



(12) PATENT

(19) NO

(11) 340411

(13) B1

NORGE

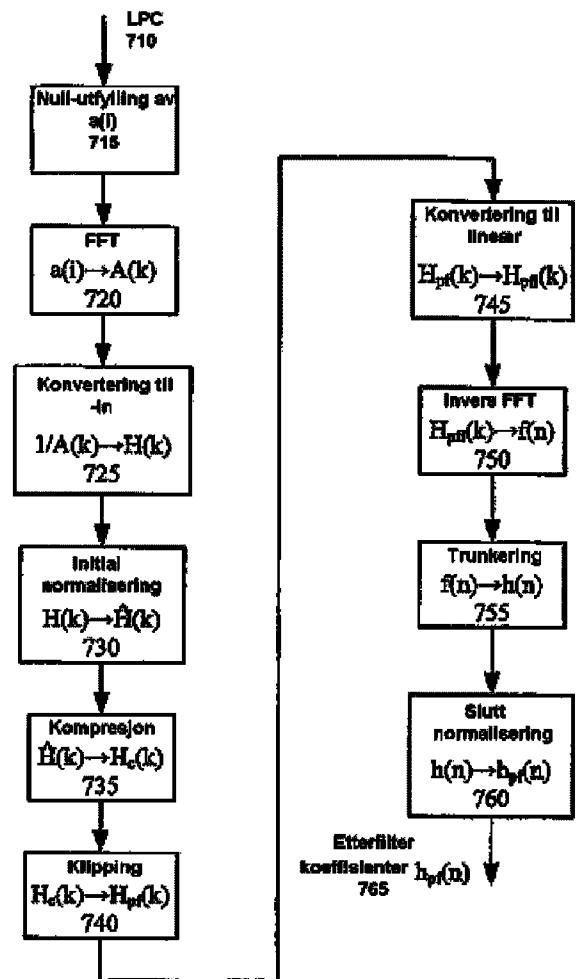
(51) Int Cl.
G10L 19/26 (2013.01)

Patentstyret

(21)	Søknadsnr	20075773	(86)	Int.inng.dag og søknadsnr	2006.04.05 PCT/US2006/12641
(22)	Inng.dag	2007.11.12	(85)	Videreføringssdag	2007.11.12
(24)	Løpedag	2006.04.05	(30)	Prioritet	2005.05.31, US, 11/142,603
(41)	Alm.tilgj	2008.02.28			
(45)	Meddelt	2017.04.18			
(73)	Innehaver	Microsoft Technology Licensing, LLC, One Microsoft Way, US-WA98052 REDMOND, USA			
(72)	Oppfinner	Xiaoqin Sun, c/o Microsoft Corp., One Microsoft Way,, US-WA98052-6399 REDMOND, USA Tian Wang, c/o Microsoft Corp., One Microsoft Way,, US-WA98052-6399 REDMOND, USA Hosam A Khalil, c/o Microsoft Corp., One Microsoft Way,, US-WA98052-6399 REDMOND, USA Kazuhito Koishida, c/o Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA Wei-Ge Chen, c/o Microsoft Corp, One Microsoft Way,, US-WA98052-6399 REDMOND, USA			
(74)	Fullmektig	Bryn Aarflot AS, Postboks 449 Sentrum, 0104 OSLO, Norge			

(54)	Benevnelse	Audiokoding etterfilter
(56)	Anførte publikasjoner	EP 1 308 932
(57)	Sammendrag	

Det beskrives teknikker og verktøy for prosessering av rekonstruerte audiosignaler. For eksempel filtreres et rekonstruert audiosignal i tidsdomenen ved bruk av filterkoeffisienter som beregnes, i det minste delvis, i frekvensdomenen. Som et annet eksempel, produsering av et sett av filterkoeffisienter for filtrering av et rekonstruert audiosignal inkluderer klipping av en eller flere topper av et sett av koeffisientverdier. Som enda et annet eksempel, for en subbåndkodek, i et frekvensområde nær en skjæring mellom to subbånd, forbedres et rekonstruert komposittsignal.



Teknisk område

Beskrevne verktøy og teknikker vedrører audiokodeker og særlig etterprosessering av dekodet tale.

5

Bakgrunn

Med fremkomsten av digitale trådløse telefonnettverk, sanntids overføring av audio over Internett og Internett telefoni har digital prosessering og levering av tale blitt alminnelig. Ingeniører bruker et mangfold av teknikker for å prosessere tale effektivt samtidig som kvaliteten opprettholdes. For å forstå disse teknikker hjelper det å forstå hvordan audioinformasjon representeres og behandles i en datamaskin.

10

I. Representasjon av audioinformasjon i en datamaskin

En datamaskin prosesserer audioinformasjon som en serie av tall som representerer audioen. Et enkelt tall kan representere et audiosampel, hvilket er en amplitudeverdi på et bestemt tidspunkt. Flere faktorer påvirker kvaliteten av audioen, inkludert sampeldybde og samplingsrate.

15

Sampeldybde (eller presisjon) angir det verdiområde av tall som brukes til å representere et sampel. Flere mulige verdier for hvert sampel gir typisk utgang med høyere kvalitet, fordi mer subtile variasjoner i amplitude kan representeres. Et åtte-bit sampel har 256 mulige verdier, mens et seksten-bit sampel har 65.536 mulige verdier.

20

Samplingsraten (vanligvis målt som antallet sampler per sekund) påvirker også kvaliteten. Jo høyere samplingsraten er, jo høyere er kvaliteten, fordi flere lyd-frekvenser kan representeres. Enkelte vanlige samplingsrater er 8.000, 11.025, 22.050, 32.000, 44.100, 48.000 og 96.000 sampler/sekund (Hz). Tabell 1 viser flere formater av audio med forskjellige kvalitetsnivåer, sammen med korresponderende rå-bit ratekostnader.

25

Sampeldybde (bits/sampel)	Samplingsrate (sampler/sekund)	Kanalmodus	Rå-bit rate (bits/sekund)
8	8.000	mono	64.000
8	11.025	mono	88.200
16	44.100	stereo	1.411.200

Tabell 1: Bit rater for audio av forskjellig kvalitet

Som tabell 1 viser er kostnaden for audio av høy kvalitet høy bit rate. Audio-informasjon av høy kvalitet forbruker store mengder av datamaskinlagring og overføringskapasitet. Mange datamaskiner og datamaskinnettverk mangler ressursene for å prosessere rå digital audio. Kompresjon (også kalt koding (encoding) eller (coding)) reduserer kostnaden med lagring og overføring av audioinformasjon ved konvertering av informasjonen til en form med lavere bit rate. Kompresjon kan være tapløs (hvor kvalitet ikke lider) eller tapsbeheftet (hvor kvalitet lider, men reduksjon i bit rate fra etterfølgende tapløs kompresjon er mer dramatisk). Dekompresjon (også kalt dekodning) ekstraherer en rekonstruert versjon av den opprinnelige informasjon fra den komprimerte form. En kodek er et koder/dekoder-system.

II. Talekodere og -dekodere

Et mål med audiokompresjon er å digitalt representere audiosignaler for å tilveiebringe maksimum signalkvalitet for en gitt mengde bits. Sagt på en annen måte, dette mål er å representere audiosignalene med færrest bits for et gitt nivå av kvalitet. Andre mål, så som robusthet mot overføringsfeil og begrensnig av den samlede forsinkelse på grunn av koding/overføring/dekodning gjelder i enkelte scenarier.

Forskjellige typer av audiosignaler har forskjellige karakteristika. Musikk karakteriseres ved store verdiområder av frekvenser og amplituder, og inkluderer ofte to eller flere kanaler. På den annen side, tale karakteriseres av mindre verdiområder av frekvenser og amplituder, og blir vanligvis representert i en enkelt kanal. Visse kodeker og prosesseringsteknikker er tilpasset til musikk og generell audio; andre kodeker og prosesseringsteknikker er tilpasset til tale.

En type av konvensjonell talekodek bruker lineær prediksjon ("LP") for å utføre kompresjon. Talekodingen inkluderer flere trinn. Koderen finner og kvantiserer koeffisienter for et lineært prediksjonsfilter som brukes til å predikere sampelverdier som lineære kombinasjoner av forutgående sampelverdier. Et restsignal (representert som et "eksitasjon"-signal) viser deler av det opprinnelige signal som ikke predikeres nøyaktig av filtreringen. På enkelte trinn bruker talekodeken forskjellige kompresjonsteknikker for stemte segmenter (karakterisert av stemmebåndvibrasjon), ustemte segmenter og stille segmenter, siden forskjellige typer av tale har forskjellige karakteristika. Stemte segmenter oppviser typisk høyt

repeterende stemmemønstre, selv i restdomenen. For stemte segmenter utfører koderen ytterligere kompresjon ved sammenligning av det inneværende restsignal med forutgående restsykluser og koding av det inneværende restsignal i form av forsinkelses- eller etterslepsinformasjon i forhold til de forutgående sykluser. Koderen

5 håndterer andre uoverensstemmelser mellom det opprinnelige signal og den predikerte, kodede representasjon (fra den lineære prediksjons- og forsinkelsesinformasjon) ved bruk av spesielt designede kodebøker.

EP 1 308 932 vedrører en fremgangsmåte og et apparat for behandling av et dekodete talesignal.

