfører at det er mulig å justere hvor fort modellen endres ved rammevis oppdatering, og at det dermed er mulig for applikasjonsutvikler å tilpasse taledetektoren til sitt bruk.

2.2.5 Begrensninger for parametersett

Det legges noen begrensninger på estimatene som gjøres i modellinitialiseringen og -oppdateringen. Disse er listet opp i ligningene (2.8a)–(2.8h):

$$\mu_{k,l,0} < \mu_{k,l,1} \tag{2.8a}$$
$$\mu_{k,l,0} < \theta_{k,l} < \mu_{k,l,1} \tag{2.8b}$$
$$\mu_{k,l,1} > \delta + \mu_{k,l,0}, \text{ for bimodal fordeling} \tag{2.8c}$$
$$\mu_{k,l,1} = \max\left\{\mu_{k,l,1}, \mu_{k,l,0} + \delta\right\}, \text{ for unimodal fordeling} \tag{2.8d}$$

$$\kappa_{k,l,0} < \kappa_{k,l,1} \tag{2.8e}$$
$$\kappa_{k,l,1} = \max\left\{\kappa_{k,l,0}, \kappa_{k,l,1}\right\} \tag{2.8f}$$

$$w_{k,l,0} > \epsilon w_{k,l,1} \tag{2.8g}$$
$$w_{k,l,0} = 1 - w_{k,l,1}, \tag{2.8h}$$

der ϵ og δ er to størrelser som settes manuelt i implementasjonen for at den bimodale fordelingsmodellen skal kunne takle en energifordeling som reelt er unimodal, ved å fungere som minstemål på henholdsvis empirisk

talesannsynlighet og forholdet mellom modellert signalenergi for tale og støy. I implementasjonen som er benyttet i dette arbeidet, er $\delta = 3{,}5\,\mathrm{dB}$. Når det gjelder begrensningen i (2.8g), er den beskrevet i [1] som $w_{k,1} = \max\{w_{k,1}, \epsilon'\}$, til tross for at den i implementasjonen fra forfatterne av [1] heller er benyttet (2.8g) og med $\epsilon = 8$. Det antydes i [1] at forfatterne har eksperimentert noe med disse parametrene, og funnet en løsning som fungerer tilfredsstillende.

2.2.6 Klassifisering

Når modellparametrene er funnet, kan det beregnes en teoretisk desisjonsgrense $\theta_{k,l}$ som skiller mellom de to modellkomponentene.

$$p\left(\theta_{k,l}|z=1,\lambda_{k,l}\right)p\left(z=1\right) = p\left(\theta_{k,l}|z=0,\lambda_{k,l}\right)p\left(z=0\right) \tag{2.9}$$

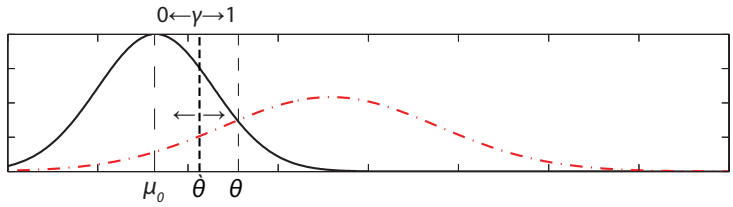
Den teoretiske desisjonsgrensen er altså krysningspunktet mellom de to modellkomponentene, jf. (2.9). Som en følge av (2.8c) vil grensen alltid befinne seg på intervallet $\mu_{k,l,0} < \theta_{k,l} < \mu_{k,l,1}$, som illustrert i Figur 2.1.

I mange sammenhenger vil det være ønskelig at klassifiseringsavgjørelsen favoriserer tale mer enn ikke-tale, da kostnaden ved feilklassifisert tale ofte er større enn ved feilklassifisert ikke-tale. Derfor er det ikke den teoretiske desisjonsgrensen som benyttes i praksis, men det er innført en parameter $0 \leq \gamma \leq 1]$ som styrer graden av talefavorisering på bekostning av ikke-tale. Det beregnes en ny desisjonsgrense $\grave{\theta}_{k,l}$ som er gitt av (2.10).

$$\grave{\theta}_{k,l} = \mu_{k,l,0} + \gamma\left(\theta_{k,l} - \mu_{k,l,0}\right), \tag{2.10}$$

der $0 \leq \gamma \leq 1$ er en *forskyvningsparameter* som bestemmer i hvor stor grad desisjonsgrensen skal forskyves mot ikke-tale-middelet $\mu_{k,l,0}$, som illustrert i figur 2.2.

Klassifisering for rammen som helhet avgjøres ved vektet, simpelt flertall blant delbåndsavgjørelsene. Vektene kan enkelt varieres for å la enkelte delbånd veie mer eller mindre i klassifiseringsavgjørelsen.



Figur 2.2: Desisjonsgrenseforskyvningen mellom $\theta_{k,l}$ og $\mu_{k,l,0}$ bestemmes av γ.

Til slutt er det implementert en glattingsmetode for rammedeteksjonsavgjørelsen, basert på AMR2-standarden, jf. [14]. Denne bidrar til en viss kontinuitet i klassifiseringen ved å korrigere sporadiske klassifiseringsfeil om de omkringliggende rammene er konsistent klassifisert. Kort oppsummert går denne metoden ut på å kreve noen få etterfølgende ikke-talerammer før algoritmen godtar at klassifiseringen av aktuell ramme går fra tale til ikke-tale. Detaljert beskrivelse av metoden finnes i [14].

2.3 Foreslåtte løsninger

Metoden som er beskrevet ovenfor i avsnitt 2.2, er grunnlaget for arbeidet i denne oppgaven. Fremgangsmåten er beholdt fra [1], men det er foreslått noen endringer, og hele implementasjonen er dermed gjort på nytt.

2.3.1 Transientdeteksjon

Det er i [2] introdusert en form for transientdeteksjon, basert på de modellbetingede sannsynlighetene for klassetilhørighet i ramme k og delbånd l, $p(z)\,p(x_{k,l}|z,\lambda_{k,l})$ for $z \in \{0;1\}$, der $p(z)$ tilnærmes av $w_{k,l,z}$. Denne sannsynligheten summeres over alle delbånd, slik at det for hver ramme beregnes et sannsynlighetsmål for hver av lydklassene, $\sum_l p(z)\,p(x_{k,l}|z,\lambda_{k,l})$ for $z \in \{0;1\}$. Fra dette beregnes det et transientmål f_k for ramme k gitt ved (2.11).

$$f_k = \frac{1}{\sum_l p\left(x_{k,l}|\lambda_{k,l}\right)}, \tag{2.11}$$

der $p\left(x_{k,l}|z,\lambda_{k,l}\right)$ er sannsynligheten for klassetilhørighet ($z = 0$ og $z = 1$) gitt modellparametrene $\lambda_{k,l}$.

Transientmålet i (2.11) sier noe om sannsynligheten for at lydrammen hverken er tale eller ikke-tale, basert på hvorvidt det glattede, logaritmiske energinivået $x_{k,l}$ sammenfaller med de to normalfordelingene som modellerer lydklassene. Dette målet er funnet i [2] til å fungere ganske godt, især ved transientlyder som er til dels kraftigere enn det typiske talenivået. Dette er stort sett tilfellet ved de transientlydene som forårsaker problemer i lydnivåutjevningssammenheng, da det er de transientlydene med høyt energinivå som får AGC til å øke dempningen. Derfor er det ikke gjort noen endringer i karakteristiske egenskaper for transientdeteksjon, og målet angitt i (2.11) benyttes som eneste egenskap for transientdeteksjon.

Selve transientdeteksjonsavgjørelsen foregår ved å klassifisere rammen som transient om f_k overstiger en gitt terskelverdi, t_{tr}. Det er i [2] benyttet en terskelverdi som er basert på sannsynligheten for tilstedeværelse av transient i signalet. Nærmere forklart finnes t_{tr} ved, for hver testfil, å estimere den kumulative fordelingsfunksjonen (eng. *Cumulative Distribution Function, CDF*) for transientmålet. Terskelverdien t_{tr} settes så slik at en persentil klassifiseres som ikke-transienter. Denne persentilen er i [2] typisk den 98-ende persentilen, altså antaes det at de 2 % høyeste verdiene av transientmålet tilhører rammer som inneholder transientlyd. Denne antagelsen baserer seg tungt på at transientandelen i hver lydfil ligger på omtrent 2 %. Denne grenseverdien passet for så vidt godt til de testdata som ble benyttet i [2], men en fastkodet grense for transientandelen i signalet er ingen god og fleksibel løsning. I tillegg er det ikke sanntidsvennlig å estimere en kumulativ fordelingsfunksjon i etterkant av hver testfils analyse, da dette krever tilgang til hele signalet før terskelen kan bestemmes.

Det er gjort forsøk med å beregne foreløpig, kumulativ fordelingsfunksjon for hver ramme, for deretter å følge prosedyren som beskrevet over, men dette er meget ressurskrevende, samtidig som det ikke utgjør noen forskjell

med tanke på antagelsen om en fast transientandel. Empirisk transientandel er ikke nødvendigvis en god prediktor for fremtidig transientandel. Denne metoden er derfor forkastet, og det er i testene i stedet benyttet et sett med terskelverdier $\mathbf{t}_{\text{tr}} = [t_{\text{tr, 1}}, t_{\text{tr, 2}}, \ldots, t_{\text{tr, B-1}}, t_{\text{tr, B}}]$, der B er antall terskelverdier som skal undersøkes. Ved å benytte en slik terskelvektor med forskjellige terskelverdier, ser en følgene endret terskelverdi har på transientdeteksjonen. Dette er illustrert i operasjonskarakteristikk-plott i avsnitt 4.3. Denne metoden er heller ikke en ideell løsning, men den er vesentlig mindre ressurskrevende i fravær av bedre løsninger.

Et annet problem som har sitt opphav i transientlydenes uforutsigbare karakter, er bestemmelse av lydenes varighet ved deteksjon. Ettersom transientlydene oftest kjennetegnes av en energirik ansats, er initiell deteksjonen av transientlydene relativt enkelt. Etterhvert som energinivået synker, vil det være mindre som skiller en transientlyd fra andre lyder med tilsvarende energinivå. Transientlydene vil altså på et tidspunkt i sitt forløp sannsynligvis passe godt til én eller begge modellkomponentene, noe som medfører at transientmålet vil ha en lav verdi, og deler av transientlyden vil klassifiseres som enten tale eller ikke-tale. For å gi en entydig merking av transientlydene som er noenlunde dekkende for reell transientlyd-varighet, er det valgt å merke et fast antall fremtidige rammer som transientlyd ved initiell transientdeteksjon.

Hvor mange rammer som skal merkes, avhenger av de mulige transientlydenes varighet. Ettersom snittlengden på de transientene som er benyttet i testdataene er ca. 480 ms og rammesteget er 10 ms, er det sagt at 47 etterfølgende rammer etter initiell transientdeteksjon skal merkes som transientlyd (totalt 48 rammer merket). Den gjennomsnittlige transientlydlengden beregnes automatisk fra transientdatabasen og gjøres tilgjengelig for transientdeteksjonsrutinen. Å benytte en slik fastkodet transientlengde er ikke en ideell løsning, men ettersom det i lydnivåutjevningssammenheng er det initielle energispranget som forårsaker problemer, er det ikke lagt mye arbeid i å finne alternative metoder for bestemmelse av transientlydlengde.

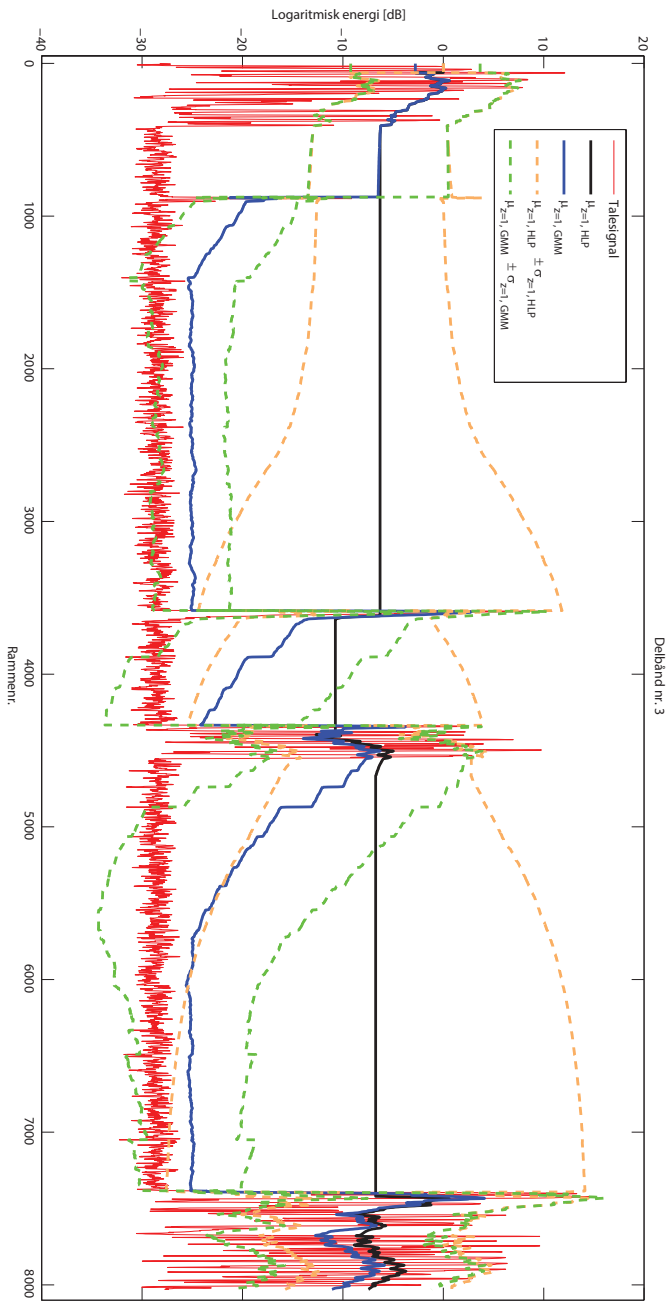
Fastkodet transientlydlengde har imidlertid noen praktiske utfordringer knyttet til seg, blant annet vil ikke transientmerkingens lengde i testfilenes fasit nødvendigvis stemme overens med den merkingen som transientdetek-

sjonsalgoritmen har produsert. I fasiten er det de faktiske transientlydene som er markert på grunnlag av transientlydens egen varighet. Ettersom det ikke er så stor variasjon i transientlydenes lengde i testdataene (estimert standardavvik på ca. 86 ms), vil ikke dette medføre de helt store, systematiske feilene, men det medfører en økt grad av unøyaktighet rundt deteksjonsresultatene. Mer om dette i avsnitt 3.2.4

2.3.2 Frysing av modellparametre

Det at talekomponentens middelverdi driver mot ikke-tale-komponentens middelverdi ved fravær av taleaktivitet løses i [1] ved å innføre en begrensning i hvor nært talekomponenten kan komme ikke-tale-komponenten, jf. (2.8c). Denne metoden fungerer ganske bra for taledeteksjon, men ettersom talekomponentens middelverdi etter en lengre periode uten reelt talepåtrykk nærmer seg ikke-tale-komponenten (omtales i denne rapporten også som *driving*, illustrert i figur 2.3), vil det ved reelt talepåtrykk være stor fare for transientdeteksjon. Dette skyldes at påtrykte talesignalets log-energinivå ofte vil ligge godt i utkant av begge modellkomponentene, og følgelig vil algoritmen komme frem til at det er lav sannsynlighet for at rammen karakteriseres av en av modellkomponentene. Dette resulterer i at transientdeteksjonen, som baserer seg på disse sannsynlighetene, kan klassifisere talerammen som en transientlyd.

For å forhindre slik uønsket feildeteksjon, foreslåes det å ikke oppdatere middelverdiparameteren for talekomponenten i de rammene som det antaes ikke inneholder taleaktivitet. Det er vurdert flere metoder for hvordan denne oppdateringsstoppen burde implementeres. For å kunne bestemme når modellparametrene for talekomponenten skal fryses, behøves det en indikator som kan si noe om tilstedeværelse av tale i signalet. Det trengs i prinsippet altså en taledetektor i taledetektoren. Denne tale-indikatoren trenger imidlertid ikke å være spesielt presis, da det ikke er strengt nødvendig at talemodellen oppdateres for hver lydramme med taleinnhold. Det er derfor undersøkt tre forskjellige metoder for å indikere når inngangssignalet ikke inneholder en talekomponent. Figur 2.3 viser effekten av driving på talekomponentens middelverdi.



Figur 2.3: Testfil på ca. 80 s der effektene av drivingsproblematikken og kontinuerlig (GMM) vs. periodevis (HLP) oppdatering av middelverdi for signalmodellens talekomponent vises.

Kullback-Leibler-divergens

Én av de metodene som er undersøkt, er å se på *Kullback-Leibler-divergens* (*KL-divergens*) [15] mellom de to normalfordelingene som modellerer ikke-tale- og talekomponenten i inngangssignalet. KL-divergens gir et mål på forskjellen mellom de to normalfordelingene, og det var tenkt å benytte dette målet for å stoppe oppdateringen når talekomponenten hadde nærmet seg entydig ikke-tale-komponenten over et visst antall lydrammer. Det ble forsøk implementert en metode for å beregne symmetrisk Kullback--Leibler-divergens for multivariate normalfordelinger. Ensidig Kullback--Leibler-divergens er gitt ved (2.12).

$$\begin{aligned} D_{\mathrm{KL}}\left(\mathcal{N}_0 \| \mathcal{N}_1\right) = \frac{1}{2}\Big[\ln \frac{|\boldsymbol{\Sigma_1}|}{|\boldsymbol{\Sigma_0}|} + \mathrm{Tr}\left[\boldsymbol{\Sigma_1^{-1}}\boldsymbol{\Sigma_0}\right] - d \\ + \left(\mu_{\mathbf{0}} - \mu_{\mathbf{1}}\right)^{\intercal} \boldsymbol{\Sigma_1^{-1}} \left(\mu_{\mathbf{0}} - \mu_{\mathbf{1}}\right) \Big], \end{aligned} \tag{2.12}$$

der $\mathcal{N}_0$ og $\mathcal{N}_1$ er normalfordelingene som representerer henholdsvis ikke-tale- og talekomponenten, $\boldsymbol{\Sigma_{0;1}}$ er de diagonale kovariansmatrisene $\boldsymbol{\Sigma_{0;1}} = \mathrm{diag}\left(\sigma^2_{(0;1),1}, \sigma^2_{(0;1),2}, \ldots, \sigma^2_{(0;1),d-1}, \sigma^2_{(0;1),d}\right)$, d er dimensjonen til kovariansmatrisen (her antall delbånd, N) og $\mu_{\mathbf{0;1}}$ er modellkomponentenes middelverdivektor $\mu_{\mathbf{0;1}} = \left[\mu_{(0;1),1}, \mu_{(0;1),2}, \ldots, \mu_{(0;1),d-1}, \mu_{(0;1),d}\right]^{\intercal}$.

Det beregnes ensidig KL-divergens $D_{\mathrm{KL}}\left(\mathcal{N}_0 \| \mathcal{N}_1\right)$ og $D_{\mathrm{KL}}\left(\mathcal{N}_1 \| \mathcal{N}_0\right)$, og den symmetriske Kullback-Leibler-divergensen finnes ved (2.13).

$$D_{\leq \mathrm{KL}}\left(\mathcal{N}_0 \| \mathcal{N}_1\right) = \frac{1}{2}\left[D_{\mathrm{KL}}\left(\mathcal{N}_0 \| \mathcal{N}_1\right) + D_{\mathrm{KL}}\left(\mathcal{N}_1 \| \mathcal{N}_0\right)\right]. \tag{2.13}$$

Innledende tester viste imidlertid at KL-divergens ikke gir noen god indikator på hvordan komponentfordelingene beveger seg. Hva de dårlige resultatene skyldes, har det ikke lyktes undertegnede å finne ut av. Resultatene var såpass dårlige at det ble bestemt å prøve ut andre metoder.