10 Selv om talekodeker som beskrevet ovenfor har god samlet ytelse for mange applikasjoner, har de flere ulemper. For eksempel reduserer tapsbeheftede kodeker typisk bit rate ved å redusere redundansen i et talesignal, hvilket resulterer i støy eller andre uønskede artefakter i dekodet tale. Enkelte kodekfilter dekodet følgelig tale for å forbedre sin kvalitet. Slike etterfilter har typisk kommet i to typer:

15 tidsdomene etterfilter og frekvensdomene etterfilter.

Gitt viktigheten av kompresjon og dekompresjon for å representere tale-signaler i datamaskinsystemer, er det ikke overraskende at etterfiltrering av rekonstruert tale har tiltrukket seg forskning. Uansett hvilke fordeler det er ved tidligere teknikker for prosessering av rekonstruert tale eller annen audio, har de ikke

20 fordelene ved de teknikker og verktøy som beskrives her.

Sammenfatning

Hovedtrekkene ved den foreliggende oppfinnelse fremgår av det selvstendige patentkrav 1. Ytterligere trekk ved oppfinnelsen er angitt i de selvstendige krav.

25 Sammenfattende er den detaljerte beskrivelse rettet mot forskjellige teknikker og verktøy for audiokodeker, og spesifikt mot verktøy og teknikker som er relatert til filtrering av dekodet tale. Beskrevne utførelser implementerer en eller flere av de beskrevne teknikker og verktøy, inkludert, men ikke begrenset til, det følgende:

I ett aspekt beregnes et sett av filterkoeffisienter for anvendelse på et rekonstruert audiosignal. Beregningen inkluderer gjennomføring av en eller flere frekvensdomeneberegninger. Et filtrert audiosignal produseres ved filtrering av i det minste en del av det rekonstruerte audiosignal i en tidsdomene ved bruk av settet av filterkoeffisienter.

30

I et annet aspekt produseres et sett av filterkoeffisienter for anvendelse på et rekonstruert audiosignal. Produksjon av koeffisientene inkluderer prosessering av et

35

sett av koeffisientverdier som representerer en eller flere topper og en eller flere daler. Prosessering av settet av koeffisientverdier inkluderer klipping av en eller flere av toppene eller dalene. I det minste en del av det rekonstruerte audiosignal filtreres ved bruk av filterkoeffisientene.

5 I et annet aspekt mottas et rekonstruert komposittsignal som er syntetisert fra flere rekonstruerte frekvens subbåndsignaler. Subbåndsignalene inkluderer et rekonstruert første frekvens subbåndsignal for et første frekvensbånd og et rekonstruert annen frekvens subbåndsignal for et annet frekvensbånd. Ved et frekvensområde rundt en skjæring mellom det første frekvensbånd og det annet
10 frekvensbånd, blir det rekonstruerte komposittsignal selektivt forbedret.

De forskjellige teknikker og verktøy kan brukes i kombinasjon eller uavhengig. Ytterligere trekk og fordeler vil klart fremgå av den følgende detaljerte beskrivelse av forskjellige utførelser som går fremover med henvisning til de ledsagende tegninger.

15 **Kort beskrivelse av tegningene**

Figur 1 er et blokkdiagram over en passende databehandlingsomgivelse hvor en eller flere av de beskrevne utførelser kan implementeres.

Figur 2 er et blokkdiagram over en nettverksomgivelse i forbindelse med hvilken en eller flere av de beskrevne utførelser kan implementeres.

20 Figur 3 er en graf som viser en mulig frekvens subbåndstruktur som kan brukes til subbåndkoding.

Figur 4 er et blokkdiagram over en sanntids talebåndkoder i forbindelse med hvilken en eller flere av de beskrevne utførelser kan implementeres.

25 Figur 5 er et flytdiagram som viser bestemmelse av kodebokparametere i en implementering.

Figur 6 er et blokkdiagram over en sanntids talebånddekoder i forbindelse med hvilken en eller flere av de beskrevne utførelser kan implementeres.

Figur 7 er et flytdiagram som viser en teknikk for bestemmelse av etterfilter koeffisienter som kan brukes i enkelte implementeringer.

30

Detaljert beskrivelse

Beskrevne utførelser er rettet mot teknikker og verktøy for prosessering av audioinformasjon i koding og/eller dekoding. Med disse teknikker blir kvaliteten av tale som er avledet fra en talekodek, så som en sanntids talekodek, forbedret. Slike

forbedringer kan være resultat av bruk av forskjellige teknikker og verktøy separat eller i kombinasjon.

Slike teknikker og verktøy kan inkludere et etterfilter som anvendes på et dekodet audiosignal i tidsdomenen ved bruk av koeffisienter som er designet eller som prosesseres i frekvensdomenen. Teknikkene kan også inkludere klipping av eller å sette lokk på filterkoeffisientverdier til bruk i et slikt filter, eller i en annen type av etterfilter.

Teknikkene kan også inkludere et etterfilter som øker størrelsen av et dekodet audiosignal ved frekvensområder hvor energien kan ha blitt dempet på grunn av dekomposisjon i frekvensbånd. Som et eksempel kan filteret øke signalet ved frekvensområder nær skjæringer mellom tilstøtende bånd.

Selv om operasjoner for de forskjellige teknikker er beskrevet i en bestemt sekvensiell rekkefølge av hensyn til presentasjonen, skal det forstås at denne måten for beskrivelse omfatter mindre omarrangeringer i rekkefølgen av operasjoner, med mindre en bestemt disposisjon er påkrevd. For eksempel kan operasjoner som er beskrevet sekvensielt i enkelte tilfelle omarrangeres eller gjennomføres samtidig. Dessuten, for enkelthets skyld, kan det være at flytskjemaer ikke viser de forskjellige måter som bestemte teknikker kan brukes på i forbindelse med andre teknikker.

Selv om bestemte databehandlingsomgivelsestrekk og audiokodektrekk er beskrevet nedenfor, kan et eller flere av verktøyene og teknikkene brukes sammen med forskjellige ulike typer av databehandlingsomgivelser og/eller forskjellige ulike typer av kodeker. For eksempel kan en eller flere av etterfilter teknikkene brukes sammen med kodeker som ikke bruker CELP-kodingsmodellen, så som adaptive differensial puls kodemodulasjonskodeker, transformasjonskodeker og/eller andre typer av kodeker. Som et annet eksempel kan en eller flere av etterfilter teknikkene brukes sammen med enkeltbåndkodeker eller subbåndkodeker. Som et annet eksempel kan en eller flere av etterfilter teknikkene anvendes på en enkeltbåndkodek i en multibåndkodek og/eller på et syntetisert eller ukodet signal som inkluderer bidrag fra flere bånd i en multibåndkodek.

I. Databehandlingsomgivelse

Figur 1 illustrerer et generalisert eksempel på en passende databehandlingsomgivelse 100 hvor en eller flere av de beskrevne utførelser kan implementeres. Det er ikke meningen at det med databehandlingsomgivelsen 100 skal foreslås noen begrensning på omfanget av bruk eller funksjonalitet av oppfinnelsen, ettersom den

foreliggende oppfinnelse kan implementeres i diverse databehandlingsomgivelser for generelt formål eller spesielt formål.

Med henvisning til figur 1 inkluderer databehandlingsomgivelsen 100 minst én prosesseringsenhet 110 og minne 120. På figur 1 er denne mest grunnleggende konfigurasjon 130 inkludert innenfor en stiplet linje. Prosesseringsenheten 110 utfører datamaskinutførbar instruksjon, og kan være en virkelig eller en virtuell prosessor. I et multiprosesseringssystem utfører flere prosesseringsenheter datamaskinutførbare instruksjoner for å øke prosesseringskraft. Minnet 120 kan være et flyktig minne (eksempelvis registre, cache, RAM), ikke-flyktig minne (eksempelvis ROM, EEPROM, flash minne, osv), eller en kombinasjon av disse to. Minnet 120 lagrer programvare 180 som implementerer en eller flere av de etterfiltrerings teknikker som her er beskrevet for en taledekoder.

En databehandlingsomgivelse 100 kan ha ytterligere trekk. På figur 1 inkluderer databehandlingsomgivelsen 100 lagring 140, en eller flere innmatingsinnretninger 150, en eller flere utgangsinnetninger 160 og en eller flere kommunikasjonssforbindelser 170. En sammenkoplingsmekanisme (ikke vist), så som en buss, kontroller eller et nettverk, kopler sammen komponentene i databehandlingsomgivelsen 100. Operativsystemprogramvare (ikke vist) tilveiebringer typisk en operasjonell omgivelse for annen programvare som utføres i databehandlingsomgivelsen 100, og koordinerer aktiviteter til komponentene i databehandlingsomgivelsen 100.

Lagringen 140 kan være uttakbar eller ikke-uttakbar, og kan inkludere magnetiske disk, magnetiske bånd eller kassetter, CD-ROMer, CD-RWer, DVDer eller et hvilket som helst annet medium som kan brukes til å lagre informasjon og som kan aksesseres innenfor databehandlingsomgivelsen 100. Lagringen 140 lagrer informasjon for programvaren 180.

Innmatingsinnretningen(e) 150 kan være en berøringsinnmatingsinnretning så som et tastatur, mus, penn eller styrekule, en stemmeinmatingsinnretning, en skanningsinnretning, nettverkadapter eller en hvilken som helst annen innretning som tilveiebringer innmating til databehandlingsomgivelsen 100. For audio kan innmatingsinnretningen(e) 150 være et lydkort, mikrofon eller annen innretning som tar imot audioinnmating i analog eller digital form, eller en CD/DVD-leser som tilveiebringer audiosampler til databehandlingsomgivelsen 100. Utgangsinnetningen(e) 160 kan være et display, en printer, en høyttaler, CD/DVD-skriver,

nettverkadapter eller en hvilken som helst annen innretning som tilveiebringer utgang fra databehandlingsomgivelsen 100.

Kommunikasjonsforbindelsen(e) 170 muliggjør kommunikasjon over et kommunikasjonsmedium til en annen databehandlingsentitet. Kommunikasjonsmediet fører informasjon, så som datamaskinutførbare instruksjoner, komprimert taleinformasjon eller andre data i et modulert datasignal. Et modulert datasignal er et signal som har ett eller flere av sine karakteristika fastsatt eller forandret på en slik måte at informasjon kodes i signalet. Som eksempel, og ikke begrensning, kommunikasjonsmedia inkluderer ledningsførte eller trådløse teknikker som er implementert med en elektrisk, optisk, RF, infrarød, akustisk eller annen bærer.

Oppfinnelsen kan beskrives i den generelle kontekst ved datamaskinlesbare media. Datamaskinlesbare media er et hvilket som helst tilgjengelig media som kan aksesseres innenfor en databehandlingsomgivelse. Som eksempel, og ikke begrensning, med databehandlingsomgivelsen 100, inkluderer datamaskinlesbare media minne 120, lagring 140, kommunikasjonsmedia og kombinasjoner av hvilket som helst av det ovenstående.

Oppfinnelsen kan beskrives i den generelle kontekst med datamaskinutførbare instruksjoner, så som de som er inkludert i programmoduler, som utføres i en databehandlingsomgivelse på en virkelig eller virtuell målprosessor. Programmoduler inkluderer generelt rutiner, programmer, biblioteker, objekter, klasser, komponenter, datastrukturer, osv, som gjennomfører bestemte oppgaver eller implementerer bestemte abstrakte datatyper. Funksjonaliteten til programmodulene kan være kombinert eller delt mellom programmoduler, etter ønske i forskjellige utførelser. Datamaskinutførbare instruksjoner for programmoduler kan utføres innenfor en lokal eller distribuert databehandlingsomgivelse.

For presentasjonens skyld kan den detaljerte beskrivelse bruke uttrykk så som "bestemme" eller "fastlegge", "generere", "justere" og "anvende" for å beskrive datamaskinoperasjoner i en databehandlingsomgivelse. Disse uttrykk er høynivå abstraksjoner for operasjoner som gjennomføres av en datamaskin, og bør ikke blandes sammen med handlinger som gjennomføres av et menneske. De faktiske datamaskinoperasjoner som korresponderer til disse uttrykk varierer i avhengighet av implementeringen.

II. Generalisert nettverksomgivelse for sanntids talekodek

Figur 2 er et blokkdiagram over en generalisert nettverksomgivelse 200 i forbindelse med hvilken en eller flere av de beskrevne utførelser kan implementeres. Et nettverk 250 separerer forskjellige komponenter på kodersiden fra forskjellige komponenter på dekodersiden.

5 De primære funksjoner ved kodersidens og dekodersidens komponenter er talekoding henholdsvis dekoding. På dekodersiden tar en inngangsbuffer 210 imot og lagrer taleinnmating 202. Talekoderen 230 tar taleinnmating 202 fra inngangsbufferen 210 og koder den.

10 Spesifikt, en rammesplitter 212 deler samplene av taleinnmatingen 202 i rammer. I en implementering er rammene uniformt tjue ms lange - 160 sampler for åtte kHz innmating og 320 sampler for 16 kHz innmating. I andre implementeringer har rammene forskjellige varigheter, er ikke-uniforme eller overlappende, og/eller samplingsraten til innmatingen 202 er forskjellig. Rammene kan være organisert i en superramme/ramme, ramme/subramme, eller en annen konfigurasjon for forskjellige trinn i kodingen og dekodingen.

15 En rammesorterer 214 klassifiserer rammene i henhold til ett eller flere kriterier, så som energi i signalet, null kryssingshastighet, langtids prediksjonsforsterkning, forsterkningsdifferensiale og/eller andre kriterier for subrammer eller de hele rammer. Basert på kriteriene klassifiserer rammesorteren de forskjellige rammer i klasser, så som stille, ustemt, stemt og overgang (eksempelvis ustemt til stemt). I tillegg kan rammene klassifiseres i henhold til typen av redundant koding, hvis det er noen, som brukes for rammen. Rammeklassen påvirker de parametere som vil bli beregnet for å kode rammen. I tillegg kan rammeklassen påvirke oppløsningen og tapsrobustheten med hvilken parametere kodes, for å tilveiebringe mer
20 oppløsning og tapsrobusthet til flere viktige rammeklasser og parametere. For eksempel, stille rammer blir typisk kodet ved svært lav rate, de er svært enkle å gjenvinne ved skjuling hvis de mistes, og de trenger kanskje ikke beskyttelse mot tap. Ustemte rammer blir typisk kodet med litt høyere rate, de er akseptabelt enkle å gjenvinne ved skjuling hvis de mistes, og de blir ikke vesentlig beskyttet mot tap.
25 Stemte rammer og overgangsrammer blir vanligvis kodet med flere bits, i avhengighet av kompleksiteten av rammen så vel som tilstedeværelsen av overganger. Stemte rammer og overgangsrammer er også vanskelige å gjenvinne hvis de mistes, og blir således beskyttet vesentlig mer mot tap. Alternativt bruker rammesorteren 214 andre og/eller ytterligere rammeklasser.

Inngangstalesignalet kan deles opp i subbåndsignaler før anvendelse av en kodingsmodell, så som CELP-kodingsmodellen, på subbåndinformasjonen for en ramme. Dette kan gjøres ved bruk av en serie av en eller flere analysefilterbanker (så som QMF-analysefiltre) 216. For eksempel, hvis det skal brukes en tre-båndstruktur, så kan det lave frekvensbåndet deles ved å sende signalet gjennom et lavpassfilter. Likeledes kan det høye båndet deles ved å sende signalet gjennom et høypassfilter. Det midtre båndet kan deles ved å sende signalet gjennom et båndpassfilter, hvilket kan inkludere et lavpassfilter og et høypassfilter i serie. Alternativt kan det brukes andre typer av filterarrangementer for subbånddekomposisjon og/eller tidsstyring av filtrering (eksempelvis før rammedeling). Hvis kun ett bånd skal dekodes for en del av signalet, kan denne delen gå utenom analysefilterbankene 216.

Antallet bånd n kan bestemmes av samplingsrate. For eksempel, i en implementering, brukes en enkeltbåndstruktur for 8 kHz samplingsrate. For 16 kHz og 22,05 kHz samplingsrater, brukes en tre-båndstruktur, som vist på figur 3. I tre-båndstrukturen på figur 3 strekker det lave frekvensbånd 310 seg over halvparten av den fulle båndbredde F (fra 0 til $0,5 F$). Den andre halvdel av båndbredden er delt likt mellom det midtre bånd 320 og det høye bånd 330. Nær skjæringene mellom båndene reduseres frekvensresponsen for ett bånd gradvis fra gjennomgangsnivået til stoppnivået, hvilket er karakterisert ved en demping av signalet på begge sider i nærheten av skjæringen. Andre oppdeler av frekvensbåndbredden kan også brukes. For eksempel, for en samplingsrate på 32 kHz, kan det brukes en fire-båndstruktur med lik innbyrdes avstand.

Det lave frekvensbåndet er typisk det viktigste bånd for talesignaler, fordi signalenergien typisk svekkes mot de høyere frekvensområder. Det lave frekvensbåndet er følgelig ofte kodet ved bruk av flere bits enn de andre båndene. Sammenlignet med en enkeltbåndkodingsstruktur, er subbåndstrukturen mer fleksibel, og tillater bedre styring av kvantiseringsstøy over frekvensbåndet. Det antas følgelig at perseptuell stemmekvalitet forbedres signifikant ved bruk av subbåndstrukturen. Imidlertid, som omtalt nedenfor, dekomposisjonen av subbåndene kan forårsake energitap for signalet ved frekvensområdene nær skjæringen mellom tilstøtende bånd. Denne energi kan forringe kvaliteten til det resulterende dekodete talesignal.

På figur 2 er hvert subbånd kodet separat, som illustrert med kodingskomponenter 232, 234. Selv om båndkodingskomponentene 232, 234 er vist separat, kan kodingen av alle båndene gjøres ved hjelp av en enkelt koder, eller de kan kodes

av separate kodere. Slik båndkoding er beskrevet i nærmere detalj nedenfor med henvisning til figur 4. Alternativt kan kodeken operere som en enkeltbåndkodek. Den resulterende kodede tale tilveiebringes til programvare for ett eller flere nettverkslag 240 gjennom en multiplekser ("MUX") 236. Nettverkslaget/-lagene 240 prosesserer den kodede tale for overføring over nettverket 250. For eksempel, nettverkslagprogramvaren pakker rammer med kodet taleinformasjon i pakker som følger RTP-protokollen, hvilke formidles over Internett ved bruk av UDP, IP og forskjellige protokoller for fysiske lag. Alternativt brukes andre og/eller ytterligere lag med programvare eller nettverksprotokoller.