Klassesannsynlighet

Ved å undersøke modellparametrene som allerede er tilgjengelige i algoritmen, kom det frem at a priori-sannsynligheten for talekomponenten i ramme k og delbånd l, $w_{k,l,z=1}$, gir en ganske god indikasjon på hvorvidt det den aktuelle rammen inneholder taleaktivitet eller ikke. Klassesannsynligheten sier noe om empirisk fordeling av de to lydklassene (tale/ikke-tale), og ettersom $w_{k,l,z=1}$ oppdateres med en $1-\alpha$ andel nye observasjoner for hver ramme, vil $w_{k,l,z=1}$ kunne predikere hvorvidt inngangssignalet inneholder tale eller ikke. Imidlertid tar det litt tid før $w_{k,l,z=1}$ reflekterer endringer i inngangssignalet. Dette skyldes den nevnte glemselsfaktoren α og dennes innvirkning på modelloppdateringen. Eksempelvis vil det etter 10 s være ca. 28 % av empirien som har sitt opphav i observasjoner før ti-sekundersperioden start[1].

Det at det er en liten forsinkelse i $w_{k,l,z=1}$ ved relativt varig endring i inngangssignalet, trenger ikke være noen ulempe i denne sammenhengen. Da det ikke nødvendigvis er ønskelig å sette for stramme tøyler på når modellen skal oppdateres, kan det være greit at det gies litt albuerom i overgangene mellom tale og ikke-tale. Det er langtidseffektene av kontinuerlig modelloppdatering uten talepåtrykk som ønskes unngått, så hvorvidt oppdatering av begge modellkomponentene skjer i kortere perioder uten talepåtrykk eller ikke, spiller ikke så stor rolle som ved lengre tids fravær av talepåtrykk.

For å avgjøre når talemiddelverdien i modellen skal fryses og når den skal oppdateres, foreslåes det benyttet alle delbånds a priori-tale-sannsynlighet $w_{k+1,l,1}$ for ramme $k+1$. Disse sannsynlighetene vektes og summeres over alle delbånd l for å gi en kvantitativ indikator, ws_{k+1}, på om gjeldende ramme er en taleramme. Denne *taleindikatoren* er gitt ved (2.14) og delbåndsvektene i $\mathbf{u} = [u_1, \ldots, u_N]$, der N er antall delbånd som er benyttet.

[1] Andel «gamle» observasjoner R_g i $w_{k,l,z=1}$ etter en tid T finnes ved $R_g = \exp\left(\frac{T \ln \alpha}{L_{steg}}\right)$, der L_{steg} er rammeskiftlengden i sekunder og α glemselsfaktoren. Etter 30 s vil det kun være ca. 2,3 % av $w_{k,l,z=1}$ som kan tilskrives gamle observasjoner.

$$ws_{k+1} = \sum_{l=1}^{N} u_l w_{k+1,l}. \tag{2.14}$$

For de rammene der indikatoren faller under en gitt terskelverdi, oppdateres ikke talekomponentens middelverdi. Alle andre modellparametre oppdateres imidlertid som vanlig. Det er forsøkt å også fryse varians $\kappa_{k,l,1}$ og klassesannsynlighet $w_{k,l,1}$, men dette har hatt liten innvirkning på drivingsproblematikken, og bidrar kun negativt til deteksjonsresultatene. Vektingen de forskjellige delbåndenes $w_{k+1,l,1}$ gies, er satt slik at frekvensområdet $\sim$200–1500 Hz vektes tyngst[2], da dette området er der hovedtyngden av taleenergien normalt ligger. Det er ikke forsøkt med andre vektinger, dette for å begrense antallet test-variabler. Terskelverdien som ws_{k+1} måles opp imot er satt ved å manuelt undersøke ws med kjennskap til hvor talesegmentene i testfilen opptrer. Det er forsøkt med noen forskjellige terskelverdier, og endt med en terskelverdi på $SPF = 0,25$ (*SPF: Speech Probability Floor*). Denne terskelen er ikke veldig sensibel for små endringer ($\pm$ 5 %), og det er funnet at 0,25 er en verdi som gir ønsket effekt mtp. oppdateringstopp ved lang tids fravær av talepåtrykk. Et utvalg andre verdier er undersøkt og vist i kapittel 4.

Betingelsen for at talemiddelverdien $\mu_{k+1,l,1}$ skal oppdateres for ramme $k+1$ kan dermed oppsummeres i ligning (2.15). $\lambda_{k+1,l}$ oppdateres om denne betingelsen er oppfylt.

$$ws_{k+1} > SPF. \tag{2.15}$$

Se for øvrig avsnitt 2.2.4 for utfyllende beskrivelse av modelloppdateringen.

Alternativ taleindikator

Ettersom det i praksis er en taleaktivitetsdetektor som er etterspurt for modelloppdateringsbetingelsen, og lydrammene allerede blir klassifisert av

[2] Vektingsvektoren som benyttes er $\mathbf{u} = [10, 40, 50, 45, 35, 15, 5, 5]$, som svarer til delbåndene {0–188; 188–427; 427–730; 730–1114; 1114–1601; 1601–2220; 2220–3004; 3004–4000} Hz

taledetektoren, er det nærliggende å benytte deteksjonsavgjørelsen direkte til dette formålet. I praksis er dette implementert ved å sette taleindikatoren ws_{k+1} slik at den garantert er større enn SPF for de rammene som klassifiseres som tale av taledetektoren. Tilsvarende settes ws_{k+1} til null om rammen klassifiseres som ikke-tale. Oppdateringsbetingelsen er dermed som gitt i (2.15), men bestemmes for denne betingelsesmetoden av uttrykket oppsummert i (2.16) i stedet for (2.14).

$$ws_{k+1} = \begin{cases} 1000 & \text{hvis } k+1 \text{ er tale} \\ 0 & \text{hvis } k+1 \text{ er ikke-tale} \end{cases} \tag{2.16}$$

Om oppdateringsbetingelsen som er beskrevet i (2.16) benyttes, omtales dette som at *(tale-)deteksjonsavgjørelsen* eller *VAD-avgjørelsen* benyttes.

Dobbelt parametersett

Ved å stoppe oppdateringen av enkelte modellparametre, påvirkes beregningen av taleindikatoren som igjen skal styre oppdateringsstopp eller ikke. Dette er en uheldig tilbakekobling, og noe en helst vil unngå, da det reduserer forutsigbarhet og robusthet for algoritmen. Problemet er foreslått løst ved å operere med to sett modellparametre: ett der oppdateringen går kontinuerlig for hver ramme (benevnes $\lambda_{k+1,\text{kont.}}$) og ett der oppdateringen av talekomponentens middelverdi $\mu_{k+1,l,1}$ stoppes etter en tids fravær av tale (benevnes $\lambda_{k+1,l,\text{bet.}}$). Beregning av ws_{k+1} gjøres på grunnlag av parametersettet $\lambda_{k+1,l,\text{kont.}}$, mens oppdateringsstopp og modellfrys kun påvirker $\lambda_{k+1,l,\text{bet.}}$. Ettersom modell-driving kun medfører problemer ved transientdeteksjon (for taledeteksjon løser parametersettbegrensningen i (2.8c) eventuelle problemer), er det transientdeteksjonen som presterer dårlig ved lengre perioder uten talepåtrykk. Metoden som er foreslått her medfører at algoritmen gir gode resultater både for taledeteksjon basert på $\lambda_{k+1,l,\text{kont.}}$ og transientdeteksjon på bakgrunn av $\lambda_{k+1,l,\text{bet.}}$. Det er gjort forsøk med både ett og to parametersett, for å kunne observere innvirkningen dette har på systemet.

Oppdateringsprosedyre

Oppdateringsrutinen slik den er beskrevet i avsnitt 2.2.4 er i utgangspunktet gjeldende for begge parametersettene som benyttes i HLP EVAD, $\lambda_{k,l,\text{kont.}}$ og $\lambda_{k,l,\text{bet.}}$ (henholdsvis *kontinuerlig* og *betinget* oppdatering). Den eneste forskjellen i oppdateringsrutinen mellom de to, er den at talemiddelverdien $\mu_{k+1,l,1,\text{bet.}}$ i $\lambda_{k,l,\text{bet.}}$ oppdateres betinget av (2.15). Om betingelsen i (2.15) ikke holder, oppdateres alle andre modellparametre i $\lambda_{k,l,\text{bet.}}$, unntatt $\mu_{k+1,l,1,\text{bet.}}$. $\lambda_{k,l,\text{kont.}}$ oppdateres betingelsesløst for hver eneste ramme.

Selve modelloppdateringen i HLP EVAD foregår én gang per ramme, og gjøres for ingen eller alle av delbåndene. Det er en nødvendighet at algoritmen har mulighet til å analysere alle delbåndene i en ramme før oppdateringsavgjørelsen gjøres, da taleindikatoren (2.14) som styrer oppdateringsbetingelsen (2.15) ser på alle delbånd samlet. Dette er også tilfellet om taledeteksjonsavgjørelsen benyttes som oppdateringsbetingelse for modellparametrene, da deteksjonsavgjørelsen taes på grunnlag av alle delbåndsavgjørelsene. For GMM EVAD oppdateres modellparametrene for hver ramme.

2.3.3 Endringer i energi-glatting

Det er ønskelig å jevne ut de største inter-rammevariasjonene i logaritmisk energinivå (som beskrevet i avsnitt 2.2.1). I [1] er det beskrevet at dette gjøres med et fempunkts *mediumfilter* («. . . is smoothed by using a five-point medium filter . . . »). Dette er ikke det samme som et *medianfilter*, som en kan, ut ifra likhet i ordlyd, forledes til å tro er blitt benyttet. Akkurat hva et *mediumfilter* er, er ikke definert i [1], men i deres MATLAB-implementasjon kommer det frem at det er brukt et fempunkts, glidende aritmetisk middel, sentrert om midten. Dette medfører at algoritmen må ha tilgang til to fremtidige rammer, noe som innfører en sanntidsforsinkelse på 20 ms. Da det er forsøkt å gjøre implementasjonen så sanntidsvennlig som mulig, er denne glattingsmetoden byttet ut med et liknende fempunkts, glidende aritmetisk middel, men nå er det de fem siste tilgjengelige rammene som benyttes, $\bar{x}_{k,l}, \ldots, \bar{x}_{k-4,l}$, der $\bar{x}_{k,l}$ er det (ikke-glattede) logaritmiske

energinivået i ramme k og delbånd l. Altså innføres det ingen forsinkelse i denne delen av systemet. Denne endringen er gjort i både GMM EVAD og den foreslåtte HLP EVAD.

Det glattede, logaritmiske energinivået $x_{k,l}$ i ramme k og delbånd l blir altså da som gitt i (2.17).

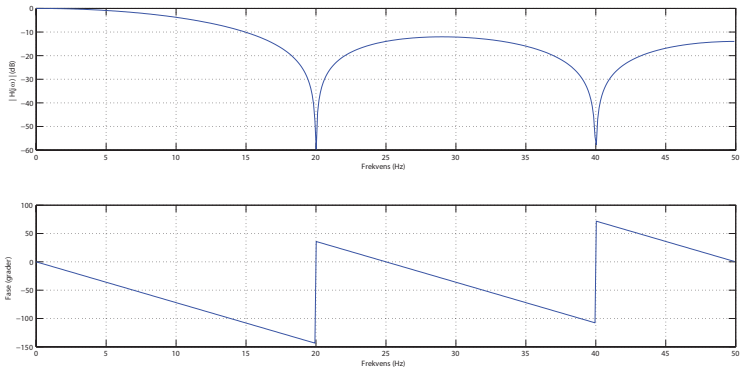
$$x_{k,l} = \frac{1}{\min\{5, k+1\}} \sum_{j=0}^{\min\{k,4\}} \bar{x}_{k-j,l}, \tag{2.17}$$

der $\bar{x}_{k,l}$ er det ikke-glattede, logaritmiske energinivået i ramme k og delbånd l. For de fire første rammene som påtrykkes systemet er det valgt å benytte det aritmetiske gjennomsnittet av de tilgjengelige rammene, da det ikke finnes tilstrekkelig antall tidligere rammer.

Det er ikke nødvendigvis ideelt å benytte et slikt glidende gjennomsnitt, blant annet fordi utsnittsvinduet (de fem rammene) ikke er sentrert om midten (altså to fremover- og to bakover-rammer) og utsnittsvinduet er rektangulært. Dette medfører enkelte signalforvrengninger, men det er ikke bedømt til å være nevneverdig problematisk i den aktuelle bruken. I dette arbeidet er det den relative forskjellen mellom de tre taledetektorene som er interessant, især forskjellen mellom GMM EVAD og HLP EVAD. Dermed er det viktigst at testgrunnlaget er så likt som mulig, slik at forskjellene som skyldes de endringene som er foreslått og implementert kommer tydelig frem. Ettersom glattingen gjøres på samme måte for både GMM EVAD og HLP EVAD, vil eventuelle signalforvrengniger påvirke de to metodene i samme grad, og dermed vil det ikke få følger for sammenligningen av de to.

Det glidende, aritmetiske gjennomsnittet som er benyttet kan uttrykkes som et filter med overføringsfunksjon gitt i (2.18), og frekvensresponsen til dette filteret er vist i figur 2.4. Punktprøvingsfrekvensen vil være 100 Hz, da rammestegslengden er 10 ms.

$$H(\jmath\omega) = \frac{1 + e^{-\jmath\omega} + e^{-\jmath 2\omega} + e^{-\jmath 3\omega} + e^{-\jmath 4\omega}}{5} \tag{2.18}$$



Figur 2.4: Frekvensrespons for glattingsfilteret som er benyttet.

Kapittel 3

Testoppsett

3.1 Testdata

For å kunne undersøke tale- og transientdeteksjonssystemene som denne rapporten omhandler på en god måte, er gode testdata avgjørende. Hva som er gode testdata vil variere, og vil aldri være noen absolutt sannhet. Talefilene som påtrykkes systemet bør forsøke å gjenskape en reell brukssituasjon for systemet på en så god som mulig måte. Samtidig er det i dette arbeidet fokusert på å løse problemet med at signalmodellen i praksis ødelegges etter lengre perioder uten tale i inngangssignalet. Samlet gir dette noen føringer for utvikling av et godt sett med testdata. Typiske brukssituasjoner er gjerne møtevirksomhet i et dedikert møterom eller kontor – omgivelser som er rimelig kontrollerte, stille og lite plaget av eksterne støykilder. Variasjon i lydnivå kan forkomme, både som resultat av møtedeltageres forskjellige avstand til mikrofon og naturlig variasjon i brukeres lydnivå på ytret tale. Slike lyddata hadde ikke vært noe problem å gjøre opptak av selv, men disse opptakene må også merkes manuelt på punktprøvenivå. Dette er meget tidkrevende arbeid, og skal korpuset ha et visst omfang, vil merkingsarbeidet gå langt utenfor tidsrammene til dette arbeidet.

3.1.1 Talesegmenter

På 80-tallet lagde imidlertid *Texas Instruments (TI)*, *Massachusetts Institute of Technology (MIT)* og *SRI International* et slikt talekorpus på bestilling av *Defence Advanced Research Projects Agency, DARPA*, det amerikanske Forsvarets forskningsenhet. *TIMIT*-databasen [16] er et talekorpus som er utviklet for å benyttes i forskning på taleteknologi og lingvistikk, og passer godt til de kravene dette arbeidet setter for et slikt korpus. Se [16] for utfyllende dokumentasjon av arbeidet som er lagt ned i innsamlingen av dette korpuset.

Det er benyttet taledata fra TIMIT-databasen, som er manuelt fonemmerket på punktprøvenivå. Det er dermed rimelig enkelt å velge ut de merkene som representerer ikke-tale, og dermed ha merking som indikerer tale/ikke-tale. TIMIT-etikettene `h#`, `pau`, `epi` og `sil` tolkes som ikke-tale/stillhet . Valget av disse merkene er basert på [17], men i en noe strengere variant i favør tale. Ikke-tale-segmentene som er interessante å merke i dette tilfellet er i hovedsak de lengre pausene i start og slutt av opptakene, samt lengre pauser mellom fullstendige ord og setninger. Dermed er ikke korte stillhetsperioder (ofte kun noen få punktprøver) initielt eller avslutningsvis i ord særlig interessant, og ikke utgjør de noen nevneverdig forskjell på resultatene, ettersom det er snakk om såpass få punktprøver sammenlignet med analyseramme-lengden.

TIMIT-filene er i utgangspunktet produsert med en punktprøvingsrate på 16 kHz, men for at de aktuelle taledeteksjonsalgoritmene skal sammenlignes på likt grunnlag, reduseres punktprøvingsraten til 8 kHz. Dette er den punktprøvingsraten taledetektoren fra ITU er designet for, og dermed dikterer denne algoritmen punktprøvingsrate på 8 kHz for alle detektorene.

3.1.2 Stillhetsperioder

For å inkludere lengre perioder uten talepåtrykk i testdataene, er det laget egne lydfiler med lengre perioder uten tale. Disse filene består av et antall talefiler fra TIMIT-databasen som er satt sammen med pausesegmenter bestående av absolutt stillhet. Strukturen på hvordan stillhetsperioder og

talefiler settes sammen, både hvor mange av hver type som skal settes sammen og i hvilken rekkefølge dette skal skje, bestemmes av bruker i MATLAB-rutinen for å generere testfiler. Stillhetsperiodenes lengde bestemmes tilfeldig innenfor et intervall gitt av bruker. Dette intervallet er satt til 3–30 sekunder i testene som er utført. Den absolutte stillheten som er tillagt blir korrumpert med støy, som beskrevet i avsnitt 3.1.4.

3.1.3 Transientlyder

Det siste spesielle signalegenskapen som er nødvendig for å få testet transientdeteksjon i de GMM-baserte taledetektorene, er åpenbart at signalet er nødt til å inneholde transientlyder. Ved utvelgelse av transientlyder til testfilene står en noe friere enn for talefiler, ettersom transientfilene ikke trenger å være merket – gitt at de kun inneholder én lyd og at denne varer fra lydfilens start til slutt. Det er valgt å legge til grunn at transientlydens lengde er bestemt av lydfilens totale lengde, selv om den energirike delen av lydfilen opptrer kun i begynnelsen av filen (se også avsnitt 1.3. Transientlydene som skaper problemer for lydnivåutjevningen er ikke entydig definert, men karakteriseres stort sett av energirike lyder med kort varighet.

For å simulere transientlyder er det benyttet et utvalg lyder fra lydscenedatabasen *Real World Computing Partnership Sound Scene Database in Real Acoustic Environments (RWCP-SSD)* [18]. Dette er et lydkorpus som er utarbeidet av forskere ved en rekke japanske institusjoner for bruk i forskning og utvikling der naturlige akustiske omgivelser er en viktig del av arbeidet. Databasen består av 105 lyder tatt opp i et ekkofritt rom, noe som passer dette arbeidets bruk bra, da det i hovedsak er den initielle ansatsen i lydene som er interessant for transientdeteksjon. For å ha et variert utvalg transientlyder tilgjengelig, er det manuelt valgt ut de 7 lydene i databasen som ansees å være de lydene som best representerer noen av de transientlydene som kan tenkes å opptre i en videokonferansesituasjon. De utvalgte lydene er presentert i tabell 3.1.

Med tilgang til tre lydkomponenter (talefiler, transientfiler og vilkårlige stillhetslengder) er testfilene blitt generert. Det er skrevet et eget MATLAB-program for generering av slike filer. Denne rutinen er laget slik at flere

Tabell 3.1: Oversikt over de transientlydene som er benyttet i testingen.