Nettverket 250 er et pakke-svitsjet nettverk så som Internett. Alternativt er nettverket 250 et lokalnett eller et annet slags nettverk.

På dekodersiden mottar og prosesserer programvare for ett eller flere nettverkslag 260 de overførte data. Nettverksprotokoller, transportprotokoller og høyere lags protokoller og programvare i nettverkslaget/-lagene 260 på dekodersiden korresponderer vanligvis til de som er i nettverkslaget/-lagene 240 på kodersiden. Nettverkslaget/-lagene tilveiebringer den kodede taleinformasjon til taledekoderen 270 gjennom en demultiplekser ("DEMUX") 276.

Dekoderen 270 dekoder hvert av subbåndene separat, hvilket er vist i bånddekodingskomponenter 272, 274. Alle subbåndene kan dekodes ved hjelp av enkelt dekker, eller de kan dekodes ved hjelp av dekodere for separate bånd.

De dekodede subbånd blir deretter syntetisert i en serie av en eller flere syntesefilterbanker (så som QMF-syntesefiltre) 280, som mater ut dekodet tale 292. Alternativt brukes andre typer av filterarrangementer for subbåndsyntese. Hvis kun ett enkelt bånd er til stede, så kan det dekodede bånd gå utenom filterbankene 280. Hvis flere bånd er til stede, kan dekodet taleutgang 292 også sendes gjennom et midtfrekvens forbedrings etterfilter 284 for å forbedre kvaliteten av den resulterende forbedrede taleutgang 294. En implementering av midtfrekvens forbedrings etterfiltret er omtalt i nærmere detalj nedenfor.

En generalisert sanntids talebånddekker er beskrevet nedenfor med henvisning til figur 6, men andre taledekodere kan brukes isteden. I tillegg kan enkelte av eller alle de beskrevne verktøy og teknikker brukes sammen med andre typer av audiokodere og -dekodere, så som musikkodere og -dekodere, eller audiokodere og -dekodere for generelt formål.

Bortsett fra disse primære kodings- og dekodingsfunksjoner kan komponentene også dele informasjon (vist i stiplede linjer på figur 2) for å styre raten, kvaliteten

og/eller tapsrobustheten for den kodede tale. Ratekontrolleren 220 betrakter et mangfold av faktorer, så som kompleksiteten av den inneværende innmating i innmatingsbufferen 210, bufferfullheten for utgangsbuffere i 230 eller et annet sted, ønsket utgangsrate, den inneværende nettverksbåndbredde, nettverks-

5 opphopning/støybetingelser og/eller dekoderens tapsrate. Dekoderen 270 mater tilbake dekodertapsrateinformasjon til ratekontrolleren 220. Nettverkslaget/-lagene 240, 260 samler inn eller estimerer informasjon om inneværende nettverksbåndbredde og opphopning/støybetingelser, hvilken mates tilbake til ratekontrolleren 220. Alternativt betrakter ratekontrolleren 220 andre og/eller ytterligere faktorer.

10 Ratekontrolleren 220 styrer talekoderen 230 for å forandre rate, kvalitet og/eller tapsrobusthet som tale kodes med. Koderen 230 kan forandre rate og kvalitet ved justering av kvantiseringsfaktorer for parametere eller forandring av oppløsningen av entropikoder som representerer parameterne. Koderen kan i tillegg forandre tapsrobusthet ved justering av raten eller typen av redundant koding.

15 Koderen 230 kan således forandre allokeringen av bits mellom primære kodingsfunksjoner og tapsrobusthetsfunksjoner, avhengig av nettverksbetingelser.

Figur 4 er et blokkdiagram over en generalisert talebåndkoder 400 i forbindelse med hvilken en eller flere av de beskrevne utførelser kan implementeres. Båndkoderen 400 korresponderer generelt til en hvilken som helst av båndkodings-

20 komponentene 232, 234 på figur 2.

Båndkoderen 400 tar imot båndinnmatingen 402 fra filterbankene (eller andre filtre) hvis signalet er delt i flere bånd. Hvis signalet ikke er delt i flere bånd, så inkluderer båndinnmatingen 402 sampler som representerer hele båndbredden. Båndkoderen produserer kodet båndutgang 492.

25 Hvis et signal er delt i flere bånd, så kan en nedsamlingskomponent 420 gjennomføre nedsampling på hvert bånd. Som et eksempel, hvis samplingsraten er satt til seksten kHz og hver ramme er tjue ms i varighet, så inkluderer hver ramme 320 sampler. Hvis ingen nedsampling ble gjennomført og rammen ble delt i den trebåndstruktur som er vist på figur 3, så ville tre ganger så mange sampler (det vil si

30 320 sampler per bånd, eller totalt 960 sampler) bli kodet og dekodet for rammen. Hvert bånd kan imidlertid nedsamples. For eksempel kan det lave frekvensbånd 310 nedsamples fra 320 sampler til 160 sampler, og hvert av de midtre bånd 320 og høye bånd 330 kan samples ned fra 320 sampler til 80 sampler, hvor båndene 310, 320, 330 strekker seg henholdsvis over halvparten, en kvart og en kvart av

35 frekvensområdet. (Graden av nedsampling 420 i denne implementering varierer i

forhold til frekvensområdene for båndene 310, 320, 330. Andre implementeringer er imidlertid mulige. I senere trinn brukes typisk færre bit for de høyere bånd, fordi signalenergien typisk faller mot de høyere frekvensområder). Dette tilveiebringer følgelig totalt 320 sampler som skal kodes og dekodes for rammen.

5 LP-analysekomponenten 430 beregner lineære prediksjonskoeffisienter 432. I en implementering bruker LP-filteet ti koeffisienter for åtte kHz inngang og 16 koeffisienter for 16 kHz inngang, og LP-analysekomponenten 430 beregnet et sett av lineære prediksjonskoeffisienter per ramme for hvert bånd. Alternativt beregner LP-analysekomponenten 430 to sett av koeffisienter per ramme for hvert bånd, en for 10 hver av to vinduer sentrert ved forskjellige lokaliseringer, eller beregner et forskjellig antall av koeffisienter per bånd og/eller per ramme.

LPC-prosesseringskomponenten mottar og prosesserer de lineære prediksjonskoeffisienter 432. LPC-prosesseringskomponenten 435 konverterer typisk LPC-verdier til en forskjellig representasjon for mer effektiv kvantisering og koding. 15 For eksempel konverterer LPC-prosesseringskomponenten 435 LPC-verdier til en linjespektralpar (line spectral pair, LSP) representasjon, og LSP-verdiene kvantiseres (så som ved hjelp av vektorkvantisering) og kodes. LSP-verdiene kan være intrakodet eller predikert fra andre LSP-verdier. Forskjellige representasjoner, kvantiseringsteknikker og kodingsteknikker er mulig for LPC-verdier. LPC-verdiene er 20 tilveiebrakt i en form som del av den kodede båndutgang 492 for pakketisering og overføring (sammen med eventuelle kvantiseringsparametere og annen informasjon som er nødvendig for rekonstruksjon). For etterfølgende bruk i koderen 400 rekonstruerer LPC-prosesseringskomponenten 435 LPC-verdiene. LPC-prosesseringskomponenten 435 kan gjennomføre interpolasjon for LPC-verdier (så 25 som ekvivalent i LSP-representasjon eller en annen representasjon) for å glatte overgangene mellom forskjellige sett av LPC-koeffisienter, eller mellom de LPC-koeffisienter som brukes for forskjellige subrammer av rammer.

Syntesefilteret (eller "korttids prediksjonsfilteret") 440 tar imot rekonstruerte LPC-verdier 438 og inkorporerer dem i filteret. Syntesefilteret 440 mottar et 30 eksitasjonssignal og produserer en approksimasjon av det opprinnelige signal. For en gitt ramme kan syntesefilteret 440 bufre et antall av rekonstruerte sampler (eksempelvis ti for et ti-tapsfilter) fra den forutgående ramme for starten av prediksjonen.

De perseptuelle vektingskomponenter 450, 455 anvender perseptuell vektning 35 på det opprinnelige signal og den modellerte utgang fra syntesefilteret 440, for

selektivt å legge mindre vekt på formant-strukturen til talesignaler, for å gjøre hørrelssystemene mindre sensitive for kvantiseringsfeil. De perseptuelle vektingskomponenter 450, 455 nyttiggjør seg psykoakustiske fenomener, så som maskering. I en implementering anvender de perseptuelle vektingskomponenter 450, 455 vektorer basert på de opprinnelige LPC-verdier 432 som er mottatt fra LP-analysekomponenten 430. Alternativt anvender de perseptuelle vektingskomponenter 450, 455 andre og/eller ytterligere vektorer.

Etter de perseptuelle vektingskomponenter 450, 455, beregner koderen 400 differansen mellom det perseptuelt vektete opprinnelige signal og perseptuelt vektete utgang fra syntesefilteret 440, for å produsere et differansesignal 434. Alternativt bruker koderen 400 en forskjellig teknikk for å beregne taleparameterne.