Navn (*org. eng.*)	**Beskrivelse**	**Lengde**
Bobleplast (*aircap*)	En boble i bobleplast sprenges.	420 ms
Plasthette (*cap1*)	Det taes en plasthette av en spray-boks.	421 ms
Porselensasjett (*china1*)	En trepinne slåes mot en porselens-asjett liggende på akustisk absorbent.	660 ms
Håndklapp (*clap1*)	To hender klappes sammen.	421 ms
Brusboks (*colacan*)	En brusboks åpnes.	500 ms
Metallplate (*cetal05*)	En trepinne slåes mot en metallplate (ca. A4-størrelse, 0,5 mm tykk) stående på akustisk absorbent.	480 ms
Treplate (*Teak3*)	En trepinne slåes mot en teakplate (ca. A5-størrelse) stående på akustisk absorbent.	440 ms
Gjennomsnitt		477 ms

parametre lett kan justeres for å produsere forskjellige testdata. Ønsket antall testfiler som skal genereres spesifiseres av bruker, samt hvor mange talefiler, stillhetsperioder og antall transientlyder hver testfil skal bestå av. Det er også mulig for bruker å angi strukturen på sammensetningen av stillhets- og taleperioder. Om ikke strukturen spesifiseres, velges denne tilfeldig, og varieres fra testfil til testfil. Det er imidlertid satt en begrensning i hvor tilfeldig tale-/stillhetsstrukturen velges, nemlig at første segment må være en talefil. Dette er gjort for å forhindre at signalmodellen skal initialiseres og bygges på et rent støysignal, da slike lange initielle stillhetsperioder ikke antaes å være naturlig forekommende i særlig grad. For stillhetsperiodenes varighet, kan både maksimal og minimal lengde defineres av bruker, så velges det en tilfeldig lengde mellom disse ytterpunktene individuelt for hver stillhetsperiode.

Transientlydene legges til de sammensatte testfilene i ønsket antall per fil. Plasseringen av disse innad i hver testfil bestemmes helt automatisk og tilfeldig, enten over stillhet eller tale, kun med den begrensningen at den aktuelle transientlyden ikke overlapper med en allerede tillagt transientlyd. Det er også mulig å bestemme en skaleringskoeffisient for hver transientlyd som skal tillegges. Skaleringskoeffisientene er begrenset til å ikke kunne resultere i at det adderte signalet havner utenfor det dynamiske området $[-1, 1]$. I tilfeller der dette forekommer, vil det beregnes maksimal tillatt skaleringskoeffisient uten at klipping oppstår[1]. Denne maksimale skaleringskoeffisienten erstatter den brukerbestemte skaleringen for gjeldende transientlyd.

Det beregnes også gjennomsnittlig lengde for alle transientlydene. Dette tallet benyttes i transientdeteksjonen som generell, fast varighet ved initiell deteksjon av transientlyd. Mer om dette i avsnitt 3.2.3.

[1] Første punktprøvetopp som overstiger maksimal tillatt amplitude finnes, og maksimal skaleringskoeffisient beregnes ved $\frac{x_{\max}-|x(n_{\max})|}{|y[n_{\max}]|}$, der $x_{\max}$ er maksimal tillatt amplitude, $|x[n_{\max}]|$ er absoluttverdien av det opprinnelige signalet før transientlyd er addert og $|y[n_{\max}]|$ er absoluttverdien av signalet etter transientlyd er tillagt med opprinnelig skaleringskoeffisient – alt for punktprøve $n_{\max}$. Dette gjentaes iterativt til det ikke lenger finnes noen punktprøver som overstiger maksimal tillatt amplitude.

3.1.4 Støy

De konkatenerte filene tillegges additiv, hvit gaussisk støy (eng. *Additive White Gaussian Noise, AWGN*) i et forsøk på å etterligne signalkarakteristikken ved stille perioder i en videokonferansesituasjon. Det er mulig andre støytyper hadde vært mer naturtro enn hvit, normalfordelt støy, men det antaes at denne støytypen modellerer en typisk reell situasjon tilfredsstillende nok for formålet i dette arbeidet. Støy legges på i hovedsak for at de kunstige stillhetsperiodene skal være mer naturtro, og ikke minst for å kunne justere signal-støy-forholdet (*SNR*) manuelt og kontrollert. Dette gjør at alle testfilene kan ha samme SNR, noe som sikrer konsistente og forutsigbare resultater.

Støy legges til rett i forkant av at hver enkelt testfil skal påtrykkes deteksjonsalgoritmene. Støysignalet som legges til er kun midlertidig tillagt, og endrer ikke testfilene varig, slik at disse kan benyttes med flere forskjellige realisasjoner av støysignalet. Støysignalet genereres med en tilfeldig-tall-generator som trekker tall etter en gaussisk fordeling (`randn` i MATLAB). Metoden for å oppnå ønsket støymengde (signal-støy-forhold) er å legge til et lite energirikt støysignal, før SNR evalueres. Om den estimerte SNR-verdien er for lav, legges det til enda et tilsvarende svakt støysignal. Denne prosessen gjentaes til signal-støy-forholdet er tilfredsstillende nært den spesifiserte verdien. I praksis stoppes iterasjonene etter at den spesifiserte SNR-verdien er passert. Som en selvfølge må det ønskede signal-støy-forholdet i utgangspunktet være lavere enn testfilens signal-støy-forhold, ettersom det ikke er involvert noen form for støyfjerning i systemet.

Estimering av signal-støy-forhold er i denne sammenhengen et kritisk punkt. Det er benyttet et verktøy for beregning av signal-støy-forhold utviklet av *NIST*[2] og beskrevet i [19]. Dette verktøyet benevnes *NIST STNR*, og er en MATLAB-implementert algoritme som tilfeldigvis baserer seg på gaussiske blandingsmodeller for taledeteksjon for å gi et rimelig godt SNR-estimat. Det beste estimatet for signal-støy-forholdet hadde vært å benytte innholdsfasiten for å avgjøre hvilke deler av signalet som er støy og hvilke som er tale, men det er ikke prioritert å implementere dette. Dette begrunnes i at variasjon av SNR og evaluering av deteksjonsalgoritmene under for-

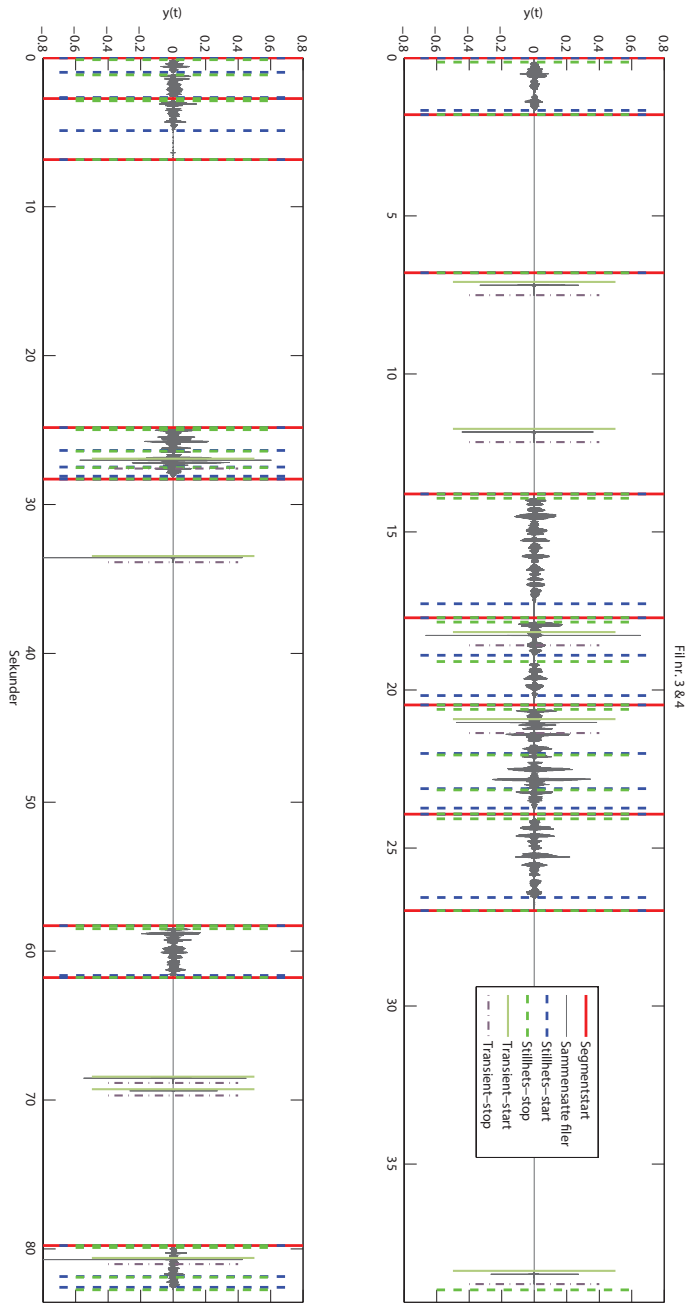
[2] *National Institute of Standards and Technology*

skjellige støyforhold kun er en begrenset del av denne oppgaven, og fordi særlig nøyaktig signal-støy-forhold ikke behøves for å undersøke generell respons på varierende støyforhold. NIST STNR er mest nøyaktig ved høy SNR [19], og i møterom er det stort sett dette som er tilfellet. Dermed er denne løsningen antatt å være god nok for dette arbeidet.

I testene som er gjort i dette arbeidet, er det benyttet 10 testfiler, hver bestående av 5 talesegmenter, 3 stillhetsperioder og 5 transientlyder. Stillhetssegmentenes varighet er tilfeldig valgt mellom 3 s og 30 s. Utvalget av transientlyder er beskrevet i tabell 3.1, og talefilene er trukket tilfeldig fra et utvalg på 4620 filer i TIMIT-korpuset. Tabell 3.2 presenterer litt statistikk om testfilene, og figur 3.1 illustrerer to testfiler med tilhørende segment-merking.

Tabell 3.2: Statistikk over testfilene som er benyttet i testene.

Samlet lengde:	636,5 s
Gjennomsnittlig lengde pr. fil:	63,6 s
Tale-andel:	20,4 %
Ikke-tale-andel:	79,6 %
Gjennomsnittlig segmentlengde:	8,0 s
Gjennomsnittlig segmentlengde, stillhet:	16,0 s
Gjennomsnittlig segmentlengde, tale:	3,1 s
Transientandel:	3,83 %
Ikke-transient-andel:	96,2 %
Gjennomsnittlig transientskaleringskoeffisient:	1,52



Figur 3.1: To testfiler med segment-merking.

3.2 MATLAB-implementasjon

3.2.1 Generering av testfiler

Metodikken for å generere lydfilene som benyttes i testingen er delvis omtalt i avsnitt 3.1, men her følger en mer utførende beskrivelse av selve MATLAB-implementasjonen som er laget for generering av testfiler.

Initielt spesifiseres det hvor mange testfiler en ønsker at skriptet skal generere, samt hvor mange talesegmenter hver fil skal bestå av. Antall stillhetsperioder i hver fil er også rimelig enkelt å justere. Det er i dette arbeidet benyttet 10 testfiler med 5 tale- og 3 pausesegmenter i hver. Deretter hentes riktig antall unike talefiler tilfeldig fra TIMIT-databasen. Tilsvarende gjøres for transientlydene, der det spesifiseres ønsket antall i hver testfil, og disse leses inn, nivånormaliseres og konverteres til den punktprøvingsfrekvensen lik den talefilene fra TIMIT-korpuset har. Det beregnes også gjennomsnittlig lengde for transientlydene, til bruk i transientdeteksjonen.

Videre angies det fra bruker en vektor som bestemmer tale-/pausestruktur (eksempelvis gir `structureOfPauses = [0, 0, 1, 0, 1, 0, 1, 0]` to initielle talesegmenter, etterfulgt av annethvert pause-/talesegment), om det er ønskelig med en fast struktur for alle testfilene. For å ha et minst mulig statisk testsett, er det innført en tilfeldig permutasjon av strukturvektoren, kun med den begrensning at det første segmentet må være et talesegment. Dette gjøres for at modellen i det hele tatt skal ha mulighet for å bli initialisert med fornuftige parametre for begge modellkomponentene. Om ikke dette gjøres, er det en viss sjanse for mange feilklassifiserte rammer før systemet påtrykkes tale, selv om oppdateringsbegrensningen (2.8c) forhindrer mye av de potensielle problemene med kun ikke-tale påtrykket systemet. Det er uansett ikke ønskelig å se på tilfeller der lengre stillhetsperioder opptrer initielt før signalmodellen er satt opp tilfredsstillende, og det er derfor valgt å ha en begrensing på initielt pausesegment.

Segmentene settes sammen slik den permuterte strukturvektoren dikterer, og start- og stopptidene for tale-/ikke-tale fra TIMIT-merkingen forskyves med en lengde tilsvarende segmentbegynnelsens forskyvning som følge av

konkateneringen. Ved å benytte denne metoden for merking av testfilene vil ikke eventuelle unøyaktigheter eller feil i TIMIT-markeringene[3] ikke forplante seg igjennom alle segmentene i testfilen. Ved å forskyve markeringstidspunktene med segmentenes plassering relativt første punktprøve i den sammensatte testfilen, vil hvert segments eventuelle unøyaktigheter forbli et lokalt problem for dette segmentet. Testfilene med markeringer er imidlertid også visuelt undersøkt, og det er ikke noe som tyder på at det er oppstått noen feil i merkingen av filene.

Så langt er det generert testfiler sammensatt av stillhetsperioder og talesegmenter. Neste steg er at bruker spesifiserer antallet transientlyder som skal legges til for hver testfil. Det er i dette arbeidet lagt på fem transientlyder for hver testfil. Det er mulig å spesifisere ønsket skaleringskoeffisient for hver av de fem transientlydene, såfremt dette ikke medfører at det dynamiske området overskrides. I slike tilfeller vil skaleringskoeffisienten reduseres automatisk slik at dette ikke skjer. Prosedyren er beskrevet i avsnitt 3.1.3. Disse skalerte lydene adderes med testfilene på tilfeldig valgte posisjoner, kun med den begrensning at de ikke får overlappe med allerede plasserte transientlyder. Skaleringskoeffisientene som er benyttet returneres sammen med testfilene fra funksjonen. Start- og stopptid for transientlydene lagres for bruk i generering av fasitmerking.

Til slutt pakkes alle testfilene i en `struct`-tabell med tilhørende, relevant informasjon om hver testfil. Det er utviklet to verktøy for å undersøke testfilene: en funksjon som plotter filene med segmentmarkeringer, og et program som beregner og viser relevant statistikk om testfilene. Begge disse funksjonene tar en filtabell med testfiler lik den som returneres fra rutinen for generering av testfiler som inngangsargument. Et plott fra visualiseringsverktøyet vises i figur 3.1, og informasjon produsert med statistikkverktøyet er listet i tabell 3.2.

[3] Markeringene i TIMIT har ved undersøkelser vist seg i noen tilfeller å ikke stemme helt overens med lydfilene, typisk ved å være forskjøvet med én punktprøve e.l. Dette medfører få praktiske konsekvenser for resultatene, men kan forårsake enkelte problemer ved indeksering i MATLAB o.l.

3.2.2 Generering av fasitvektor

For å evaluere de forskjellige tale- og transientdetektorene, må det finnes en rett klassifisering – en fasit. Med en kombinasjon av de håndmerkede TIMIT-filene, de automatisk genererte pausesegmentene samt transientlydenes plassering og varighet, har en tilgang til den informasjonen som trengs for å kunne utarbeide en slik fasit. Det er utviklet en ny metode for generering av *fasitvektor*: vektoren som, på grunnlag av start- og stoppmarkeringer generert fra TIMIT-databasen og rutinen for generering av testfiler, fungerer som fasit for rammeklassifisering. Denne forteller, på rammenivå, hva de forskjellige rammene i testfilene inneholder. Det benyttes fire lydklasser: tale (merkes `1`), ikke-tale (`0`), transient i tale (`-1`) og transient i ikke-tale (`-2`). Ettersom informasjonen som finnes om testfilenes lydinnhold er på punktprøvenivå med 16 kHz punktprøvingsfrekvens, og deteksjonsalgoritmene arbeider på rammenivå og 8 kHz punktprøvingsfrekvens, må det gjøres noen valg for hvordan innholdsinformasjonen skal transformeres fra den ene formen til den andre.

Metoden for generering av testfiler som er utviklet her, arbeider, som innholdsinformasjonen, på 16 kHz (benevnes her også *fullrate*, mens ønsket punktprøvingsrate 8 kHz benevnes som *halvrate* eller *målrate*). Ramme- og steglengde tilpasses fullrate, slik at punktprøve-lengden på disse størrelsene er konsistent med den sekund-lengden som er bestemt av bruker. Funksjonen som beregner fasitvektorene tar inngangsargumentene ramme- og steglengde, start-/stoppunkter for stillhetssegmentene, total lengde på den aktuelle testfilen, testfilens punktprøvingsfrekvens, desimeringsfaktoren som gir ønsket punktprøvingsrate og start-/stoppunkter for transientlydene, om det er noen. Med andre ord støtter metoden filer både med og uten tillagte transientlyder, samt lydfiler med annen punktprøvingsrate enn TIMITs 16 kHz.

Første steg mot en rammebasert fasitvektor er å lage en fasitvektor på punktprøvenivå ved fullrate, uten at det taes hensyn til rammeinndelingen som gjøres av deteksjonsalgoritmen. Det opprettes en nullvektor for dette formålet, og det gåes systematisk igjennom alle talesegmentene som er markert i TIMIT-databasen. For hvert talesegment markeres segmentet som tale, gitt at segmentet har varighet lenger enn en satt minsteterskel.

Denne terskelen er satt til å være en kvart steglengde (tilsvarer her 2,5 ms). Terskelen er innført for å luke ut eventuelle unøyaktigheter (typisk av noen punktprøvers varighet) som følge av sammensetningen av testfilene. I tillegg er så korte talesegmenter helt uinteressant, da de uansett aldri vil resultere i taleramme i fasit.

Nettopp hvordan innholdet til fasitvektorens rammer bestemmes, leder til neste skritt i merkingsrutinen: for hver ramme undersøkes den tilhørende punktprøvemerkingen for hvor stor andel av denne som er merket som tale. Om mer enn halvparten av rammens punktprøver er merket som tale, merkes også rammen som dette, før rammen forskyves med en steglengde og prosedyren gjentaes. Her er ramme- og steglengde justert til å korrespondere med fullrate, i dette tilfellet oppjustert med en faktor på 2, slik at den resulterende fasitvektoren passer til målraten. Rammene (både lengde og sekvensiell forskyvning) som benyttes i denne metoden korresponderer dermed nøyaktig til rammene som benyttes i deteksjonsalgoritmene.

Videre brukes samme metode for transientmerkingen som for talemerkingen: for hver ramme undersøkes det korresponderende rammesegmentet i fullrate-fasitvektoren, og om mer enn halvparten av rammens punktprøver er merket som transientlyd, merkes også rammen som transientlyd. Det benyttes her to forskjellige merkinger, én for transientlyd i tale og én for transientlyd i ikke-tale, avhengig av hva rammen allerede er merket som. Slik beholdes informasjonen om det bakenforliggende innholdet når rammen merkes som transientlyd. Dette er viktig for å kunne beregne korrekte treffrater ved evaluering av de forskjellige deteksjonsmetodene. Mer om dette i avsnitt 3.2.4

Fasitvektor, samt tilhørende merkingsforklaring leveres som utgangsdata fra rutinen.

3.2.3 Tale- og transientdetektor

Mye av tiden som er gått med i dette arbeidet, er gått til å skrive og teste en ny MATLAB-implementasjon av HLP EVAD, samt test-rammeverk. Denne taledetektoren er basert på GMM EVAD som beskrevet i [1], men det er gjort en rekke endringer som har medført at det var nødvendig å

skrive ny programkode. Virkemåte og en del av beregningene som gjøres er beholdt fra tidligere implementasjon – for eksempel gjelder dette hvordan logaritmisk energinivå beregnes. Den viktigste årsaken til at implementasjonen er gjort på nytt er for å gjøre implementasjonen sanntidsvennlig. Med dette menes det at koden med enkelhet skal kunne skrives om til å kjøre i sanntid, altså med kontinuerlig lydpåtrykk fra en lydkilde. Slik algoritmen er beskrevet i [1] og implementert i deres MATLAB-versjon, gjøres all analyse og klassifisering for alle rammene i inngangssignalet per delbånd, før prosessen gjentaes for neste delbånd osv. Dette er særdeles lite sanntidsvennlig, da algoritmen er avhengig av å ha tilgang til hele inngangssignalet før én eneste ramme kan klassifiseres.

For å gjøre taledetektoren sanntidsvennlig, er det derfor tatt utgangspunkt i å analysere og klassifisere hver enkelt ramme fortløpende, både for tale- og transientdeteksjon. Dette muliggjør at nye lydrammer kan påtrykkes detektorene fortløpende, uten særlig annen forsinkelse enn rammestegslengden og prosesseringstiden. Sanntidstilpasningen har medført et behov for endring av metoden for glatting av det logaritmiske energinivået per delbånd over flere rammer. Som nevnt i avsnitt 2.3.3, beregnes det glidende gjennomsnittet av fem sammenhengende rammer sentrert om den gjeldende rammen, noe som medfører to rammers forsinkelse. Dette er endret i implementasjonene av både GMM EVAD og HLP EVAD, slik at det nå er den gjeldende rammen og de fire foregående rammene $(\bar{x}_{k,l}, \ldots, \bar{x}_{k-4,l})$ som det beregnes et glidende, aritmetisk gjennomsnitt av.

Videre er det de $M + 1$[4] første rammene som benyttes til initialisering av signalmodellen, slik at ingen deteksjon gjøres før ramme $M + 1$. For de M første rammene beregnes det kun log-energinivået $x_{k,l}$ som modellen initialiseres med i ramme $M + 1$. De initielle modellparametrene $\lambda_{0,l,kont.}$ benyttes så for å klassifisere de $M + 1$ første rammene. De samme modellparametrene benyttes også til å initialisere modellen som benyttes for transientdeteksjon, $\lambda_{0,l,bet.}$.

For rammenummer $M + 2$ og videre er det følgende prosedyre som er gjeldende:

1. Logaritmisk energi-innhold $\bar{x}_{k+1,l}$ for alle mel-delbånd l beregnes for

[4] NB: i koden er antallet $M + 1$ bestemt av variabelen `M`, altså er `M` $= M + 1 \ldots$

rammen. Funksjonen `melbankm` fra lydbehandlingsbiblioteket VOICEBOX [13] benyttes for mel-bånds-deling av spekteret, og MATLABs innebygde `fft` brukes for å finne signalenergien gitt av (2.1).

2. For hvert delbånd utføres følgende:

   2.1. Logaritmisk energinivå glattes med metode gitt i (2.17).

   2.2. Det beregnes betinget simultansannsynlighet for klassetilhørighet, $p(x_{k+1,l}|z, \lambda_{k,l,kont.})$, gitt ved produktet av $p(z) \rightarrow w_{k,l}$ og (2.3).

   2.3. Foreslåtte oppdateringer for modellparametrene beregnes ved (2.7c), (2.7e) og (2.7g).

3. Taleindikatoren gitt av (2.14) beregnes for rammen.

4. Om $ws_{k+1} < SPF$ oppdateres kun $\lambda_{k+1,\forall l,kont.}$, mens talekomponenten $\mu_{k+1,\forall l,1,bet.}$ i $\lambda_{k+1,\forall l,bet.}$ holdes uendret. Overstiger ws_{k+1} terskelverdien, oppdateres begge parametersettene, både $\lambda_{k+1,\forall l,kont.}$ og $\lambda_{k+1,\forall l,bet.}$. Dette gjelder for alle modellparametrene i de to parametersettene.

5. For hvert delbånd gjøres så:

   5.1. Desisjonsgrense $\grave{\theta}_{k+1}$ bestemmes på grunnlag av $\lambda_{k+1,l,kont.}$.

   5.2. Om $\grave{\theta}_{k+1} < x_{k+1,l}$ klassifiseres delbånd l i ramme $k+1$ som tale.

6. Avhengig av eventuell delbåndsvekting og terskelverdi for andel vektede delbånd som må være klassifisert som tale, taes klassifiseringsavgjørelsen for rammen. Denne glattes med metoden som er omtalt i avsnitt 2.2.6. Dette er den endelige rammeavgjørelsen for taledeteksjon som ligger til grunn for resultatene i 4.2.

7. Transientmålet beregnes som gitt i (2.11), og på grunnlag av dette målet og terskelen satt for transientdeteksjon klassifiseres rammen som transient eller ikke-transient. Tidligere taleaktivitetsdeteksjon taes vare på ved å benytte fire forskjellige markeringer: tale, ikke-tale, transient i tale og transient i ikke-tale. Om rammen er bestemt av tidligere detektert transientramme til også å skulle merkes som transientramme, gjøres dette uavhengig av transientmålets verdi i

Tabell 3.3: Oversikt over mulige testutfall.

Fasit \ Klassifisering	**Tale/transient**	**Ikke-tale/ikke-transient**
Tale/transient	TP	FN
Ikke-tale/ikke-transient	FP	TN

forhold til transientterskelen. Transientavgjørelsen som taes her er grunnlaget for resultatene som er presentert for transientdeteksjon i avsnitt 4.3.

8. Modellene oppdateres som beskrevet i avsnitt 2.2.4.

3.2.4 Beregning av resultater

Med klassifiserte rammer fra de forskjellige deteksjonsmetodene og fasitvektorer for alle testfilene, ligger alt til rette for å beregne forskjellige evalueringsmål for metodene. Det beregnes fire mål: *korrekt aksept* (eng.: *True Positive, TP*), *korrekt avvisning* (eng.: *True Negative, TN*), *feilaktig aksept* (eng.: *False Positive, FP*) og *feilaktig avvisning* (eng.: *False Negative, FN*). Disse beregnes for hver av metodene for tale- og transientdeteksjon – her fire algoritmer for taledeteksjon og to for transientdeteksjon. De fire målene beregnes ved å gå igjennom alle rammene sekvensielt og telle opp hvor mange rammer som enten er feil (FP & FN) eller korrekt (TP & TN) klassifisert. Dette gjøres enkelt ved å sammenligne den aktuelle rammens klassifisering med fasitvektorens merking for korresponderende ramme. Er en tale-/transientklassifisert ramme fasitmerket som tale/transient, økes telleren for TP med én. Tilsvarende om en ikke-tale-/ikke-transientklassifisert ramme er fasitmerket som nettopp dette, er det TN som økes med én. Likeledes økes FP om en fasitmerket ikke-tale-/ikke-transient-ramme klassifiseres som tale/transient. Siste mulighet er da tale-/transient-merket ramme som feilklassifiseres som ikke-tale/ikke-transient, hvor det er FN som øker med én. De mulige utfallene er oppsummert i tabell 3.3.

Ettersom testfilene er de samme for testingen av taleaktivitetsdeteksjon og transientdeteksjon, vil det påtrykkes lydfiler som inneholder transientlyder i alle tilfeller. For evaluering av taledetektor uten transientdeteksjon må det taes en avgjørelse om hvorvidt rammene som inneholder (og er merket) transientlyd skal tolkes som tale eller ikke-tale. En rett-frem løsning på denne utfordringen vil være å se på hva testfilene inneholdt i den aktuelle rammen *før* transientlydene ble tillagt. Dette vil imidlertid ikke representere det signalinnholdet deteksjonsalgoritmen faktisk har undersøkt, noe som gjør denne metoden til en relativt urimelig og feilaktig metode. Ettersom det antaes at transientlyd-rammene har et energi-innhold som er på nivå eller høyere enn talerammene, vil det være naturlig å anta at langt de fleste transientlydrammene vil bli klassifisert som talerammer. I ren taledeteksjonssammenheng er det denne antagelsen som er lagt til grunn når alle transientmerkede rammer tolkes som om de var talerammer. Rent praktisk medfører dette altså at fasitmerkene tale (`1`), transient i tale (`-1`) og transient i ikke-tale (`-2`) alle behandles som om de skulle vært talemerker (`1`).

De endelige prestasjonsmålene som benyttes er *korrekt akseptrate* (eng.: *True Positive Rate, TPR*) og *korrekt avvisningsrate* (eng.: *True Negative Rate, TNR*). Disse beregnes som gitt i ligningene (3.1a) og (3.1b).

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}, \tag{3.1a}$$

$$TNR = \frac{TN}{N} = \frac{TN}{FP + TN}, \tag{3.1b}$$

der P er antall rammer klassifisert som tale/transient og N er antall rammer klassifisert som ikke-tale/ikke-transient, for henholdsvis tale-/transientdeteksjon. TPR og TNR beregnes for hver av de seks deteksjonsmetodene, og returneres til testrammeverket.

For evaluering av transientdeteksjonen er det også noen utfordringer. Især gjelder dette for bestemmelse av transientlydenes varighet, og hvordan treffrater skal beregnes for transientdeteksjonen. Problematikken er også omtalt i avsnitt 2.3.1. Ettersom transientlydene ofte har en karakteristisk,

men allikevel uforutsigbar, fremtoning, er det vanskelig å bestemme varigheten av en transientlyd med særlig sikkerhet. Dette er heller ikke spesielt viktig i lydnivåutjevningssammenheng, da det oftest er den store, initielle energiøkningen som skaper problemer for AGC. Som en følge av at overgangen fra transientlyd til ikke-transientlyd er noe uklar, er det bestemt at alle detekterte transientlyder har en fastsatt lengde, og detekteres én transientramme, vil et visst antall etterfølgende rammer også klassifiseres som transientlyd. Dette stemmer ikke nødvendigvis overens med hva som er det faktiske innholdet i signalet, gitt av fasitvektoren. Ikke bare kan deteksjonsalgoritmen feilklassifisere en ramme (og dermed flere etterfølgende rammer), men selv ved korrekt klassifisering fra første transientholdige ramme vil det være stor sannsynlighet for at den fasitmerkede lengden ikke stemmer overens med det antallet rammer som er blitt transientklassifiserte. Dette fører til dårligere resultater.

Det kan være et alternativ å se på andre måter å beregne evalueringsmål for transientdeteksjon, men såfremt transientlydene er så varierende som de tenderer til å være, vil det være utfordringer og uheldige konsekvenser knyttet til de fleste metoder. Det er i dette arbeidet fastholdt ved metoden med fastkodet transientlydlengde og rammevis evaluering mot fasitvektor. Selv om dette ikke er noen ideell løsning, fungerer den rimelig greit for å indikere presisjonsnivå for de to forskjellige transientdeteksjonsmetodene. Eventuelle unøyaktigheter i de forskjellige prestasjonsmålene vil antageligvis jevne seg noe ut over alle testfilene, gitt at den fastkodede transientlengden er satt noenlunde riktig i forhold til lydene i transientlyddatabasen. I testene er denne satt til det aritmetiske gjennomsnittet av lydenes varighet, og transientlydenes lengde er rimelig jevnt fordelt rundt dette snittet. Resultatene for transientdeteksjon er dermed ikke nødvendigvis direkte sammenlignbare med de tilsvarende prestasjonsmålene for taleaktivitetsdeteksjon, men relativt mellom de to forskjellige transientdeteksjonsmetodene er det grunnlag for direkte sammenligning.

3.2.5 Rammeverk

For å sy hele testrutinen sammen, er det benyttet et rammeverk som kaller de forskjellige subrutinene i testoppsettet. De fleste brukerparametre styres

fra rammeverket, og det er rammeverket som holder styr på testfilene og sender disse til deteksjonsalgoritmene én etter én. Testfilene leses inn fra en fil som er forhåndsgenerert, slik at testdataene alltid er de samme som for tidligere gjennomførte tester. Støypåvirkningen er imidlertid unik fra iterasjon til iterasjon av deteksjonsalgoritmene. Dette for at det skal være mulig å justere signal-støy-forholdet med ellers identiske testfiler, samt fordi dette realiserer støyens naturlige, kontinuerlig varierende karakteristikk best.

Deteksjonsalgoritmene initialiseres på nytt for hver iterasjon av testfil og terskelpar, da det ikke er sett hensiktsmessig å beholde modellparametrene fra forrige testfil som initial-modellparametre for den aktuelle testfilen. Modellinitialiseringen er meget rask ($<$1 s), og de eventuelle feilklassifiseringene som måtte skyldes dårlige modellparametre antaes å være så få at dette ikke vil påvirke sluttresultatet i nevneverdig grad.

Rammeverket kjører deteksjonsalgoritmene med forskjellige terskelverdier, både et sett med delbåndsterskler og et sett med terskler for transientmålet f_k. Delbåndstersklene er terskler for hvor mange taleklassifiserte delbånd som må til for at rammen som helhet skal taleklassifiseres. Disse terskelene korresponderer med antall delbånd fra ett til N. I dette arbeidet blir det dermed 8 terskler. Det er valgt å benytte et tilsvarende antall terskelverdier for transientmålet, både av bekvemlighetsgrunner (like mange som delbåndstersklene) og fordi det viser seg å være et passe antall for å få tilstrekkelig mange prestasjonsmål. Da delbåndstersklene for taledeteksjonsavgjørelsen er helt uavhengig transientdeteksjonen, har det vært mulig å kombinere de to terskelsettene for effektiv kjøring av testoppsettet. Rent praktisk byttes altså begge de to terskelverdiene ut for hver gjennomkjøring av testfil-settet. ITU VAD, Sohn VAD og rutinen for generering av fasitvektor kjøres kun én gang per testfil, da varierende terskelverdier ikke har noen innvirkning på resultatene herfra.

Resultatene fra hver eneste terskel og testfil lagres etter hver iterasjon. Det beregnes et fil-lengde-vektet snitt av resultatene fra de individuelle testfilene for hvert terskelpar. Slik blir resultatene så like som mulig de fra et sanntidssystem der alle rammer teller likt i prestasjonsevalueringen, som er det endelige målet med dette arbeidet. De endelige resultatene blir dermed, for hver deteksjonsmetode, et prestasjonsmålpar bestående

av TPR og TNR for hver av terskelparene. Resultatene er vist i kapittel 4.

Kapittel 4

Resultater

4.1 Målemetode og -enheter

Taledetektorene vurderes etter TPR (kalles også *sensitivitet* eller *tale-/transient-treffrate*) og TNR (kalles også *spesifisitet* eller treffrate for ikke-tale/ikke-transient) (se avsnitt 3.2.4). Disse målene er komplementære med henholdsvis prestasjonsmålene *feilaktig avvisningsrate* (FNR) og *feilaktig akseptrate* (FPR), slik at $TPR = 1 - FNR$ og $TNR = 1 - FPR$. Dermed er deteksjonsprestasjonene entydig beskrevet av TPR og TNR. Disse to målene plottes mot hverandre i et *operasjonskarakteristikk-plott* (eng.: *Receiver Operating Charachteristic, ROC*). Kurvene i disse plottene fåes ved å endre én algoritmeparameter: for taledeteksjon er det terskelverdien for hvor mange taleklassifiserte delbånd som må til for taleklassifisert ramme som er variert, mens det for transientdeteksjon er terskelverdien t_{tr} for transientmålet i (2.11) som justeres. Se for øvrig avsnitt 2.3.1. Det gjøres oppmerksom på at operasjonskarakteristikk-plottene ikke følger konvensjonen med å vise TPR mot FPR [20], men heller TPR mot TNR. Dette medfører at idealresultat er i øvre høyre hjørne av plottet, og ikke øvre venstre som konvensjonelt. Dette er gjort for å enkelt kunne sammenligne karakteristikkplottene med tidligere arbeider [2, 1]. Alle plottene benytter også samme utsnitt og akse-skalaer, slik at operasjonskarakteristikkplott i rapporten lett kan sammenlignes mot hverandre.

Alle taledetektorene gies lydfiler med punktprøvingsfrekvens 8 kHz, da det er denne punktprøvingsfrekvensen ITU VAD er designet for. Dette medfører et behov for å re-punktprøve TIMIT-filene ned fra 16 kHz til 8 kHz. Dette er gjort med MATLABs innebygde funksjon `decimate`, som også sørger for å lavpass-filtrere signalet med et passende filter før desimering.

Siden operasjonskarakteristikk-plottene ikke legger opp til direkte avlesning av sammenhengen mellom terskelverdier (både delbånds- og transientmål-terskler), TNR og TPR, er tersklene markert direkte i plottene. For delbåndstersklene er det markert som piler med tilhørende antall taleklassifiserte delbånd som må til for taleklassifisering av rammen, mens det for transientdeteksjonsplottene er markert tilsvarende, men da med terskelverdiene for transientmålet. Pilene er posisjonert ut ifra kurven for HLP EVAD, men GMM EVAD følger samme tendens som HLP EVAD, slik at markeringen også indikerer omtrentlig posisjon for GMM EVAD-punktene. Det er også i vedlegg B vist separate plott for TNR og TPR som funksjon av tersklene. Disse korresponderer til figurene i avsnitt 4.

Tabell 4.1 viser systemparametrene som er benyttet i testene om ikke annet er spesifisert, mens tabell 4.2 viser inngangsparametrene som er benyttet for generering av testfiler. Testfilene er holdt konstante for alle forsøkene som er gjort.

Tabell 4.1: Oversikt over standardparametre som er benyttet i testingen.

Symbol	Variabel	Verdi	Beskrivelse
N	`num_chnl`	8	Antall delbånd.
$M+1$	`M`	60	Antall rammer for initialisering.
α	`alpha`	0,99	Glemselsparameter for modelloppdatering.
δ	`theta`	3,5	Minimum forskjell mellom $\mu_{k,l,0}$ og $\mu_{k,l,1}$.
ϵ	-	8	Begrensingsparameter for modellkomponentenes innbyrdes variansforskjell (fastkodet).
γ	`deciGamma`	0,45	Bestemmer desisjonsgrensens forskyvning mot ikke-tale ($1 \rightarrow 0$).
-	`SNR`	30	Ønsket signal-støyforhold [dB].
-	`framesz`	160	Rammelengde i punktprøver @ 8 kHz.
-	`stpsz`	80	Rammestegslengde i punktprøver @ 8 kHz.
-	`voting`	[1/8 2/8 3/8 4/8 5/8 6/8 7/8 8/8]	Delbåndsterskler.
-	`weights`	[1 1 1 1 1 1 1 1]	Delbåndsvekter.
$\mathbf{t_{tr}}$	`threshVector`	[5 10 15 20 25 30 50 70]	Transientmål-terskler.
-	`VADDECIDEUPDATE`	0	Om VAD-avgjørelsen (`1`) eller ws_{k+1} (`0`) skal brukes.
-	`hlpupd`	1	Benytt dobbelt parametersett (`1`) eller ikke (`0`).
SPF	`SPF`	0,25	Oppdateringsterskel.

Tabell 4.2: Oversikt over inngangsparametre for generering av testfiler.

Symbol	Variabel	Verdi	Beskrivelse
-	`noFiles`	10	Antall testfiler.
-	`noSegmentsInEach`	5	Antall talesegmenter i hver testfil.
-	`pauseLength`	[3 30]	[*min. maks.*] tillatt lengde for pausesegmenter.
-	`structureOfPauses`	[0 0 1 0 1 0 1 0]	Bestemmer hvor mange pausesegmenter (`1`-ere) i hver testfil, samt (permuterbar) struktur for segmentene (tale = `0`).
-	`numTrans`	5	Antall transientlyder per testfil.
-	`scalingInit`	[1 1,5 2 3 3]	Ønskede skaleringer for transientlydene.