Eksitasjons parameteriseringskomponenten 460 søker å finne den beste kombinasjon av adaptive kodebokindekser, fast kodebokindekser og forsterkningskodebokindekser i form av å minimere differansen mellom det perseptuelt vektete opprinnelige signal og syntetiserte signal (uttrykt ved vektet middelkvadratfeil eller andre kriterier). Mange parametere beregnes per subramme, men mer generelt kan parameterne være per superramme, ramme eller subramme. Som omtalt ovenfor kan parameterne for forskjellige bånd i en ramme eller subramme være forskjellige. Tabell 2 viser de tilgjengelige typer av parametere for forskjellige rammeklasser i en implementering.

Rammeklasse	Parameter(e)
Stille	Klasseinformasjon; LSP; forsterkning (per ramme for generert støy)
Ustemt	Klasseinformasjon; LSP; puls, tilfeldige og forsterkningskodebokparametere
Stemt	Klasseinformasjon; LSP; adaptive, puls, tilfeldige og forsterkningskodebokparametere (per subramme)
Overgang	

Tabell 2: Parametere for forskjellige rammeklasser

På figur 4 deler eksitasjons parameteriseringskomponenten 460 rammen i subrammer og beregner kodebokindekser og forsterkninger for hver subramme som passende. For eksempel kan det antall og type av kodeboktrinn som skal brukes, og oppløsningene til kodebokindeksene, initialt bestemmes ved hjelp av en kodingsmodus, hvor modusen er diktert av ratekontrollkomponenten som er omtalt

ovenfor. En bestemt modus kan også diktere kodings- og dekodingsparametere andre enn antall og type av kodeboktrinn, for eksempel oppløsningen av kodebokindeksene. Parameterne for hvert kodeboktrinn bestemmes ved optimering av parameterne for å minimere feil mellom et målsignal og bidraget fra dette

5 kodeboktrinnet til det syntetiserte signal. (Som her brukt betyr uttrykket "optimere" å finne en passende oppløsning under anvendbare skranker, så som forvrengningsreduksjon, parametersøketid, parametersøkekompleksitet, bit rate for

10 parametere, osv, i motsetning til å gjennomføre et fullt søk på parameterrommet. Tilsvarende bør uttrykket "minimere" forstås med henblikk på å finne en passende oppløsning under anvendbare skranker.) Optimeringen kan for eksempel gjøres ved bruk av en modifisert middelkvadratfeilteknikk. Målsignalet for hvert trinn er differansen mellom restsignalet og summen av bidragene fra de forutgående kodeboktrinn, hvis det er noen, til det syntetiserte signal. Alternativt kan det brukes andre optimeringsteknikker.

15 Figur 5 viser en teknikk for bestemmelse av kodebokparametere i henhold til en implementering. Eksitasjons parameteriseringskomponenten 460 gjennomfører teknikken, potensielt i forbindelse med andre komponenter, så som en ratekontroller. Alternativt gjennomfører en annen komponent i en koder teknikken.

20 Med henvisning til figur 5, for hver subramme i en stemt ramme eller overgangsramme, bestemmer 510 eksitasjons parameteriseringskomponenten 460 om hvorvidt en adaptiv kodebok kan brukes for den inneværende subramme. (For eksempel kan ratestyringen diktere at ingen adaptiv kodebok skal brukes for en bestemt ramme.) Hvis den adaptive kodebok ikke skal brukes, så vil en adaptiv kodeboksvitsj at ingen adaptive kodebøker skal brukes 535. For eksempel kan dette

25 gjøres ved å sette et én bit flagg ved det rammenivå som viser at ingen adaptive kodebøker brukes i rammen, ved spesifisering av en bestemt kodingsmodus ved rammenivået, eller ved å sette et én bit flagg for hver subramme, hvilket viser at ingen adaptiv kodebok brukes i subrammen.

30 Med fortsatt henvisning til figur 5, hvis en adaptiv kodebok kan brukes, så fastlegger komponenten 460 adaptive kodebokparametere. Disse parametere inkluderer en indeks, eller pitch-verdi, som viser et ønsket segment av eksitasjons-signalhistorien, så vel som en forsterkning som skal anvendes på det ønskede segment. På figurene 4 og 5 gjennomfører komponenten 460 et lukket sløyfe pitch-søk 520. Dette søket begynner med den pitch som er bestemt av den valgfrie åpen

35 sløyfe pitch søkekomponent 425 på figur 4. En åpen sløyfe pitch søkekomponent

425 analyserer det vektete signal som produseres av vektingskomponenten 450 for å estimere dens pitch. Idet den begynner med denne estimerte pitch, optimerer det lukkede sløyfe pitch søk 520 pitch-verdien for å redusere feilen mellom målsignalet og det vektete syntetiserte signal som genereres fra et angitt segment av eksitasjonssignalhistorien. Den adaptive kodebok forsterkningsverdi blir også optimert 525. Den adaptive kodeboks forsterkningsverdi angir en multiplikator for påføring på de pitch predikerte verdier (verdiene fra det angitte segment i eksitasjonssignalhistorien) for å justere skalaen til verdiene. Den forsterkning som multipliseres med de pitch predikerte verdier er den adaptive kodebok sitt bidrag til eksitasjonssignalet for den inneværende ramme eller subramme.

Forsterkningsoptimeringen 525 og det lukkede sløyfe pitch søk 520 produserer en forsterkningsverdi henholdsvis en indeksverdi som minimerer feilen mellom målsignalet og det vektete syntetiserte signal fra den adaptive kodebok sitt bidrag.

Hvis komponenten 460 bestemmer 530 at den adaptive kodebok skal brukes, så signaleres 540 de adaptive kodebokparametere i bit strømmen. Hvis ikke, så angis det at ingen adaptiv kodebok brukes for subrammen 535, så som ved å sette et én bit subrammenivåflagg som omtalt ovenfor. Denne bestemmelse 530 kan inkludere bestemmelse av om hvorvidt den adaptive kodebok sitt bidrag til den bestemte subramme er signifikant nok til å være verdt det antall av bits som er påkrevd for å signalisere den adaptive kodebok sine parametere. Alternativt kan en annen basis brukes for bestemmelsen. Dessuten, selv om figur 5 viser signalering etter bestemmelsen, blir signaler alternativt bearbeidet satsvis inntil teknikken avsluttes for en ramme eller superramme.

Eksitasjons parameteriseringskomponenten 460 bestemmer også 550 om hvorvidt det brukes en pulskodebok. Bruken eller ikke bruken av pulskodeboken angis som en del av en samlet kodingsmodus for den inneværende ramme, eller den kan angis eller bestemmes på andre måter. En pulskodebok er en type av fast kodebok som spesifiserer en eller flere pulser som skal bidras til eksitasjonssignalet. Pulskodebokparametere inkluderer par av indekser og fortegn (forsterkninger kan være positive eller negative). Hvert par angir en puls som skal inkluderes i eksitasjonssignalet, hvor indeksen angir posisjonen til pulsen og fortegnet angir polariteten til pulsen. Det antall av pulser som er inkludert i pulskodeboken og som brukes som bidrag til eksitasjonssignalet kan variere i avhengighet av kodingsmodusen. I tillegg kan antallet av pulser avhenge av om hvorvidt eller ikke det brukes en adaptiv kodebok.

Hvis pulskodeboken brukes, så optimeres 555 pulskodebokparameterne for å minimere feil mellom bidraget fra de angitte pulser og et målsignal. Hvis en adaptiv kodebok ikke brukes, så er målsignalet det vektete opprinnelige signal. Hvis en adaptiv kodebok brukes, så er målsignalet differansen mellom det vektete opprinnelige signal og bidraget fra den adaptive kodebok til det vektete syntetiserte signal. Ved et punkt (ikke vist), blir pulskodebokparameterne deretter signalert i bit strømmen.

Eksitasjons parameteriseringskomponenten 460 bestemmer også 565 om hvorvidt noen tilfeldige faste kodeboktrinn skal brukes. Antallet (hvis noe) av de tilfeldige kodeboktrinn angis som del av en samlet kodingsmodus for den innværende ramme, eller det kan bestemmes på andre måter. En tilfeldig kodebok er en type av fast kodebok som bruker en forhåndsdefinert signalmodell for de verdier den koder. Kodebokparameterne kan inkludere utgangspunktet for et angitt segment av signalmodellen og et fortegn som kan være positivt eller negativt. Lengden eller verdiområdet av det angitte segment er typisk fast, og blir derfor ikke typisk signalert, men alternativt signaleres en lengde eller utstrekning av det angitte segment. En forsterkning multipliseres med verdiene i det angitte segment for å produsere bidraget fra den tilfeldige kodebok til eksitasjonssignalet.

Hvis det brukes minst ett tilfeldig kodeboktrinn, så optimeres 570 kodeboktrinnparameterne for kodeboken for å minimere feilen mellom bidraget fra det tilfeldige kodeboktrinn og et målsignal. Målsignalet er differansen mellom det vektete opprinnelige signal og summen av bidragene til det vektete syntetiserte signal fra den adaptive kodebok (hvis det er noen), pulskodeboken (hvis det er noen) og de tidligere bestemte tilfeldige kodeboktrinn (hvis det er noen). Ved et punkt (ikke vist), blir de tilfeldige kodebokparametere deretter signalert i bit strømmen.