4.2 Taledeteksjon

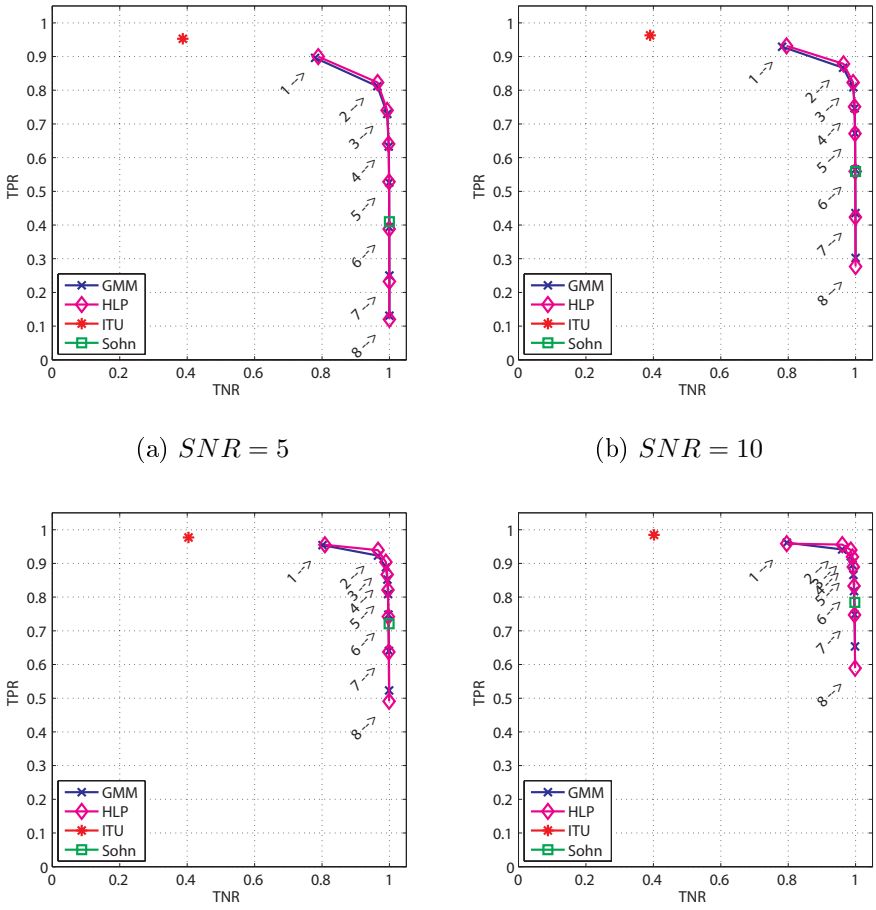
For testene som er gjennomført med taledetektorene er virkningen av forskjellige parametre undersøkt. Det er tatt utgangspunkt i et standard oppsett av systemparametre gitt i tabell 4.1, og enkelte parametre er endret for å demonstrere virkningen disse har på taledetektorene. Om ikke annet er spesifisert, er det parametrene i tabell 4.1 som er benyttet i forsøkene.

4.2.1 Signal-støy-forhold

Testfilenes signal-støy-forhold er variert for å undersøke effektene dette har på taledeteksjonsalgoritmene. Det er benyttet fire verdier for SNR: 5 dB, 10 dB, 20 dB og 30 dB. Dette er i utgangspunktet ikke et veldig bredt område av SNR-verdier, men det antaes at dette området dekker de brukssituasjonene systemet i dette arbeidet er utviklet for. Møterom bør ikke ha et særlig høyt støynivå om det skal fungere etter hensikten, og talenivået er normalt sett tilstrekkelig høyt til å gi SNR i det aktuelle området, om ikke høyere[1].

Figur 4.1 viser taledetektorenes prestasjoner i operasjonskarakteristikkplott for de forskjellige SNR-verdiene.

[1] Gjennomsnittlig støynivå i møterom skal være maksimalt 28 dB for bygg som følger [10, kl. C]. Med et normalt talenivå på mellom 60 dB ± 5 dB [21], vil dermed signal-støy-forholdet ligge rundt de testede 30 dB.



(a) $SNR = 5$ (b) $SNR = 10$

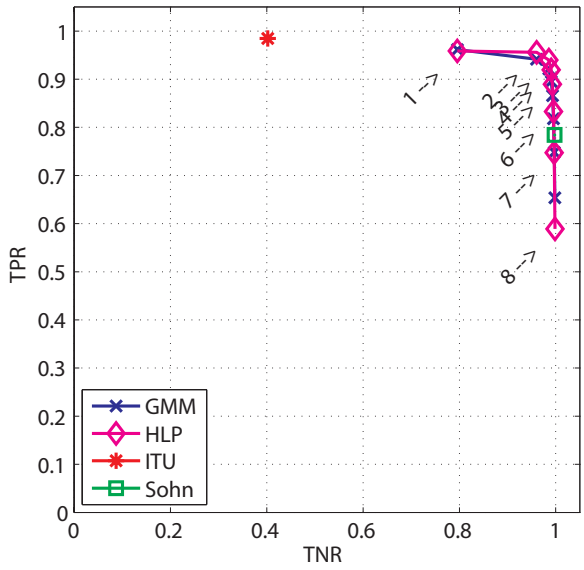
(c) $SNR = 20$ (d) $SNR = 30$

Figur 4.1: Operasjonskarakteristikk for taledeteksjon ved varierende signal-støy-forhold.

4.2.2 Desisjonsgrenseforskyvning γ

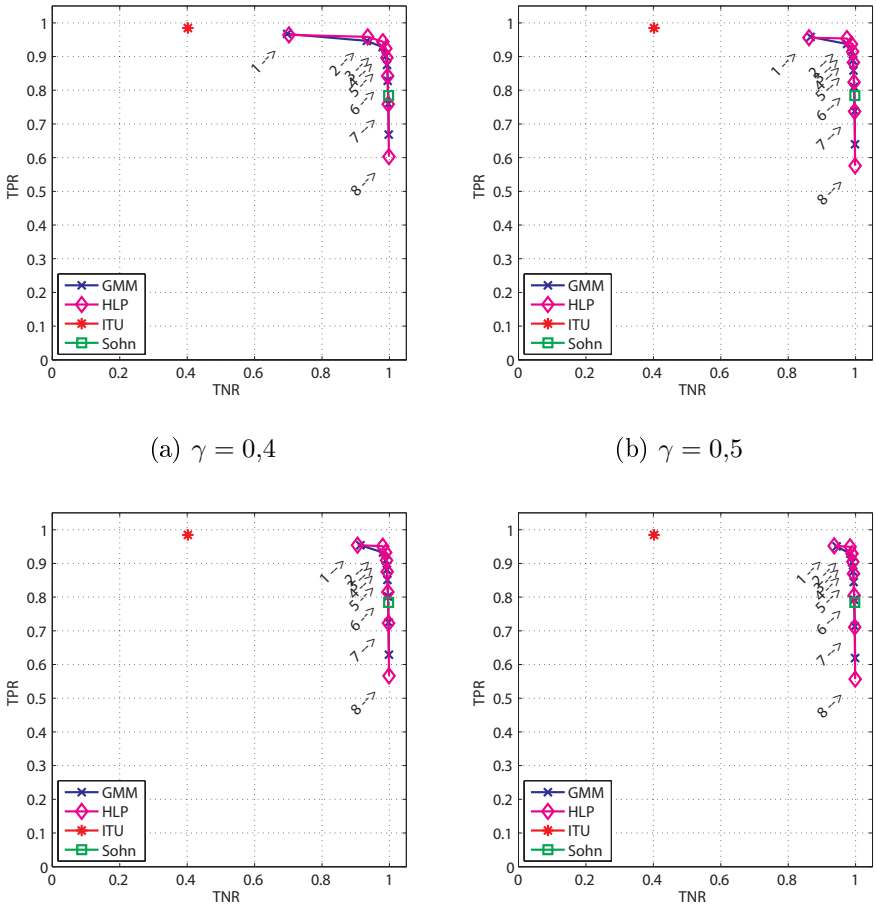
For å se effekten desisjonsgrenseforskyvning har på deteksjonsnøyaktigheten, varieres γ. Parameteren gjør det mulig å flytte desisjonsgrensen θ mer

eller mindre mot ikke-tale-komponenten, slik at taleklassifisering favoriseres på bekostning av ikke-tale. I figur 4.2 er operasjonskarakteristikken vist med standardverdien for $\gamma = 0{,}45$. Plottet gjenspeiler også systemprestasjoner ved alle parametres standardverdier, som gitt i tabell 4.1.



Figur 4.2: Operasjonskarakteristikk med standardparametre, der γ er satt til 0,45 ved $SNR = 30$.

I figur 4.3 vises operasjonskarakteristikk for fire γ-verdier rundt standardverdien, og forflytningen av kurvene fra plott til plott illustrerer parameterens betydning for deteksjonsresultatene.

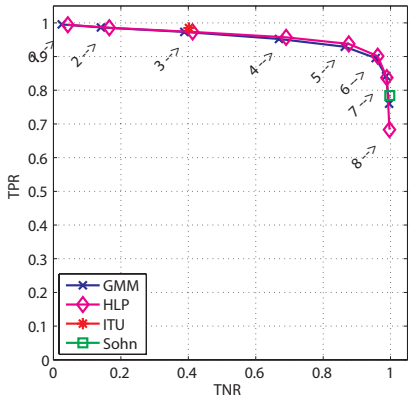


(a) $\gamma = 0{,}4$ (b) $\gamma = 0{,}5$

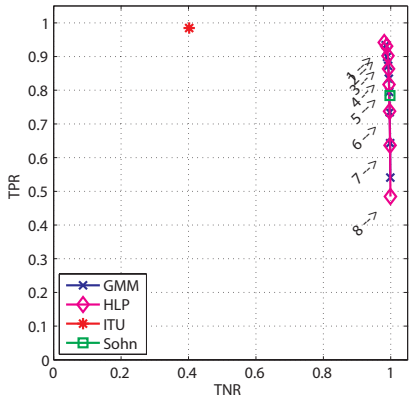
(c) $\gamma = 0{,}55$ (d) $\gamma = 0{,}6$

Figur 4.3: Alternative γ-verdier ved $SNR = 30$.

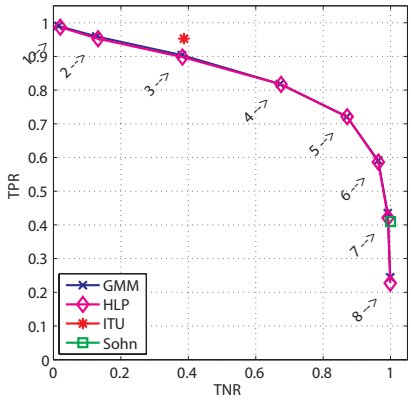
For å demonstrere effekten av ekstremverdier for $\gamma \in [0, 1]$, er det i figur 4.4 vist operasjonskarakteristikk for $\gamma = 0{,}1$ og $\gamma = 1$ (medfører $\grave{\theta} \rightarrow \theta$) ved $SNR = 30$ (figur 4.4a og 4.4b) og $SNR = 5$ (figur 4.4c og 4.4d). Figurene illustrerer betydningen γ har på resultatene relativt til signal-støyforholdet, samt effektene av stor variasjon i γ.



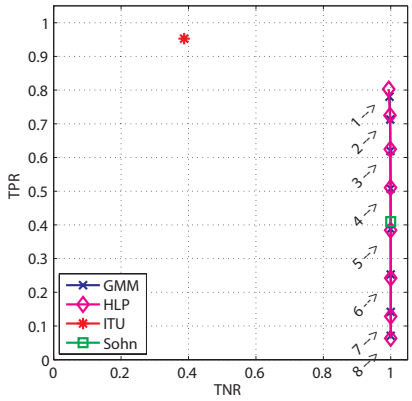
(a) $\gamma = 0{,}1$ og $SNR = 30$



(b) $\gamma = 1$ og $SNR = 30$



(c) $\gamma = 0{,}1$ og $SNR = 5$



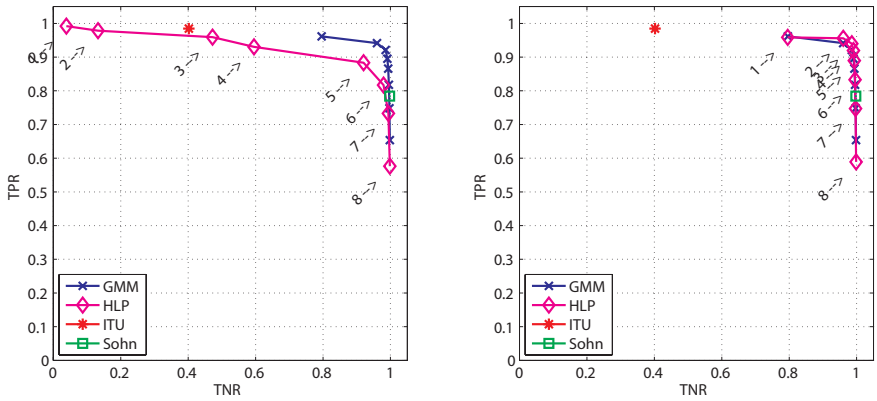
(d) $\gamma = 1$ og $SNR = 5$

Figur 4.4: Operasjonskarakteristikk for taledeteksjon ved høy/lav γ og høyt/lavt signal-støy-forhold.

4.2.3 Dobbelt parametersett

Figur 4.5 viser virkningen av å benytte dobbelt parametersett når signal-modellen fryses ved fravær av tale. Venstre figur (figur 4.5a) er operasjons-

karakteristikk med enkelt parametersett λ, mens høyre figur (figur 4.5b) viser effekten av å benytte dobbelt parametersett $\lambda_{\text{bet.}}$ (for transientdeteksjon, kan fryses) og $\lambda_{\text{kont.}}$ (for taledeteksjon, fryses aldri). Se også avsnitt 4.3.2.

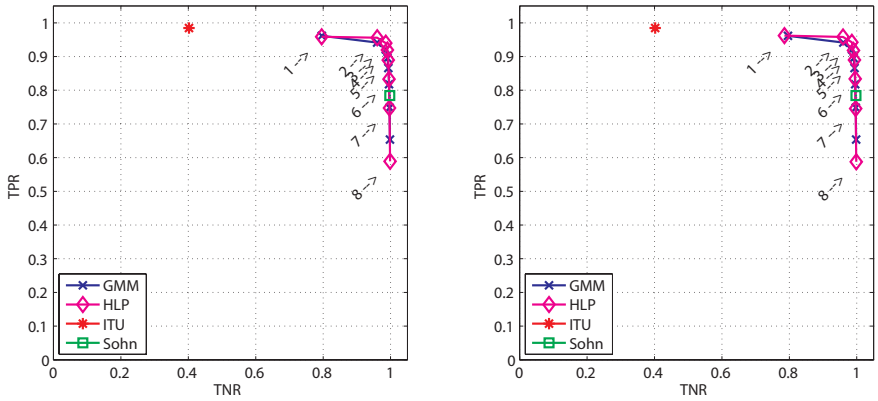


(a) `hlpupd = 0` (b) `hlpupd = 1`

Figur 4.5: Operasjonskarakteristikk for taletdeteksjon med (a) enkelt og (b) dobbelt parametersett.

4.2.4 Modelloppdateringsbetingelser

Det er i figur 4.6b vist innvirkningen det har på operasjonskarakteristikken for taledeteksjon når VAD-avgjørelsen benyttes som oppdateringsbetingelse for oppdatering av modellparametrene $\lambda_{\text{bet.}}$. Motsatt viser figur 4.6a operasjonskarakteristikk for taledeteksjon om taleindikatoren ws_{k+1} benyttes for å avgjøre hvorvidt $\lambda_{\text{bet.}}$ skal oppdateres eller ikke. For taledeteksjon skal ikke denne forskjellen bety noe som helst for resultatene, noe figur 4.6 viser. Disse resultatene er tatt med her i hovedsak for å illustrere hvor liten innvirkning oppdateringsstopp av $\lambda_{\text{bet.}}$ har på resultatene for taledeteksjon. Plottene er resultatet av å skru av og på `VADDECIDEUPDATE`.

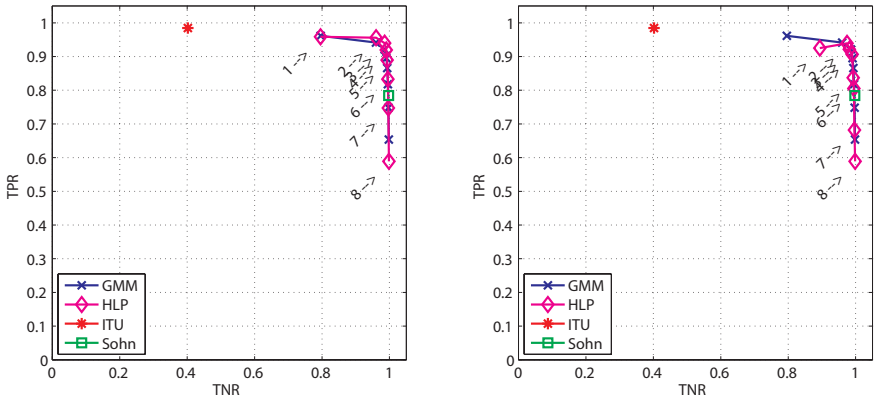


(a) Taleindikator ws_{k+1} (b) VAD-avgjørelse

Figur 4.6: Operasjonskarakteristikk for taledeteksjon med forskjellige betingelser for modellfrys.

4.2.5 Delbåndsvekting

Rammeavgjørelsen om rammen inneholder taleaktivitet eller ikke, taes på grunnlag av et vektet snitt av delbåndsavgjørelsene (styres av `weights`) og en delbåndsterskel (gitt av `voting`). Det er i figur 4.7 illustrert hvordan et likevektet og et talevektet sett med delbåndsvekter påvirker taledetektorenes prestasjoner. For likevektede delbånd, gies det like stor vekt til alle mel-frekvens-delbånd, mens det for talevektede delbånd vektes med `weights = [ 10 40 50 45 35 15 5 5 ]`, der hvert vektorelement svarer til korresponderende delbånd.



(a) Likevektede (b) Talevektede

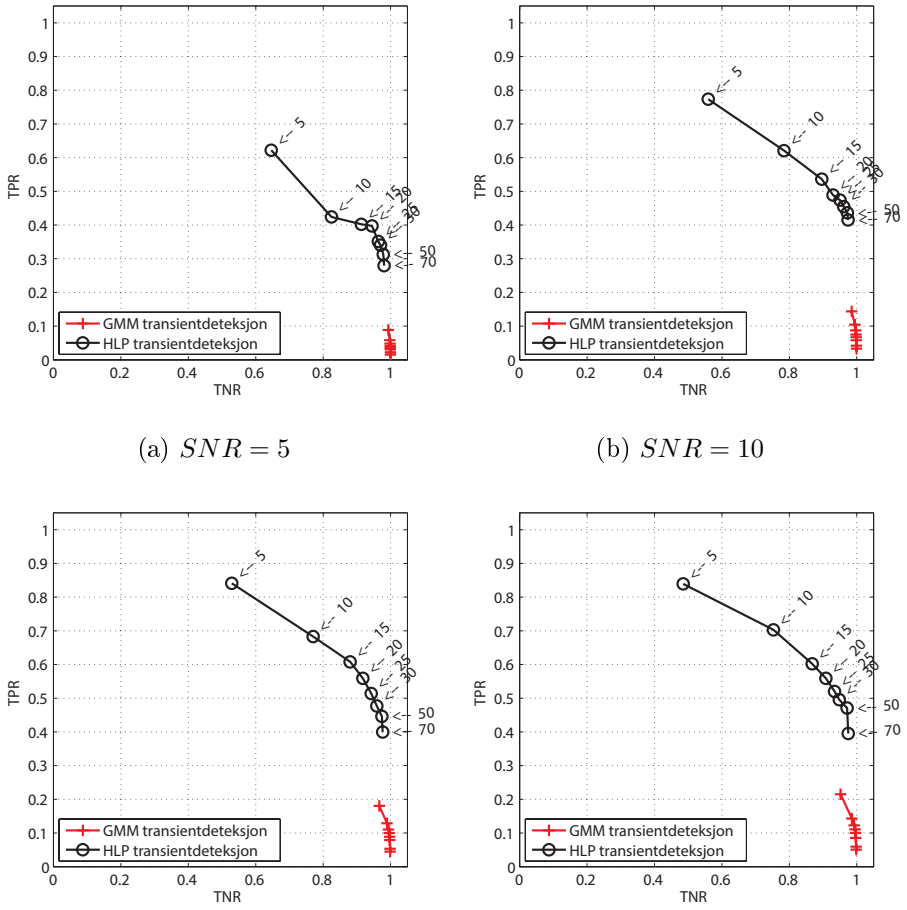
Figur 4.7: Operasjonskarakteristikk ved (a) likevektede og (b) talevektede delbåndsavgjørelser.

4.3 Transientdeteksjon

For transientdetektorene er det også undersøkt virkningen av forskjellige systemparametre. Det er også her tatt utgangspunkt i det standardoppsettet av systemparametre gitt i tabell 4.1, og enkelte parametre er endret for å demonstrere virkningen disse har på transientdetektorene. Om ikke annet er spesifisert, er det parametrene i tabell 4.1 som er benyttet i forsøkene. Noen av testene er gjort tilsvarende som for taledetektorene, mens andre undersøker parametre unike for transientdeteksjonsalgoritmen, og er dermed kun utført for transientdetektorene.

4.3.1 Signal-støy-forhold

For å undersøke deteksjonsmetodene under forskjellige nivåer av bakgrunnsstøy, er prestasjonene undersøkt ved fire forskjellige SNR-verdier. Disse er som for taledetektorene 5 dB, 10 dB, 20 dB og 30 dB, og resultatene er vist i figur 4.8.



(a) $SNR = 5$ (b) $SNR = 10$

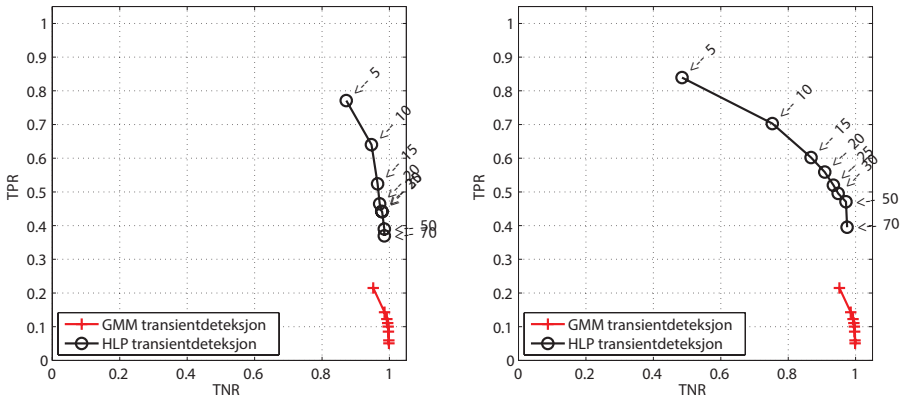
(c) $SNR = 20$ (d) $SNR = 30$

Figur 4.8: Operasjonskarakteristikk for transientdeteksjon ved varierende signal-støy-forhold.

4.3.2 Dobbelt parametersett

I figur 4.9 demonstreres prestasjonsforskjellen i transientdeteksjonen ved å benytte henholdsvis ett og to parametersett i tale- og transientdeteksjons-

algoritmen. Ved bruk av to parametersett, $\lambda_{\text{kont.}}$ og $\lambda_{\text{bet.}}$, vil det være $\lambda_{\text{bet.}}$ som benyttes for transientklassifiseringen.

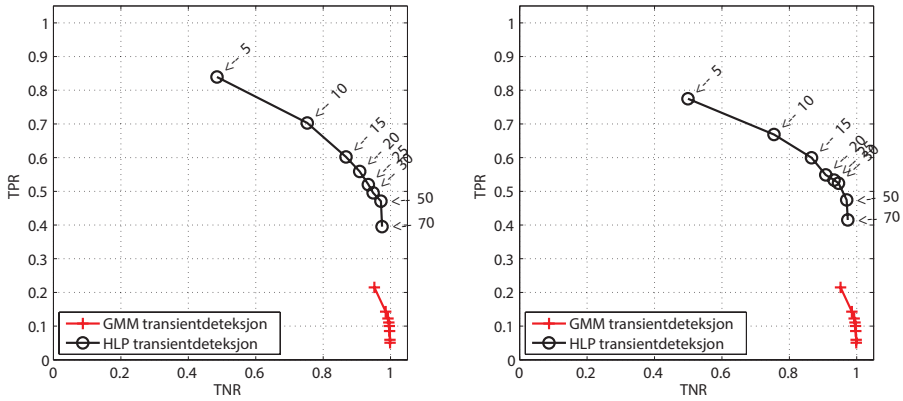


(a) `hlpupd = 0` (b) `hlpupd = 1`

Figur 4.9: Operasjonskarakteristikk for transientdeteksjon med (a) enkelt og (b) dobbelt parametersett.

4.3.3 Modelloppdateringsbetingelser

Det er i figur 4.10 vist innvirkningen det har på operasjonskarakteristikken for transientdeteksjon når taledeteksjonsavgjørelsen benyttes som oppdateringsbetingelse for oppdatering av modellparametrene $\lambda_{\text{bet.}}$. Denne parameteren skal i teorien kun ha innvirkning på transientdeteksjonen, da parameteren kun dikterer om $\lambda_{\text{bet.}}$ skal oppdateres eller ikke. Dermed er forskjellene mellom figurene 4.10a og 4.10b større enn for tilsvarende figurer for taledeteksjon, figurene 4.6a og 4.6b. Plottene er resultatet av å skru av og på `VADDECIDEUPDATE`.



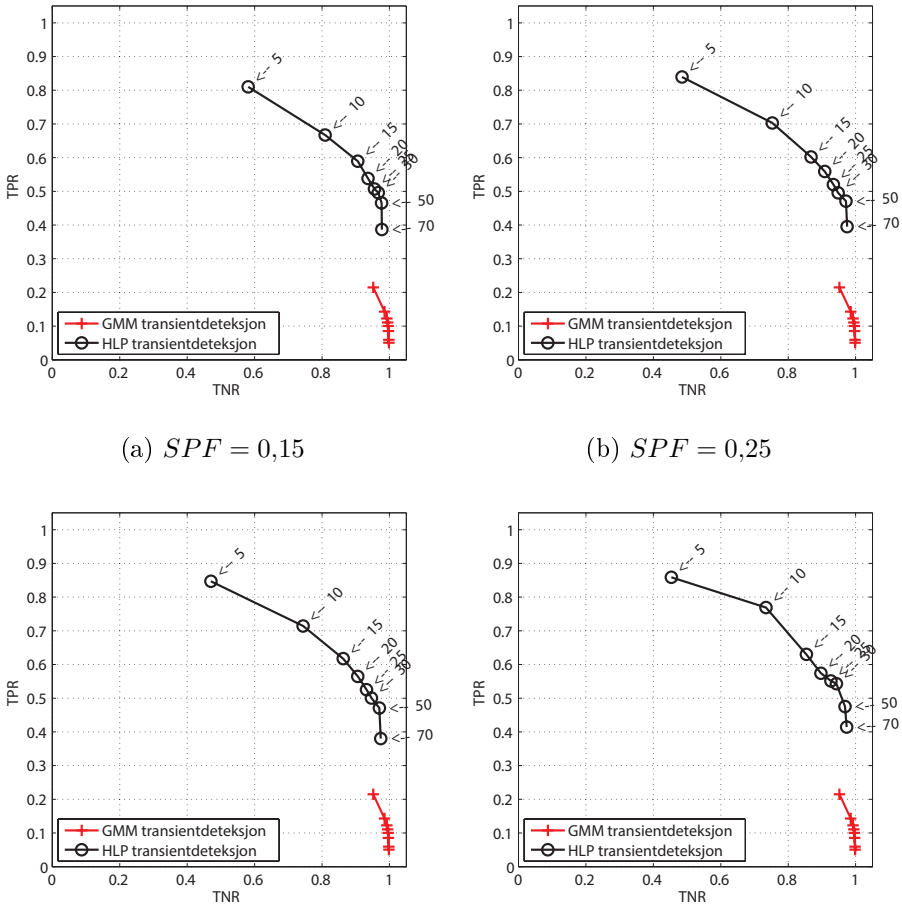
(a) Taleindikator ws_{k+1}

(b) VAD-avgjørelse

Figur 4.10: Operasjonskarakteristikk for transientdeteksjon med forskjellige betingelser for modellfrys.

4.3.4 Oppdateringsterskel SPF

Når oppdateringen av λ_{bet}. bestemmes av ws_{k+1}, er dette med basis i en grenseverdi SPF for taleaktivitetsmålet ws_{k+1}. Det er i figur 4.11 vist hvordan fire forskjellige terskler for taleaktivitetsmålet gir seg utslag i endrede operasjonskarakteristikker for transientdeteksjonen. Sannsynlighetsmålet SPF varieres fra 0,15, via 0,25 og 0,35, opp til 0,5. Verdier over dette gir ikke særlige endringer, ettersom det da stilles for høye krav til talesannsynlighet for modelloppdatering. Nivåer særlig lavere enn 0,15 medfører at oppdaterings-stoppen ikke fungerer særlig godt.



(a) $SPF = 0{,}15$

(b) $SPF = 0{,}25$

(c) $SPF = 0{,}35$

(d) $SPF = 0{,}5$

Figur 4.11: Operasjonskarakteristikk for transientdeteksjon ved forskjellige SPF.

Kapittel 5

Kommentarer og videre arbeid

Resultatene for HLP EVAD som vises i avsnitt 4 er, ved fornuftige systemparametre, jevnt over like gode, eller bedre enn, resultatene for GMM EVAD. Samtidig observeres det fra figur 4.2 at både Sohn VAD og ITU VAD presterer betraktelig dårligere under de gitte testforholdene. Dette danner grunnlag for å hevde at endringene som er foreslått i denne rapporten generelt sett fungerer etter hensikten.

Det er gjennomgående små forskjeller mellom GMM EVAD og HLP EVAD når det kommer til taledeteksjonsprestasjoner. Disse systemene burde, for taledeteksjon, i teorien yte tilnærmet identisk, ettersom de benytter delvis samme fremgangsmåte. I denne rapporten omtales tale- og transientdetektorene ofte på en slik måte at de kan oppfattes som to separate systemer. Dette er imidlertid aldri tilfellet, og nøyaktigheten til den ene detektoren avhenger til en viss grad av den andre. Resultatene i operasjonskarakteristikkplottene for taledeteksjon kan derfor ikke sees på helt isolert fra de tilsvarende resultatene for transientdeteksjon. For GMM EVAD og HLP EVAD antaes det at den lille differansen som er mellom deres resultater for taledeteksjon skyldes GMM EVADs svake resultater for transientdeteksjon og følgene av dette: rammer som egentlig er fasitmerket transient klassifiseres av GMM EVAD som tale, eller at rammer med faktisk taleaktivitet

klassifiseres av HLP EVAD som transient. Begge disse tilfellene er kilder til redusert treffrate, og vil kunne medføre en liten forskjell i ytelesen mellom de to. Ettersom transientandelen er forholdsvis lav, vil ikke dette medføre store utslag i treffratene, men tydeligvis nok til at det vises, eksempelvis i figur 4.2.

Både tale- og transientdeteksjon er gjennomført med testfiler som inneholder transientlyder. Det kan kanskje synes uheldig å vurdere en taleaktivitetsdetektor ved å påtrykke testfiler som inneholder en tredje lydklasse. Dette burde imidlertid ikke sees som urimelig. Dels fordi transientlyder opptrer i de reelle situasjonene testdataene er ment å skulle simulere, og dermed bidrar til å gjøre testene mer virkelighetstro. Ettersom transientlydene stort sett (i det minste initielt) er energirike lyder, er det stor sannsynlighet for at taledetektorene klassifiserer disse som tale. Når da fasitmerkingen for transienter behandles som om de er talemerker ved beregning av resultater for ren taledeteksjon (se avsnitt 3.2.4), burde ikke det faktum at testdataene inneholder transientlyder påvirke resultatene for taledeteksjon i særlig grad.

5.1 Taledeteksjon

Forbedringer av taledeteksjonsresultatene i seg selv har ikke vært spesielt i fokus i dette arbeidet, da taledeteksjon allerede fungerer rimelig tilfredsstillende under forventede bruksforhold, jf. figur 4.1. Annet enn forsøket som er gjennomført med talevektet rammeavgjørelse, er ingen av de endringene eller parametervariasjonene som er gjort i dette arbeidet gjort med den hensikt å forbedre taledeteksjonen spesielt. Det har imidlertid vært en målsetting å forbedre resultatene for transientdeteksjon uten at dette går på bekostning av taledetektorens nøyaktighet.

5.1.1 Signal-støy-forhold

For de forskjellige SNR-verdiene som er forsøkt, vist i figur 4.1, observeres det at utviklingen i klassifiseringsnøyaktighet er som forventet: prestasjonene synker med signal-støy-forholdet. Det oppnås relativt gode treffrater

selv ved 5 dB (figur 4.1a). Dette er ikke veldig overaskende, da støytypen er forholdsvis «enkel», altså at den er jevn og veldefinert, samtidig som 5 dB innebærer at talesignalets energi tross alt ligger noe over støyenergien. Det er verdt å nevne at ITU VAD presterer rimelig likt under de forskjellige SNR-verdiene. Dette skyldes antageligvis at metoden er utviklet for bruk i støyende omgivelser, og dermed benytter seg av robuste signalegenskaper som takler lave SNR-verdier godt. ITU VAD skiller seg også ut ved å ha jevnt over høy taletreffrate (TPR) på bekostning av treffraten for ikke-tale (TNR). Dette er et bevisst valg fra utviklernes side, ettersom denne taledetektoren er utviklet for bruk i talekoding. Som nevnt i avsnitt 1, er det i slike situasjoner taletreffrate som har høyest prioritet. Dette går nødvendigvis på bekostning av treffraten for ikke-tale, ettersom det i tvilssituasjoner er taleklassifisering som favoriseres.

Når det gjelder lave signal-støy-forhold, er det verdt å merke seg oppdateringsbegrensningen i (2.8c). I praksis sier denne at modellkomponentene må ha et signal-støy-forhold på minimum δ, ettersom det er *logaritmisk* energi som modelleres. Dermed vil SNR-verdier lavere enn δ antageligvis gi dårlige resultater med både GMM EVAD og HLP EVAD.

5.1.2 Desisjonsgrenseforskyvning γ

Den parameteren som gir størst uttelling på klassifiseringsnøyaktigheten, er desisjonsgrenseforskyvningen γ. Parameterens innvirkning sees tydelig i figur 4.3, og er enkeltparameteren som i hovedsak styrer hvorvidt taleklassen skal favoriseres i deteksjonsavgjørelsen. Som (2.10) impliserer, bekrefter figurene 4.3 og 4.4 at klassifiseringen tenderer mot økt taletreffrate med synkende verdier for γ.

For bruk i lydnivåutjevningssammenheng vil det være, som for talekoding, viktigst å detektere talerammene korrekt, slik at disse håndteres på en måte som resulterer i best mulig brukeropplevelse[1]. Det vil sjelden medføre særlige problemer, eventuelt økt prosessering, om noen segmenter med ikke-tale skulle behandles som tale. Det vil derfor i slike systemer være gunstig å sette γ lavt, selv om resultatene i 4.2.2 viser at det ikke nødvendigvis er

[1] Hvordan disse skal håndteres i AGC-systemet er utenfor dette arbeidet.

mye å hente i økt taletreffrate – om ikke TNR skal nedprioriteres totalt, jf. 4.4a og 4.4c.

Figur 4.4 viser ekstremverdier av γ, og det kommer tydelig frem at parameteren har stor innvirkning på resultatene. For $\gamma = 1$ i figurene 4.4b og 4.4d sees det at treffraten for ikke-tale er tilnærmet én, noe som skyldes at desisjonsgrensen, som gitt av (2.9), er i krysningspunktet mellom de to modellkomponentene. Dette setter høyere krav til at signalenergien må være tilstrekkelig høy jevnt over flere delbånd, noe som reduserer mulighetene for delbåndsklassifiserte talerammer, hvilket igjen reduserer sjansen for at rammen taleklassifiseres. Støy er lettere å detektere med de testdataene som benyttes, da talemerkede rammer kan inneholde mye forskjellig hva energinivå angår, mens flertallet[2] av de rammene som er fasitmerket som ikke-tale inneholder additiv, hvit, gaussisk støy, med ganske klart definert energi-innhold. Støyrammene vil i så måte være meget godt tilpasset den gaussiske modellen som brukes i HLP EVAD og GMM EVAD, ettersom støyen er generert fra en tilsvarende modell. Talerammene er naturlig forekommende, og signalmodellen vil i utgangspunktet ikke passe like godt til tale-energien som støy-energien gjør. Dette kan være årsak til at ikke-taletreffraten i de fleste tilfeller holder seg på jevnt over på et høyere nivå enn taletreffraten.

Forskjellen mellom høy og lav SNR-verdi resulterer, som forventet, i hovedsak i generell nedgang i deteksjonsnøyaktighet. Sammenligningen mellom høy/lav SNR og høy/lav γ er inkludert for å vise hvordan γ på mange måter betyr mer for deteksjonsprestasjonene enn SNR.

5.1.3 Dobbelt parametersett

Ved å benytte to modellparametersett ved frysing av modellparametre, $\lambda_{\text{kont.}}$ og $\lambda_{\text{bet.}}$, oppnås det stor positiv effekt på taledeteksjonsresultatene, som figur 4.5 viser. Ved `hlpupd = 0` (figur 4.5a) benyttes det kun ett parametersett, λ, som fryses ved fravær av taleaktivitet i inngangssignalet, diktert av (2.14) og (2.15) (se avsnitt 2.3.2 for utdypende beskrivelse).

[2] De kunstig innsatte pausene er av forholdsvis mye lenger varighet enn de stillhetsperiodene som måtte eksistere i talefilene fra TIMIT.

Det å fryse modellparametrene er ikke ønskelig i seg selv, men det er tenkt at dette skal forhindre at drivingsproblematikken volder for mye skade på modellparametrene. Driving medfører nemlig ikke særlige problemer for binær klassifisering, som ren taledeteksjon er. Det er først når detektoren skal kunne skille mellom ikke-tale, tale og transientlyder drivingsproblematikken gjør seg gjeldende. For to-delt klassifisering stoppes drivingen ved oppdateringsbegrensning (2.8c), og taledeteksjon fungerer tilfredsstillende. For tre-delt klassifisering derimot, vil transientsannsynlighetsmålet i (2.11), etter noe tids fravær av taleaktivitet, gi store verdier ved gjenopptatt taleaktivitet. Dette skjer fordi sannsynligheten for klassetilhørighet for både ikke-tale og tale vil bli forholdsvis lav etter at talekomponenten av modellen har blitt oppdatert med ikke-tale-data. Dermed vil de fleste innkommende talerammene etter en stillhetsperiode klassifiseres som transientlyder, inntil talemodellen er tilstrekkelig gjenopprettet.

Dermed er det å få til både tale- og transientdeteksjon samtidig, med tilfredsstillende nøyaktighet, vanskelig med ett parametersett. Ettersom oppdateringsbetingelsene (2.14) og (2.15) er gitt av λ_k, vil det medføre en noe uheldig tilbakekobling, da modelloppdateringsbetingelsen avhenger av modellparametrene. Begge disse problemene, både drivings- og tilbakekoblingsproblematikken, løses ved å benytte to parametersett, noe som vises av figur 4.5b. Med denne løsningen kjøres taledeteksjonen, med modellparametersettet $\lambda_{\text{kont.}}$, som normalt for HLP EVAD (tilsvarende som for GMM EVAD), mens et kopisett, $\lambda_{bet.}$, med modellparametre for bruk i transientdeteksjonen kun oppdateres når (2.14) indikerer taleaktivitet.

5.1.4 Modelloppdateringsbetingelser

Da dette arbeidet ble iverksatt, ble det raskt klart at for å løse drivingsproblematikken måtte modelloppdateringen begrenses på en eller annen måte. Som beskrevet i avsnitt 2.3.2, er det undersøkt flere mulige løsninger for å bestemme når modellen burde oppdateres fullstendig og når kun støy-komponenten burde oppdateres. Som utgangspunkt for videre arbeid, ble taleaktivitetsindikatoren i (2.14) valgt, og dette er den løsningen som er benyttet i resten av arbeidet.