Komponenten 460 bestemmer deretter 580 om hvorvidt noen flere tilfeldige kodeboktrinn skal brukes. I så fall blir parameterne for det neste tilfeldige kodeboktrinn optimert 570 og signaler som beskrevet ovenfor. Dette fortsetter inntil alle parameterne for de tilfeldige kodeboktrinn har blitt bestemt. Alle de tilfeldige kodeboktrinn kan bruke den samme signalmodell, selv om de trolig vil angi forskjellige segmenter fra modellen og ha forskjellige forsterkningsverdier. Alternativt kan forskjellige signalmodeller brukes for forskjellige tilfeldige kodeboktrinn.

Hver eksitasjonsforsterkning kan kvantiseres uavhengig eller to eller flere forsterkninger kan kvantiseres sammen, som det bestemmes av ratekontrolleren og/eller andre komponenter.

Selv om en bestemt rekkefølge her har blitt fremsatt for optimering av de forskjellige kodebokparametere, kan andre rekkefølger og optimeringsteknikker brukes. For eksempel kan alle tilfeldige kodebøker optimeres samtidig. Således, selv om figur 5 viser sekvensiell beregning av forskjellige kodebokparametere, blir

5 alternativt to eller flere forskjellige kodebokparametere optimert i fellesskap (eksempelvis ved i fellesskap å variere parameterne og evaluere resultatene i henhold til en ikke-lineær optimeringsteknikk). I tillegg kan andre konfigurasjoner av kodebøker eller eksitasjonssignalparametere brukes.

Eksitasjonssignalet i denne implementering er summen av eventuelle bidrag

10 fra den adaptive kodebok, pulskodeboken og den eller de tilfeldige kodeboktrinn. Alternativt kan komponenten 460 på figur 4 beregne andre og/eller ytterligere parametere for eksitasjonssignalet.

Med henvisning til figur 4 blir kodebokparametere for eksitasjonssignalet signalert eller på annen måte tilveiebrakt til en lokal dekoder 465 (innelukket av

15 stiplede linjer på figur 4) så vel som båndutgangen 492. Således, for hvert bånd, inkluderer koderutgangen 492 utgangen fra LPC-prosesseringskomponenten 435 som er omtalt ovenfor, så vel som utgangen fra eksitasjons parameteriseringskomponenten 460.

Bit raten for utgangen 492 avhenger delvis av de parametere som brukes av

20 kodebøkene, og koderen 400 kan styre bit rate og/eller kvalitet ved svitsjing mellom forskjellige sett av kodebokindekser, ved bruk av integrerte koder, eller ved bruk av andre teknikker. Forskjellige kombinasjoner av kodeboktypene og trinnene kan gi forskjellige kodingsmodi for forskjellige rammer, bånd og/eller subrammer. For eksempel kan en ustemt ramme bruke kun ett tilfeldig kodeboktrinn. En adaptiv

25 kodebok og en pulskodebok kan brukes for en stemt ramme med lav rate. En ramme med høy rate kan kodes ved bruk av en adaptiv kodebok, en pulskodebok og en eller flere tilfeldige kodeboktrinn. I en ramme kan kombinasjonen av alle kodingsmodi for alle subbåndene til sammen kalles et modussett. Det kan være flere

30 forhåndsdefinerte modussett for hver samplingsrate, med forskjellige modi som korresponderer til forskjellige kodings bit rater. Ratestyringsmodulen kan bestemme eller influere på den modus som er satt for hver ramme.

Med fortsatt henvisning til figur 4, utgangen fra eksitasjons parameteriseringskomponenten 460 mottas av kodebokrekonstruksjonskomponenter 470, 472, 474,

474, 476 og forsterkningsapplikasjonskomponenter 480, 482, 484, 486 som

35 korresponderer til de kodebøker som brukes av parameteriseringskomponenten 460.

Kodeboktrinnene 470, 472, 474, 476 og korresponderende forsterknings-applikasjonskomponenter 480, 482, 484, 486 rekonstruerer bidraget fra kodebøkene. Disse bidrag summeres for å produsere et eksitasjonssignal 490, som mottas av syntesefiltret 440, hvor det brukes sammen med de "predikerte" sampler fra hvilke etterfølgende lineær prediksjon skjer. Forsinkede deler av eksitasjonssignalet blir også brukt som et eksitasjonshistoriesignal av den adaptive kodebok-rekonstruksjonskomponent 470 for å rekonstruere etterfølgende adaptive kodebok-parametere (eksempelvis pitch bidrag), og av parameteriseringskomponenten 460 for beregning av etterfølgende adaptive kodebokparametere (eksempelvis pitch indeks og pitch forsterkningsverdier).

Med henvisning tilbake til figur 2, båndutgangen for hvert bånd tas imot av MUXen 236, sammen med andre parametere. Slike andre parametere kan blant annet inkludere informasjon, rammeklasseinformasjon 222 fra rammesorteren 214 og rammekodingsmodi. MUXen 236 konstruerer applikasjonslagpakker for sending til annen programvare, eller MUXen 236 putter dataene i nytte-dataene til pakker som følger en protokoll så som RTP. MUXen kan bufre parametere, for å tillate selektiv repetisjon av parametere for forover rettet feilkorreksjon i senere pakker. I en implementering pakker MUXen 236 den primære kodete taleinformasjon for en ramme i en enkelt pakke, sammen med forover rettet feilkorreksjonsinformasjon for alle eller en del av en eller flere forutgående rammer.

MUXen 236 tilveiebringer tilbakemelding, så som inneværende bufferfullhet for ratestyringsformål. Mer generelt kan forskjellige komponenter i koderen 230 (inkludert rammesorteren 214 og MUXen 236) tilveiebringe informasjon til en ratekontroller 220, så som den som er vist på figur 2.

Bit strømmen DEMUX 276 på figur 2 tar imot kodet taleinformasjon som inn-mating og analyserer den for å identifisere og prosessere parametere. Parametere kan inkludere rammeklasse, en representasjon av LPC-verdier og kode-bokparametere. Rammeklassen kan angi hvilke andre parametere som er tilstede for en gitt ramme. Mer generelt bruker DEMUXen 276 de protokoller som brukes av koderen 230 og ekstraherer de parametere som koderen 230 pakker i pakker. For pakker som mottas over et dynamisk pakkesvitsjet nettverk, inkluderer DEMUXen 276 en dirringsbuffer for å glatte ut korttids fluktusjoner i pakkerate over en gitt tidsperiode. I enkelte tilfeller regulerer dekodeeren 270 bufferforsinkelsen og administrerer når pakker leses ut fra bufferen, for å integrere forsinkelse, kvalitetskontroll, skjuling av manglende rammer, osv, i dekodning. I andre tilfeller

administrerer en applikasjonslagkomponent darringsbufferen, og darringsbufferen fylles ved en variabel rate og tømmes av dekoderen 270 ved en konstant eller relativt konstant rate.

5 DEMUXen 276 kan motta flere versjoner av parametere for et gitt segment, inkludert en primær kodet versjon og en eller flere sekundære feilkorreksjonsversjoner. Når feilkorreksjon mislykkes, bruker dekoderen 270 skjulingsteknikker, så som parameterrepetisjon eller estimering basert på informasjon som ble korrekt mottatt.

10 Figur 6 er et blokkdiagram over en generalisert sanntids talebånddekode 600 i forbindelse med hvilken en eller flere beskrevne utførelser kan implementeres. Bånddekode 600 korresponderer generelt til en hvilken som helst av bånddekodingskomponenter 272, 274 på figur 2.

15 Bånddekode 600 tar imot kodet taleinformasjon 692 for et bånd (som kan være det fullstendige båndet, eller ett av flere subbånd) som inngang og produserer en filtrert rekonstruert utgang 604 etter dekoding og filtrering. Komponentene i dekode 600 har korresponderende komponenter i koderen 400, men samlet sett er dekode 600 enklere, siden den mangler komponenter for perseptuell vektning, eksitasjonsprosesseringsløyfe og ratekontroll.

20 LPC-prosesseringskomponenten 635 mottar informasjon som representerer LPC-verdier i den form som er tilveiebrakt av båndkoderen 400 (så vel som eventuelle kvantiseringsparametere og annen informasjon som er nødvendig for rekonstruksjon). LPC-prosesseringskomponenten 635 rekonstruerer LPC-verdiene 638 ved bruk av de inverse av konverteringen, kvantiseringen, kodingen, osv, som tidligere ble anvendt på LPC-verdiene. LPC-prosesseringskomponenten 635 kan
25 også gjennomføre interpolasjon for LPC-verdier (i LPC-representasjon eller en annen representasjon, så som LSP) for å glatte overgangene mellom forskjellige sett av LPC-koeffisienter.

30 Kodeboktrinnene 670, 672, 674, 676 og forsterkningsapplikasjonskomponenter 680, 682, 684, 686 dekode parameterne i hvilke som helst av de korresponderende kodeboktrinn som brukes til eksitasjonssignalet og beregner bidraget fra hvert kodeboktrinn som brukes. Generelt korresponderer konfigurasjonen og operasjonene av kodeboktrinnene 670, 672, 674, 676 og forsterkningskomponentene 680, 682, 684, 686 til konfigurasjonen og operasjonene av kodeboktrinnene 470, 472, 474, 476 og forsterkningskomponentene 480, 482, 484, 486 i
35 koderen 400. Bidragene fra de brukte kodeboktrinn summeres, og det resulterende

eksitasjonssignal 690 mates inn i syntesefilteret 640. Forsinkede verdier av eksitasjonssignalet 690 brukes også som en eksitasjonshistorie av den adaptive kodebok 670 ved beregning av bidragene fra den adaptive kodebok for etterfølgende partier av eksitasjonssignalet.

5 Syntesefilteret 640 tar imot rekonstruerte LPC-verdier 638 og inkorporerer dem i filteret. Syntesefilteret 640 lagrer tidligere rekonstruerte sampler for prosessering. Eksitasjonssignalet 690 sendes gjennom syntesefilteret for å danne en approksimasjon av det opprinnelige talesignal.

10 Det rekonstruerte subbåndsignal 602 mates også inn i et korttids etterfilter 694. Korttids etterfilteret produserer en filtrert subbåndutgang 604. Flere teknikker for beregning av koeffisienter for korttids etterfilteret 694 er beskrevet nedenfor. For adaptiv etterfiltrering kan dekoderen 270 beregne koeffisientene fra parametere (eksempelvis LPC-verdier) for den kodede tale. Koeffisientene blir alternativt tilveiebrakt gjennom en annen teknikk.

15 Med henvisning tilbake til figur 2, som omtalt ovenfor, hvis det er flere subbånd, blir subbåndutgangen for hvert subbånd syntetisert i syntesefilterbankene 280 for å danne taleutgangen 292.

20 De relasjoner som er vist på figurene 2-6 angir generelle strømmer av informasjon, andre relasjoner er av hensyn til enkelheten ikke vist. Avhengig av implementering og den type av kompresjon som ønskes, kan komponenter adderes, utelates, deles i flere komponenter, kombineres med andre komponenter og/eller byttes ut med like komponenter. For eksempel, i omgivelsen 200 som er vist på figur 2, kan ratekontrolleren 220 kombineres med talekoderen 230. Potensielt tilførte komponenter inkluderer en multimediakodings- (eller avspillings-) applikasjon som administrerer talekoderen (eller dekoderen) så vel som andre kodere (eller dekodere) 25 og samler inn nettverks- og dekodertilstandsinformasjon, og som gjennomfører funksjoner med adaptiv feilkorreksjon. I alternative utførelser prosesserer forskjellige kombinasjoner og konfigurasjoner av komponenter taleinformasjon ved bruk av de teknikker som her er beskrevet.

30

III. Etterfilter teknikker

35 I enkelte utførelser anvender en dekode eller et annet verktøy et korttids etterfilter for å rekonstruere audio, så som rekonstruert tale, etter at den har blitt dekodet. Et slikt filter kan forbedre den perseptuelle kvalitet av den rekonstruerte tale.

Etterfiltre er typisk enten tidsdomene etterfiltre eller frekvensdomene etterfiltre. Et konvensjonelt tidsdomene etterfilter for en CELP kodek inkluderer et alle-pol lineært prediksjonskoeffisient syntesefilter som er skalert med én konstant faktor og et alle-null lineært prediksjonskoeffisient inversfilter som er skalert med en annen konstant faktor.

I tillegg opptrer i mange talesignaler et fenomen som er kjent som "spektral helling", fordi amplitudene til lavere frekvenser i normal tale ofte er høyere enn amplitudene til høyere frekvenser. Frekvensdomene amplitudespektrumet for et talesignal inkluderer således ofte en skråning, eller "helling". Den spektrale helling fra den opprinnelige tale bør således være tilstede i et rekonstruert talesignal. Imidlertid, hvis koeffisienter for et etterfilter også inkorporerer en slik helling, så vil effekten av hellingen bli forstørret i etterfiltrets utgang, slik at det filtrerte talesignal vil bli forvrengt. Enkelte tidsdomene etterfiltre har således også et første ordens høypassfilter for å kompensere for spektral helling.

Karakteristikkene ved tidsdomene etterfiltrene blir derfor typisk styrt av to eller tre parametere, hvilket ikke tilveiebringer mye fleksibilitet.

Et frekvensdomene etterfilter har, på den annen side, en mer fleksibel måte til å definere etterfilter karakteristikaene. I et frekvensdomene etterfilter, bestemmes filterkoeffisientene i frekvensdomenen. Det dekodete talesignal blir transformert til frekvensdomenen, og filtreres i frekvensdomenen. Det filtrerte signal blir deretter transformert tilbake til tidsdomenen. Det resulterende filtrerte tidsdomenesignal har imidlertid typisk et forskjellig antall av sampler enn det opprinnelige, ufiltrerte tidsdomenesignal. For eksempel kan en ramme som har et 160 sampler konverteres til frekvensdomenen ved bruk av en 256-punkters transformasjon, så som en 256-punkters hurtig Fourier-transformasjon (fast Fourier transform, "FFT"), etter utfylling eller inkludering av senere sampler. Når en 256-punkters invers FFT anvendes til å konvertere rammen tilbake til tidsdomenen, vil dette gi 256 tidsdomene sampler. Det gir derfor nittiseks ekstra sampler. De ekstra nittiseks sampler kan overlappes, med og adderes til, respektivt sampler i de første nittiseks sampler i den neste ramme. Dette blir ofte referert til som overlappings-adderingsteknikken. Transformasjonen av talesignalet, så vel som implementeringen av teknikker så som overlappings-adderingsteknikken kan signifikant øke kompleksiteten ved den samlede dekodere, særlig for kodeker som ikke allerede inkluderer frekvens transformasjonskomponenter. Frekvensdomene etterfiltre blir følgelig typisk kun brukt for sinusbaserte talekodeker fordi anvendelsen av slike filtre på ikke-sinusbaserte kodeker innfører for

mye forsinkelse og kompleksitet. Frekvensdomene etterfiltre har også typisk mindre fleksibilitet til å forandre rammestørrelse hvis kodekrammestørrelsen varierer under koding, fordi kompleksiteten ved overlappings-adderingsteknikken som er omtalt ovenfor kan bli hindrende hvis en ramme av forskjellige størrelse (så som en ramme med 80 sampler, istedenfor 160 sampler) påtreffes.

Selv om bestemte databehandlingsomgivelsestrekk og audiokodektrekk er beskrevet ovenfor, kan et eller flere av verktøyene og teknikkene brukes sammen med forskjellige ulike typer av databehandlingsomgivelser og/eller forskjellige ulike typer av kodeker. For eksempel kan en eller flere av etterfilterteknikkene brukes sammen med kodeker som ikke bruker CELP-kodingsmodellen, så som adaptive differensial pulskodemodulasjonskodeker, transformasjonskodeker og/eller andre typer av kodeker. Som et annet eksempel kan en eller flere av etterfilterteknikkene brukes sammen med enkeltbåndkodeker eller subbåndkodeker. Som et annet eksempel kan en eller flere av etterfilterteknikkene anvendes på et enkelt bånd i en multibåndkodek og/eller på et syntetisert eller ukodet signal som inkluderer bidrag fra flere bånd i en multibåndkodek.

A. Eksempel hybrid korttids etterfiltre

I enkelte utførelser inkorporerer den dekodek, så som dekodek 600 som er vist på figur 6, et adaptivt, tidsfrekvens "hybrid" filter for etterprosessering, eller et slikt filter anvendes på utgangen fra dekodek 600. Alternativt er et slikt filter inkorporert i eller anvendes på utgangen fra en annen type av audiodekoder eller prosesseringsverktøy, for eksempel en talekodek som er beskrevet et annet sted i den foreliggende søknad.

Med henvisning til figur 6, i enkelte implementeringer er korttids etterfiltret 694 et "hybrid"-filter som er basert på en kombinasjon av tidsdomene- og frekvensdomeneprosesser. Koeffisientene til etterfiltret 694 kan fleksibelt og effektivt designet primært i frekvensdomenen, og koeffisientene kan anvendes på korttids etterfiltret 694 i tidsdomenen. Kompleksiteten ved denne løsningsmåte er typisk lavere enn standard frekvensdomene etterfiltre, og den kan implementeres på en måte som innfører neglisjerbar forsinkelse. Filtret kan i tillegg tilveiebringe mer fleksibilitet enn tradisjonelle tidsdomene etterfiltre. Det antas at et slikt hybridfilter signifikant kan forbedre utgangstalekvaliteten uten at dette krever overdrevent stor forsinkelse eller dekodek kompleksitet. I tillegg, fordi filtret 694 anvendes i tidsdomenen, kan det anvendes på rammer av en hvilken som helst størrelse.

Etterfiltret 694 kan generelt være et filter med endelig impulsrespons (finite impulse response, "FIR"), hvis frekvensrespons er resultatet av ikke-lineære prosesser som gjennomføres på logaritmen til et størrelsesspektrum for et LPC-syntesefilter. Størrelsesspektrumet til etterfiltret kan designes slik at filtret 694 kun demper ved spektrale daler, og i enkelte tilfeller blir i det minste en del av størrelsesspektrumet klippet til å være flatt rundt formantregioner. Som omtalt nedenfor kan FIR etterfilter koeffisientene fremskaffes ved trunkering av en normalisert sekvens som er et resultat av den inverse Fourier-transformasjon av det prosesserte størrelsesspektrum.