Ser en på kravene som stilles til den signalegenskapen som skal avgjøre

hvorvidt $\lambda_{\text{bet.}}$ skal oppdateres eller ikke, er det åpenbart at det som er ønskelig, er en taleaktivitetsdetektor. Deteksjonsavgjørelsen for hver ramme er tilgjengelig uten ekstra kostnad, og burde derfor være det foretrukne valget for oppdateringsbetingelsen, gitt at en har tro på at taledeteksjonsalgoritmen som benyttes er den mest effektive tilgjengelig.

Ideen om å benytte selve taledeteksjonsavgjørelsen som oppdateringsbetingelse er, utrolig nok, en idé som først har oppstått sent i arbeidet. Av forskjellene mellom figurene 4.6a og 4.6b, sees det imidlertid at betydningen av å benytte den beste mulige taledetektoren i modelloppdateringsavgjørelsen ikke nødvendigvis er så stor. Resultatene er nærmest identiske hverandre, men kurven for HLP EVAD ved bruk av ws_{k+1} (figur 4.6a) er smått forskjøvet oppover og mot venstre, altså noe høyere taletreffrate, men litt lavere treffrate for ikke-tale.

5.1.5 Delbåndsvekting

Ved bruk av mel-frekvens-delbånd vil delbåndenes frekvens-båndbredde øke med delbåndsnummeret, slik at de høyere delbåndene vil inneholde et større frekvensområde enn de lave delbåndene. Ettersom de lavfrekvente delene av signalet inneholder mer tale-energi, vil mel-delbånd til en viss grad sørge for at delbåndene er oppdelt slik at tale-energien fordeles jevnt over alle delbåndene. Med en antagelse om at en stor del av tale-energien befinner seg i frekvensområdet 200 Hz til 1500 Hz, er det også forsøkt å legge mer vekt på de delbåndene som korresponderer til dette frekvensområdet i taleaktivitetsavgjørelsen. Resultatene av vektet og uvektet deteksjonsavgjørelse sees i figur 4.7. Resultatene i avsnitt 4.2.5 viser at talevektet deteksjonsavgjørelse presterer litt dårligere enn likevektet. Det er imidlertid kun forsøkt ett sett med vektingskoeffisienter, og disse er satt etter intuisjon, noe som kombinert ikke gir et spesielt godt grunnlag for å si noe generelt om hvorvidt vekting har noe for seg eller ikke.

Vekting av delbåndsavgjørelsene er muligens ikke en ideell løsning for å øke deteksjonspresisjonen, og det foreslåes heller å undersøke mulighetene for å benytte delbånds-sannsynligheten for klassetilhørighet direkte i rammeavgjørelsen. Slik beholdes mest mulig nøyaktighet i desisjonsavgjørelsen frem til rammeavgjørelsen taes. Dette kan eventuelt kombineres med metoder

for å utnytte sammenhengen mellom rammene, for eksempel ved bruk av overgangssannsynlighetsmodeller. Det antaes å ligge et uutnyttet potensial i inter-ramme-korrelasjonen, og at denne kan utnyttes bedre enn hva som er tilfellet med glattingsmetoden fra [14]. Dette viser arbeidene i [8] med tydelighet, der det benyttes en mer avansert metode for inter-rammeglatting av deteksjonsavgjørelsen (dette er konseptuelt beskrevet i avsnitt 2.1.2).

5.2 Transientdeteksjon

Transientdeteksjon oppnår generelt bedre resultater med endringene som er gjort i HLP EVAD sammenlignet med GMM EVAD. I hovedsak antaes dette å skyldes den endrede modelloppdateringsmetoden med dobbelt parametersett.

Det at treffraten for ikke-transient, som logisk sett burde være relativt enkelt å detektere, ikke er høyere enn det den gjennomgående er, mistenkes det at skyldes det at transientlydlengden er fastkodet i deteksjonsalgoritmen. Dette medfører at om én ramme detekteres som transient, vil et antall (47 i testene) etterfølgende rammer også markeres som transient. Så om den initielt klassifiserte rammen, og eventuelt de 47 etterfølgende, egentlig ikke inneholder transient, vil de allikevel feilmerkes, uavhengig av om algoritmen klassifiserer de etterfølgende rammene som transientlyd eller ikke. Om feilklassifisering av transientlyder skjer gjentatte ganger, vil det kunne ha til dels stor påvirkning på treffraten for ikke-transient-klassen (TNR).

En mulig løsning på dette problemet vil være å heller benytte seg av en overgangssannsynlighetsmodell à la den som benyttes i [8]. Enda bedre hadde det nok vært om det utvikles en metode for å avgjøre transientlydenes varighet. Det er nærliggende å anta at transientlydene har et bredspektret frekvensinnhold, ettersom brå steg i tid gir bredt spekter. Dermed kan muligens nullkrysningsraten eller lignende benyttes for å avgjøre når transientlyden er over, jf. [4].

Et problem med metoden som benyttes for deteksjon av transientlyder, er det at signalenergien glattes over flere rammer (se avsnitt 2.3.3). Dette

ødelegger noe av karakteristikken ved transientlydene, og bidrar nok en del til å redusere treffratene for transientdetektorene. Glattingen er imidlertid lik for begge transientdetektorene, så reduksjonen vil kun være absolutt, og ikke relativ mellom de to. Om glattingsmetoden skulle vært fjernet ville taledeteksjonen sannsynligvis blitt mer ustabil og resultatene antageligvis dårligere. Glattingen har som hensikt å redusere store, brå variasjoner i energinivå mellom rammene, og det er på mange måter akkurat det en transientdetektor ønsker å detektere. I så måte har tale- og transientdeteksjon motstridende interesser, og det burde kanskje undersøkes mulighetene for andre metoder for transientdeteksjon separat fra taledeteksjon.

5.2.1 Signal-støy-forhold

Som for taledeteksjon, vil reduksjon i signal-støy-forhold som regel medføre reduserte treffrater for transientdetektoren. Figur 4.8 bekrefter at dette er tilfellet. Figuren viser også at det ikke er særlig forskjell i ytelse for de to høyeste SNR-verdiene. Dette skyldes nok at SNR-nivået er tilstrekkelig høyt for at transientdeteksjonen skal fungere rimelig greit. Litt rart, imidlertid, er det at det for transientmål-terskel $t_{tr} = 0{,}15$ oppnås så vidt bedre resultater for $SNR = 20$ enn $SNR = 30$, både for TNR og TPR. Dette er sannsynligvis et tilfeldig utslag, og bør nok ikke tillegges for mye vekt. Ellers observeres det at resultatene for HLP EVAD er jevnt over mye bedre enn for GMM EVAD, som stort sett feiler i å detektere transientlydene (lav TPR). Dette viser klart de negative effektene driving har på transientdeteksjon.

5.2.2 Dobbelt parametersett

Forskjellen i treffrate mellom å stoppe oppdateringen av talekomponenten ved fravær av taleaktivitet og å oppdatere modellen uavhengig av inngangssignal vises tydelig i figur 4.9a, der GMM EVAD oppdateres kontinuerlig, mens HLP EVAD stopper oppdateringen ved indikasjon på fravær av tale. Sammenlignes figurene 4.9a og 4.9b, observeres det at treffratekurven strekker seg mer ut i bredden for figur 4.9b enn for 4.9a, samtidig som 4.9b ikke har nevneverdig bedre transient-treffrate. En kan derfor ar-

gumentere for at best resultater for transientdeteksjon oppnås med enkelt parametersett med oppdateringsstopp. Med ett enkelt parametersett lider imidlertid taledeteksjonen (se figur 4.5a), mens økningen i presisjon for transientdeteksjon ikke går ut over taledetektorens nøyaktighet om det benyttes dobbelt parametersett. Utfordringene som er nevnt innledningsvis i avsnitt 5.2 bidrar imidlertid til at transientdeteksjonsnøyaktigheten ikke er helt på topp.

5.2.3 Modelloppdateringsbetingelse

Deteksjonsresultatene for transientdeteksjon påvirkes ikke spesielt mye om det er deteksjonsavgjørelsen eller taleindikatoren ws_{k+1} som benyttes i modelloppdateringsbetingelsen. Deteksjonsresultatene når VAD-avgjørelsen benyttes (figur 4.10b) er litt mer konsistente for alle de forskjellige transientmåltersklene enn om taleindikatoren ws_{k+1} benyttes (figur 4.10a). Resultatene ligger imidlertid litt høyere i transient-treffrate for ws_{k+1}, uten at treffraten for ikke-transient er synlig redusert. Den økte evnen til å klassifisere transientlyder korrekt kan skyldes at taleindikatoren ws_{k+1} har innebygd en viss treghet i indikasjonsevnen for taleaktivitet, ettersom den avhenger av $w_{k,l}$ som oppdateres gradvis. Dermed vil ikke korte transientlyder, som av taledeteksjonsalgoritmen muligens vil klassifiseres som tale, og dermed oppfylle oppdateringsbetingelsen, bidra til modelloppdateringen. Dermed vil ikke talekomponenten oppdateres med transientlyd-data, og algoritmen vil derfor lettere klassifisere transientlyder korrekt.

For de høyeste terskelverdiene t_{tr} har oppdateringsbetingelsen mindre å si. En mulig forklaring på dette er at de transientlydene som detekteres korrekt ved såpass høye terskelverdier (nærmest) uansett ligger såpass mye høyere enn talesignalet i energi, slik at de relativt lett klassifiseres korrekt, mer eller mindre uavhengig av hvor «transientpåvirket» modellparametrene måtte være.

5.2.4 Oppdateringsterskel SPF

Med økt terskelverdi for oppdateringsbetingelsen ved bruk av ws_{k+1} som taleindikator, flyttes operasjonskarakteristikkurven for HLP EVAD seg

oppover og mot venstre, jf. figur 4.11. Argumentene fra forrige avsnitt gjør seg også gjeldende her, da terskelen styrer hvor strengt modellparametrene skal oppdateres. Det observeres også her at terskelverdien har mindre innvirkning dess høyere transientmålterskelen t_{tr} er satt, med samme begrunnelse som gitt i 5.2.3.

5.3 Forbedringer

Selv om noen av problemene ved GMM EVAD er forsøkt løst i HLP EVAD, er det fremdeles flere utfordringer som gjenstår. Mange av disse er fremsatt løpende igjennom rapporten, men det suppleres med noen flere her.

Fastsatte terskler, både for taleindikatoren ws_{k+1} og for transientmålet f_k, er lite fleksibelt, og det hadde antagelig vært gunstig om disse var av en mer dynamisk og signalbestemt karakter. Videre er det flere utfordringer knyttet til transientlyder og deteksjon av disse. På et høyere abstraksjonsnivå enn selve transientdeteksjonsalgoritmene kan det nevnes utfordringen med i det hele tatt å entydig definere hva en transientlyd er, samt i hvor lang tid denne er et problem for taletydelighet i lydsignaler. Dette er avgjørende for å kunne utvikle gode metoder for å detektere både ansats og ende på slike transientlyder, slik at følgene av problemene de medfører kan reduseres.

Når det gjelder taledeteksjon, har HLP EVAD klart å beholde de gode egenskapene ved GMM EVAD, og oppnår tilsvarende eller litt bedre resultater enn GMM EVAD. Det er nok mulig å få algoritmen til å prestere enda litt bedre, men dette burde nok gjøres med en konkret målsetting, ettersom det er mange parametre som kan justere resultatet i ønsket retning.

Kapittel 6

Konklusjon

Både tale- og transientdeteksjonsnøyaktigheten har jevnt over blitt bedre med de foreslåtte endringene i HLP EVAD. Det er i hovedsak innføringen av oppdateringsstopp for modellparametrene med dobbelt parametersett som har bidratt til å løse drivingsproblematikken som er tilfellet for GMM EVAD. Det har vært arbeidet en del med løsninger for å realisere en slik oppdateringsstopp – i hovedsak med hva som skal betinge oppdateringsstopp.

Ettersom MATLAB-implementasjonen som fantes for GMM EVAD [1, 2] var lite sanntidsvennlig, samt hadde en del elementer der mulighetene for feil i implementasjon var tilstede, falt valget ned på å gjøre implementasjonen i stor grad på nytt. I den sammenheng ble det også implementert noen nye løsninger for testoppsettet. Eksempelvis ble det lagt en del arbeid ned i å lage en rutine som kan generere testdata fleksibelt, med gode muligheter for fleksibilitet i testdataenes sammensetning. Implementasjonen av deteksjonsalgoritmen er også nå tilpasset sanntidskjøring, da det er tilstrebet å finne løsninger som kan konverteres til sanntidskjøring med minimale inngrep.

Som utbrodert i avsnitt 5.2, vil glattingen som utføres i GMM EVAD og HLP EVAD føre til at deler av det høye energinivået som kjennetegner transientlyder går tapt. Dette vanskeliggjør transientdeteksjonen, men om slik glatting skal benyttes blir en avveining mellom god taledeteksjon eller

god transientdeteksjon. Denne avveiningen er til en viss grad gjennomgående for alle endringer som skal gjøres. Endringer som skal bidra til økt nøyaktighet eller løse et problem for én av deteksjonstypene vil ofte medføre dårligere nøyaktighet eller innføre et nytt problem for den andre deteksjonstypen.

Derfor er det ekstra interessant å se at de løsningene som er presentert og testet i denne rapporten i all hovedsak bidrar til forbedrede resultater for både tale- og transientdeteksjon.

Bibliografi

[1] D. Ying, Y. Yan, D. Jianwu og F.K. Soong. Voice Activity Detection Based on an Unsupervised Learning Framework. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8), November 2011.

[2] B. Paulsrud. Voice Activity Detection with Focus on Low SNR and Transient Noise. Hovedfagsoppgave, Norwgian University of Science and Technology, 2013.

[3] A. Benyassine, E. Shlomot, H.Y. Su, D. Massaloux, C. Lamblin og J.P. Petit. ITU-T Recommendation G. 729 Annex B: a silence compression scheme for use with G. 729 optimized for V. 70 digital simultaneous voice and data applications. *Communications Magazine, IEEE*, 35(9):64–73, 1997.

[4] Lawrence R Rabiner og Marvin R Sambur. An algorithm for determining the endpoints of isolated utterances. *Bell System Technical Journal*, 54(2):297–315, 1975.

[5] Frank K Soong og Biing-Hwang Juang. Line Spectrum Pair (LSP) and speech data compression. I *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, bind 9, side 37–40. IEEE, 1984.

[6] Ian V. McLoughlin. Line Spectral Pairs. *Signal processing*, 88(3):448–467, 2008.

[7] JA Haigh og JS Mason. Robust Voice Activity Detection using Cepstral Features. I *TENCON'93. Proceedings. Computer, Communication,*

Control and Power Engineering. 1993 IEEE Region 10 Conference on, side 321–324. IEEE, 1993.

[8] J. Sohn, N.S. Kim og W. Sung. A Statistical Model-Based Voice Activity Detection. *Signal Processing Letters, IEEE*, 6(1):1–3, 1999.

[9] Rick Jeffs, Scott Holden og Dennis Bohn. Dynamics Processors – Technology & Application Tips. Rapport 155, Rane Corporation, 2005.

[10] Norsk Standard. NS 8175:2012: Lydforhold i bygninger – Lydklasser for ulike bygningstyper, 2012.

[11] ITU-T. ITU-T Recommendation G. 729 Annex B enhancements in voice-over-IP applications - Option 1. 2005.

[12] P. Kabal. ITU VAD Implementation, 2008.

[13] M. Brookes et al. VOICEBOX: Speech Processing Toolbox for MATLAB. *Software, available [Mar. 2011] from www. ee. ic. ac. uk/hp/staff/dmb/voicebox/voicebox. html*, 1997.

[14] Special Mobile Group (SMG) Technical Committee of the European Telecommunications Standards Institute (ETSI). Digital cellular telecommunications system (Phase 2+); Voice Activity Detector (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels; General description (GSM 06.94 version 7.1.0 Release 1998). side 13–15, 1999.

[15] J.R. Hershey og P.A. Olsen. Approximating the Kullback-Leibler divergence between Gaussian mixture models. I *IEEE International Conference on Acoustics, Speech and Signal Processing*, bind 4, side 317–320, 2007.

[16] W.M. Fisher, G.R. Doddington og K.M. Goudie-Marshall. The DARPA Speech Recognition Research Database: Specifications and Status. I *Proceedings of DARPA Workshop on Speech Recognition*, side 93–99, 1986.

[17] K.-F. Lee og H.-W. Hon. Speaker-independent phone recognition using hidden Markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(11):1641–1648, 1989. ISSN 0096-3518.

[18] S.i Nakamura, K. Hiyane, F. Asano, T. Nishiura og T. Yamada. Acoustical Sound Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition. I *LREC*, 2000.

[19] J. Fiscus og C. Wierzynski. NIST STNR Documentation, 1992.

[20] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition. ISSN 0167-8655.

[21] Wayne O. Olsen. Average Speech Levels and Spectra in Various Speaking/Listening Conditions: A Summary of the Pearson, Bennett, & Fidell (1977) Report. *American Journal of Audiology*, 7(2):21–25, 12 1998. ISBN 10590889.

Vedlegg A

`matlab-filer.zip`

MATLAB-kode for implementasjon av de forskjellige aktivitetsdetektorene, samt testrutiner for disse. Testdataene er også inkludert i denne filen. Oversikt over innholdet i filen finnes i filen `%%%DESCRIPTION.txt`.

Vedlegg B

Separate treffrateplott

B.1 Taledeteksjon

Alle plottene i dette kapittelet svarer til korresponderende operasjonskarakteristikkplott i kapittel 4. Plottene benevnes *treffrateplott*, og viser treffratene som funksjon av terskelverdier (delbåndsterskler for taledeteksjon og transientmål-terskler for transientdeteksjon).

B.1.1 Signal-støy-forhold

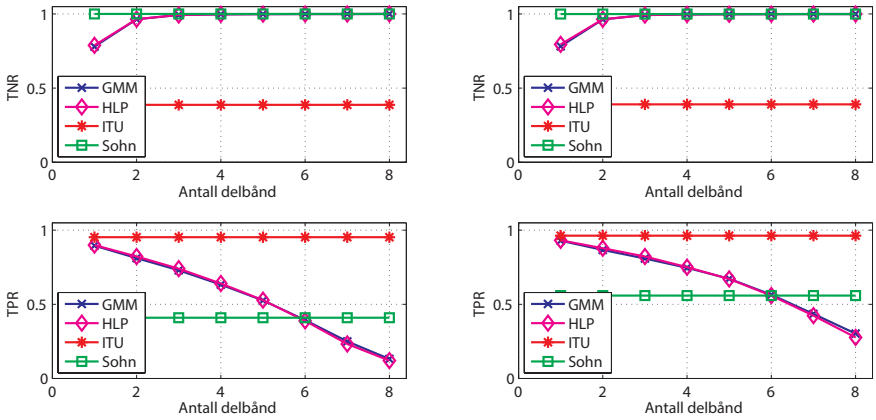
Figur B.1 svarer til figur 4.1.

B.1.2 Desisjonsgrenseforskyvning γ

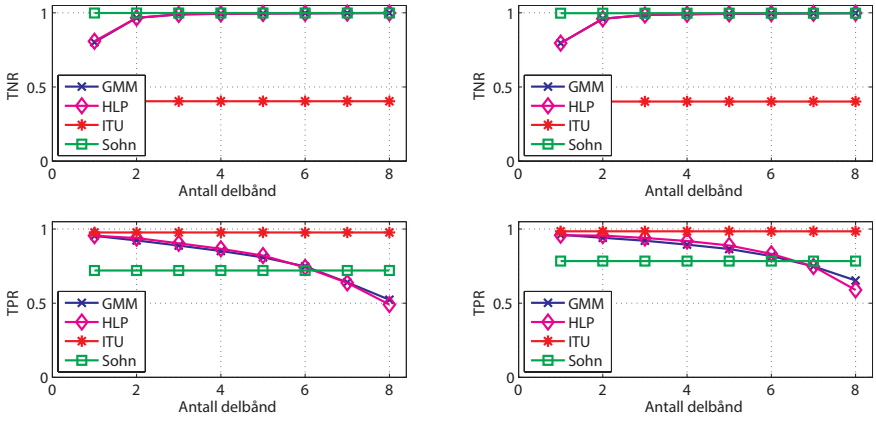
Figur B.2 svarer til figur 4.2.

Figur B.3 svarer til figur 4.3.

Figur B.4 svarer til figur 4.4.

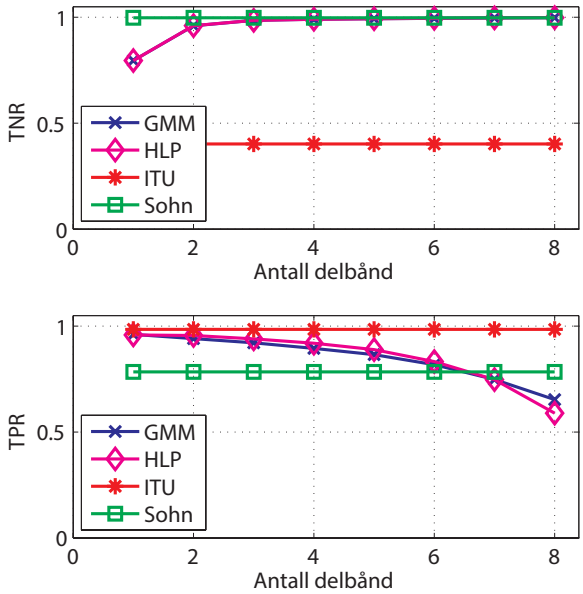


(a) $SNR = 5$ (b) $SNR = 10$

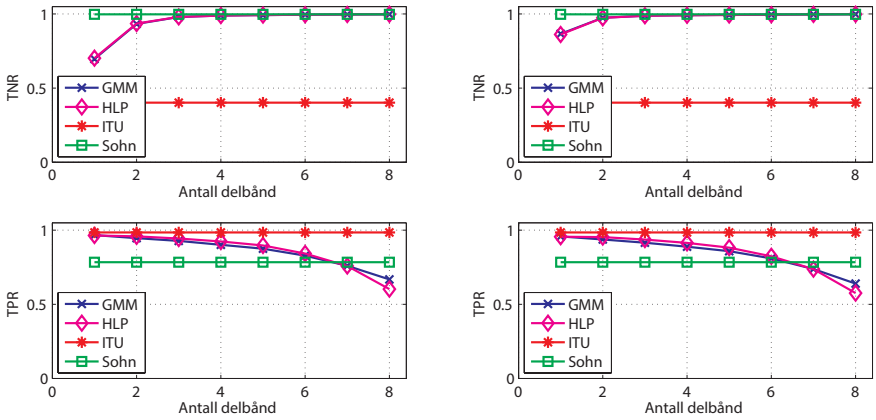


(c) $SNR = 20$ (d) $SNR = 30$

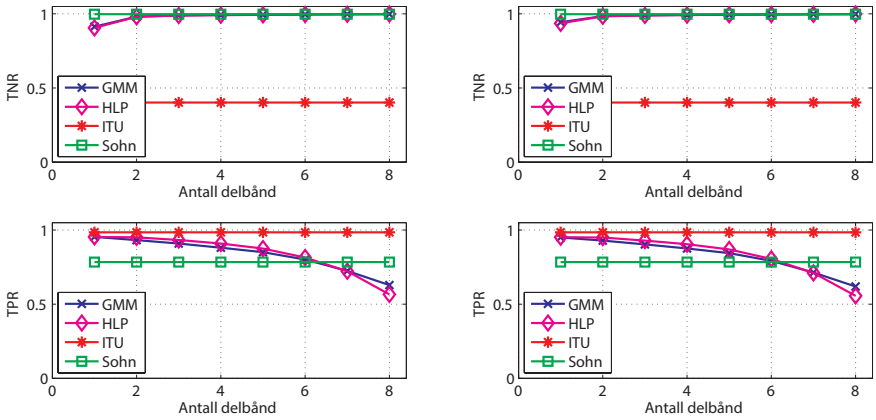
Figur B.1: Treffrate for taledeteksjon ved forskjellig signal-støy-forhold.



Figur B.2: Operasjonskarakteristikk med standardparametre, der γ er satt til 0,45 ved $SNR = 30$.

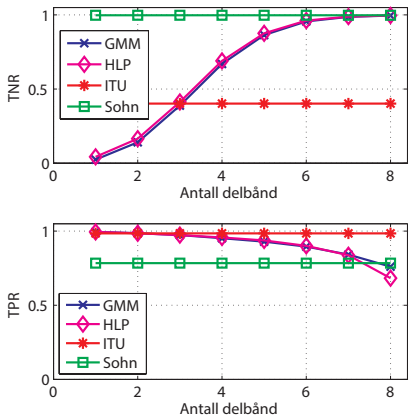


(a) $\gamma = 0{,}4$ (b) $\gamma = 0{,}5$

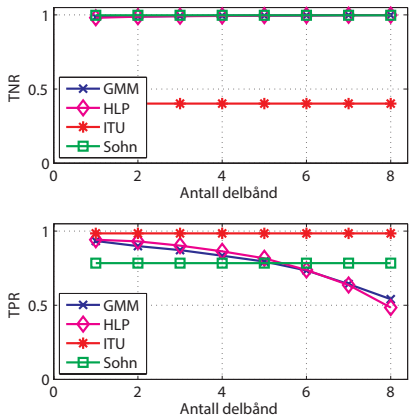


(c) $\gamma = 0{,}55$ (d) $\gamma = 0{,}6$

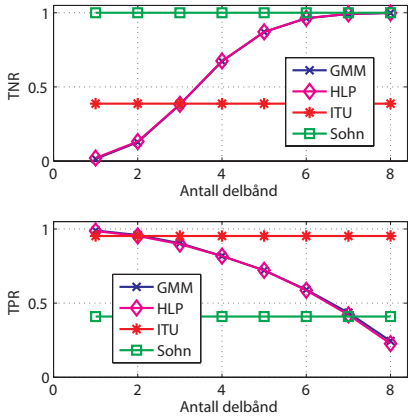
Figur B.3: Alternative γ-verdier ved $SNR = 30$.



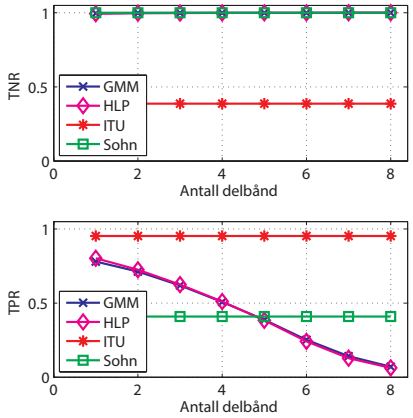
(a) $\gamma = 0{,}1$ og $SNR = 30$



(b) $\gamma = 1$ og $SNR = 30$



(c) $\gamma = 0{,}1$ og $SNR = 5$

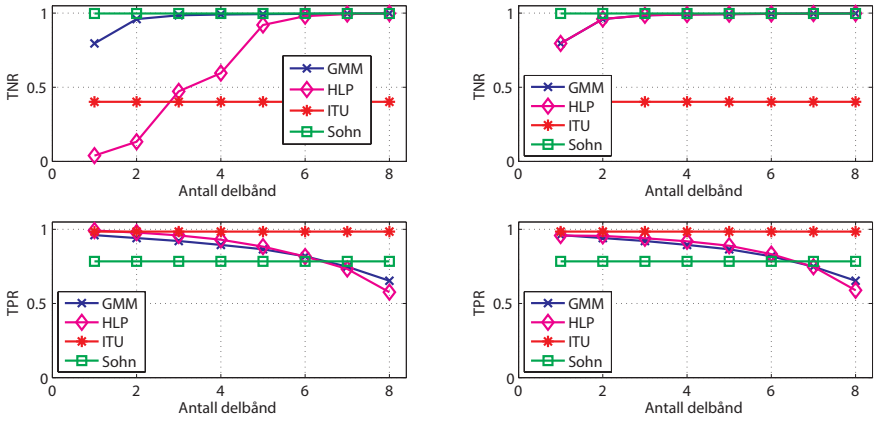


(d) $\gamma = 1$ og $SNR = 5$

Figur B.4: Treffrate for taledeteksjon ved høy/lav γ og høyt/lavt signal-støy-forhold.

B.1.3 Dobbelt parametersett

Figur B.5 svarer til figur 4.5.



(a) `hlpupd = 0` (b) `hlpupd = 1`

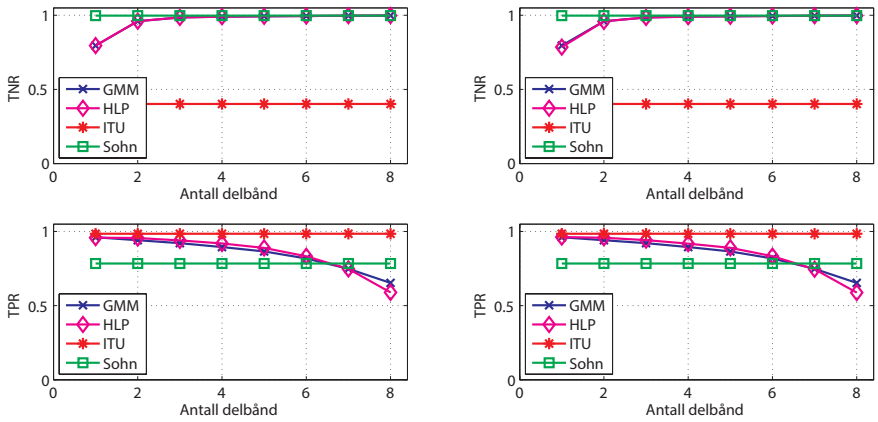
Figur B.5: Treffrate for taletdeteksjon med (a) enkelt og (b) dobbelt parametersett.

B.1.4 Modelloppdateringsbetingelser

Figur B.6 svarer til figur 4.6.

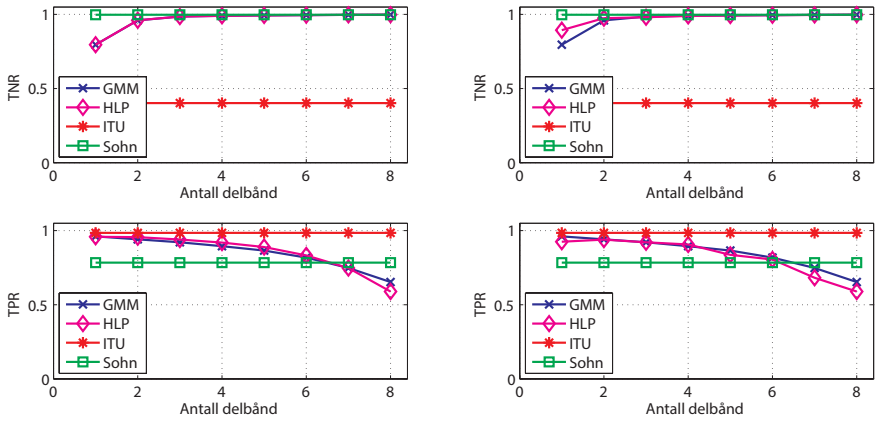
B.1.5 Delbåndsvekting

Figur B.7 svarer til figur 4.7.



(a) Taleindikator ws_{k+1} (b) VAD-avgjørelse

Figur B.6: Treffrate for taledeteksjon med forskjellige betingelser for modellfrys.



(a) Likevektede (b) Talevektede

Figur B.7: Treffrate ved (a) likevektede og (b) talevektede delbåndsavgjørelser.

B.2 Transientdeteksjon

B.2.1 Signal-støy-forhold

Figur B.8 svarer til figur 4.8.

B.2.2 Dobbelt parametersett

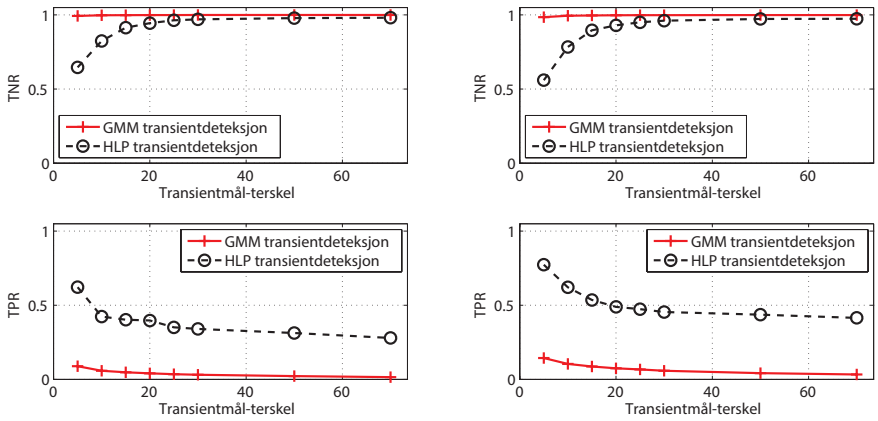
Figur B.9 svarer til figur 4.9.

B.2.3 Modelloppdateringsbetingelser

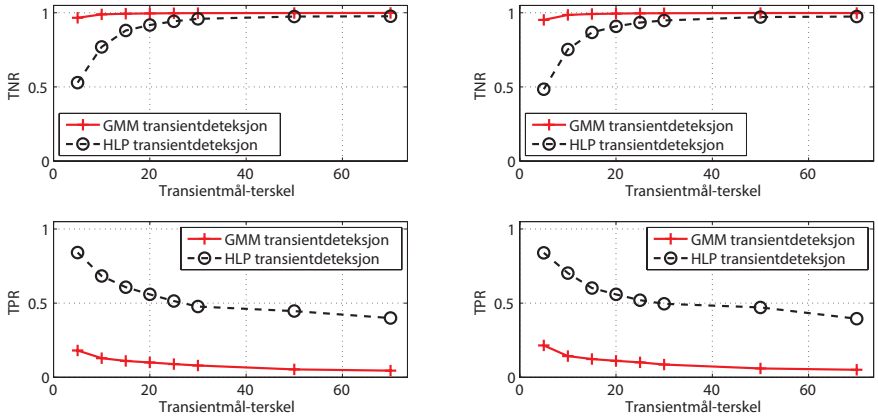
Figur B.10 svarer til figur 4.10.

B.2.4 Oppdateringsterskel SPF

Figur B.11 svarer til figur 4.11.

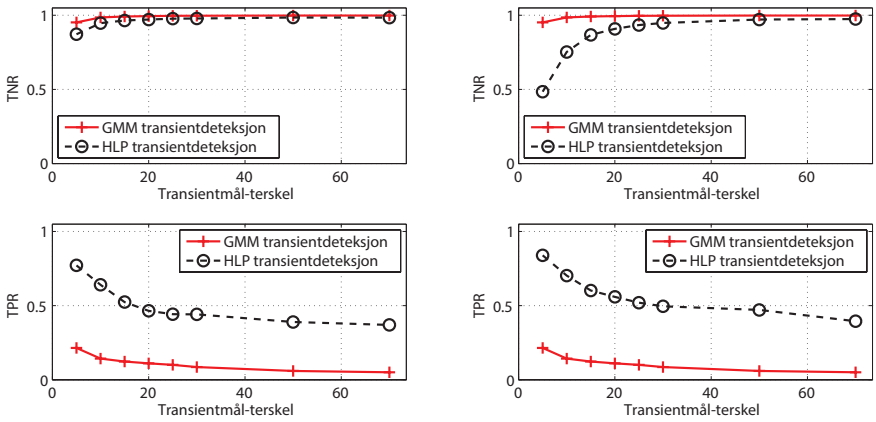


(a) $SNR = 5$ (b) $SNR = 10$



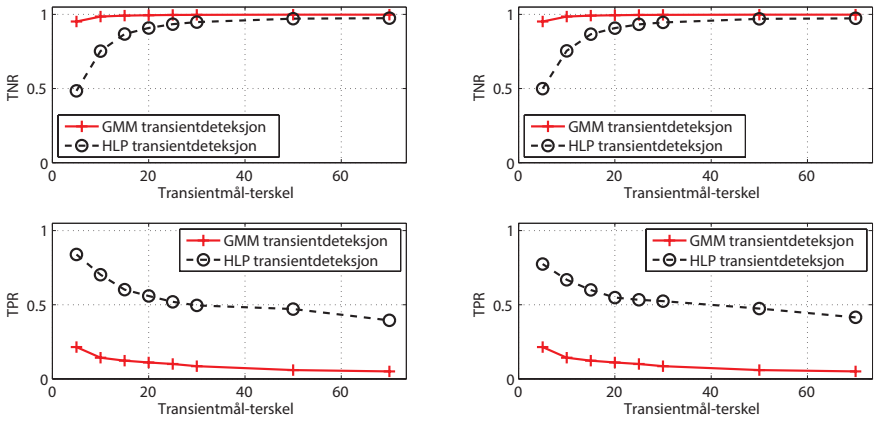
(c) $SNR = 20$ (d) $SNR = 30$

Figur B.8: Treffrate for transientdeteksjon ved forskjellig signal-støy-forhold.



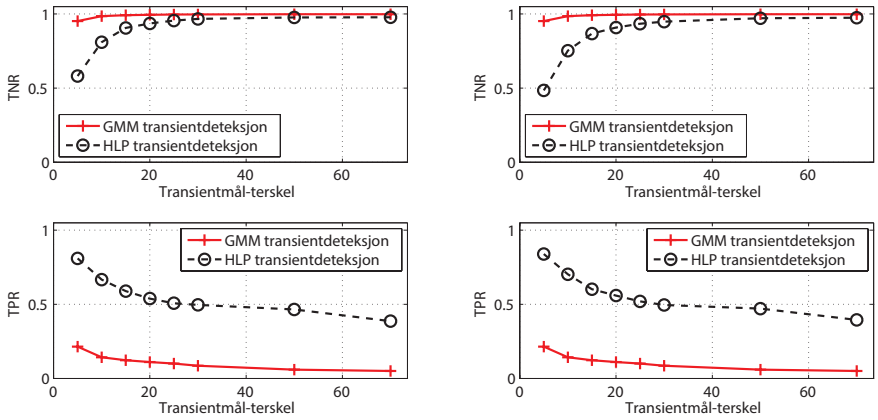
(a) `hlpupd = 0` (b) `hlpupd = 1`

Figur B.9: Treffrate for transientdeteksjon med (a) enkelt og (b) dobbelt parametersett.

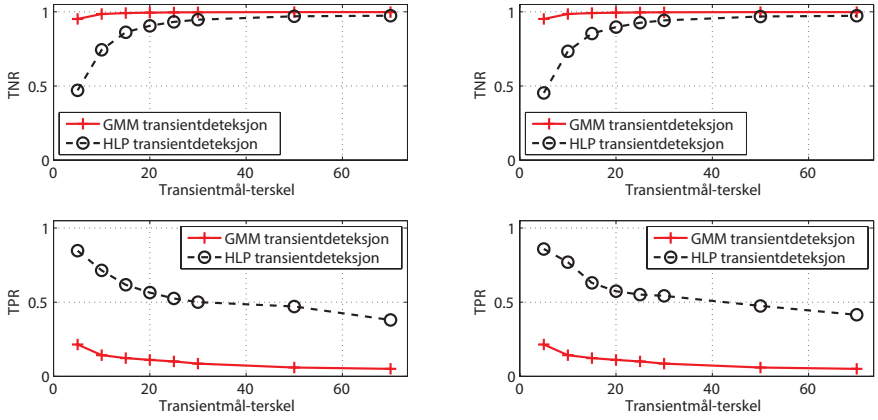


(a) Taleindikator ws_{k+1} (b) VAD-avgjørelse

Figur B.10: Treffrate for transientdeteksjon med forskjellige betingelser for modellfrys.



(a) $SPF = 0{,}15$ (b) $SPF = 0{,}25$



(c) $SPF = 0{,}35$ (d) $SPF = 0{,}5$

Figur B.11: Treffrate for transientdeteksjon ved forskjellige SPF.