Filtret 694 anvendes på den rekonstruerte tale i tidsdomenen. Filtret kan anvendes på hele båndet eller på et subbånd. I tillegg kan filtret brukes alene eller i forbindelse med andre filtre, så som langtids etterfiltre og/eller midtfrekvens forbedringsfilteret som er omtalt i nærmere detalj nedenfor.

Det beskrevne etterfilter kan opereres i forbindelse med kodeker ved bruk av forskjellige bit rater, forskjellige samplingsrater og forskjellige kodingsalgoritmer. Det antas at etterfiltret 694 er i stand til å produsere en signifikant kvalitetsforbedring i forhold til bruken av stemmekodeker uten etterfiltret. Spesifikt antas det at etterfiltret 694 reduserer den sansbare kvantiseringsstøy i frekvensområder hvor signaleffekten er relativt lav, det vil si i spektrale daler mellom formanter. I disse områder er signal-til-støy-forholdet typisk dårlig. Med andre ord, på grunn av det svake signal, er den støyen som er til stede relativt sett sterkere. Det antas at etterfiltret forbedrer den samlede talekvalitet ved å dempe støynivået i disse områder.

De rekonstruerte LPC-koeffisienter 638 inneholder ofte formantinformasjon, fordi frekvensresponsen til LPC-syntesefiltret typisk følger den spektrale innhylling av inngangstalen. LPC-koeffisienter 638 brukes følgelig til å avlede koeffisientene for korttids etterfiltret. Fordi LPC-koeffisientene 638 forandres fra en ramme til den neste eller på en annen basis, tilpasses etterfilter koeffisientene som avledes fra dem også fra ramme til ramme eller på en annen basis.

En teknikk for beregning av filterkoeffisientene for etterfiltret 694 er illustrert på figur 7. Dekoderen 600 på figur 6 gjennomfører teknikken. Alternativt gjennomfører en annen dekode eller etterfiltreringsverktøy teknikken.

Dekoderen 600 fremskaffer et LPC-spektrum ved null-utfylling av et sett av LPC-koeffisienter $a(i)$, hvor $i = 0, 1, 2, \dots, P$, og hvor $a(0) = 1$. Settet av LPC-koeffisienter 710 kan fremskaffes fra en bit strøm hvis en lineær prediksjonskodek så som en CELP-kodek, brukes. Alternativt kan settet av LPC-koeffisienter 710

fremskaffes ved analysering av et rekonstruert talesignal. Dette kan gjøres selv om kodeken ikke er en lineær prediksjonskodek. P er LPC-rekkefølgen for LPC-koeffisientene $a(i)$ som skal brukes ved bestemmelse av etterfilter koeffisientene. Null-utfylling involverer generelt forlenging av et signal (eller spektrum) med nuller for å forlenge dets tidsgrenser (eller frekvensbåndgrenser). I prosessen avbilder null-utfylling et signal med lengde P på signal med lengde N , hvor $N > P$. I en fullbånd kodek implementering er P ti for en åtte kHz samplingsrate, og seksten for samplingsrater som er høyere enn åtte kHz. Alternativt er P en annen verdi. For subbånd kodeker kan P være en forskjellig verdi for hvert subbånd. For eksempel, for en seksten kHz samplingsrate som bruker tre subbånd strukturen som er illustrert på figur 3, kan P være ti for det lave frekvensbånd 310, seks for det midtre bånd 320 og fire for det høye bånd 330. I en implementering er N 128. Alternativt er N et annet tall, så som 256.

Dekoderen 600 gjennomfører deretter en N -punkts transformasjon, så som en FFT 720, på de null-utfylte koeffisienter, hvilket gir et størrelsesspektrum $A(k)$. $A(k)$ er spektrumet for det null-utfylte LPC inverse filter, for $k = 0, 1, 2, \dots, N-1$. Det inverse av størrelsesspektrumet (næmlig $1/|A(k)|$) gir størrelsesspektrumet for LPC-syntesefiltret.

Størrelsesspektrumet for LPC-syntesefiltret blir valgfritt konvertert til den logaritmiske domene 725 for å redusere dets størrelsesområde. I en implementering er denne konverteringen som følger

$$H(k) = \ln \frac{1}{|A(k)|}$$

hvor \ln er den naturlige logaritme. Andre operasjoner kan imidlertid brukes for å redusere verdiområdet. For eksempel kan en operasjon med logaritme med grunntall ti brukes istedenfor en operasjon med naturlig logaritme.

Tre valgfrie ikke-lineære operasjoner er basert på verdiene av $H(k)$; normalisering 730, ikke-lineær kompresjon 735 og klipping 740.

Normalisering 730 er tilbøyelig til å gjøre verdiområdet av $H(k)$ mer konsistent fra ramme til ramme og bånd til bånd. Både normalisering 730 og ikke-lineær kompresjon 735 reduserer verdiområdet for det ikke-lineære størrelsesspektrum, slik at talesignalet ikke endres for mye av etterfiltret. Alternativt kan ytterligere og/eller andre teknikker brukes til å redusere verdiområdet for størrelsesspektrumet.

I en implementering blir initial normalisering 730 gjennomført for hvert bånd i en multibåndkodek som følger:

$$\hat{H}(k) = H(k) - H_{\min} + 0,1$$

5

hvor H_{\min} er minimumsverdien av $H(k)$, for $k=0, 1, 2, \dots, N-1$.

Normalisering 730 kan gjennomføres for en fullbåndkodek som følger:

$$\hat{H}(k) = \frac{H(k) - H_{\min}}{H_{\max} - H_{\min}} + 0,1$$

10

hvor H_{\min} er minimumsverdien av $H(k)$, og H_{\max} er maksimumsverdien av $H(k)$, for $k=0, 1, 2, \dots, N-1$. I begge normaliseringsligninger ovenfor adderes en konstant verdi på 0,1 for å hindre at maksimums- og minimumsverdiene av $\hat{H}(k)$ blir 1 henholdsvis 0, hvilket gjør ikke-lineær kompresjon mer virksom. Andre konstante verdier, eller andre teknikker, kan alternativt brukes for å forhindre null-verdier.

15

Ikke-lineær kompresjon 735 gjennomføres for ytterligere å justere det dynamiske område av det ikke-lineære spektrum som følger:

$$H_c(k) = \beta * |\hat{H}(k)|^\gamma$$

20

hvor $k = 0, 1, \dots, N-1$. Følgelig, hvis en 128-punkts FFT ble brukt til å konvertere koeffisientene til frekvensdomenen, så er $k = 0, 1, \dots, 127$. I tillegg, $\alpha = \eta * (H_{\max} - H_{\min})$, med η og γ tatt som passende valgte konstante faktorer. Verdiene av η og γ kan velges i henhold til typen av talekodek og kodingsraten. I en implementering velges η - og γ -parameterne eksperimentelt. For eksempel velges γ som en verdi fra området av 0,125 til 0,135, og η velges fra området fra 0,5 til 1,0. De konstante verdier kan justeres basert på preferanser. For eksempel fremskaffes et område av konstante verdier ved analysering av den predikerte spektrumforvrengning (overveiende rundt topper og daler) som er et resultat av forskjellige konstante verdier. Det er typisk ønskelig å velge et område som ikke overstiger et forhåndsbestemt nivå av predikert forvrengning. Sluttverdiene blir da valgt fra et sett av verdier innenfor området ved bruk av resultatene fra subjektive lyttetester. For eksempel, i et etterfilter med en åtte

25

30

kHz samplingsrate, er η 0,5 og γ er 0,125, og i et etterfilter med en seksten kHz samplingsrate, er η 1,0 og γ er 0,135.

Klipping 740 kan anvendes på det komprimerte spektrum, $H_c(k)$, som følger:

$$H_{pr}(k) = \begin{cases} \lambda * H_{\text{middel}} & H_c(k) > \lambda * H_{\text{middel}} \\ H_c(k) & \text{ellers} \end{cases}$$

5

ellers

10 hvor H_{middel} er middelveien av $H_c(k)$ og λ er en konstant. Verdien av λ kan velges forskjellig i henhold til typen av talekodek og kodingsraten. I enkelte implementeringer velges λ eksperimentelt (så som en verdi fra 0,95 til 1,1), og den kan justeres basert på preferanser. For eksempel kan sluttverdiene av λ velges ved bruk av resultatene fra subjektive lyttetester. For eksempel, i et etterfilter med en åtte kHz samplingsrate, er λ 1,1, og i etterfilter som opererer ved en seksten kHz samplingsrate, er λ 0,95.

15 Klippingsoperasjonen setter lokk på verdiene av $H_{pr}(k)$ ved et maksimum eller tak. I de ovenstående ligninger representeres dette maksimum som $\lambda * H_{\text{middel}}$. Alternativt brukes andre operasjoner til å sette lokk på verdiene av størrelsesspektrumet. For eksempel kan taket være basert på medianverdien av $H_c(k)$, isteden for middelveien. Videre, istedenfor å klippe alle de høye $H_c(k)$ -verdier til en spesifikk maksimumsverdi (så som $\lambda * H_{\text{middel}}$), kan verdiene klippes i henhold til en mer kompleks operasjon.

20 Klipping er tilbøyelig til å resultere i filterkoeffisienter som vil dempe tale-signalet ved dets daler uten i vesentlig grad å forandre talespektrumet i andre regioner, så som formantregioner. Dette kan hindre etterfiltret i å forvreng taleformantene, hvilket gir taleutgang med høyere kvalitet. I tillegg kan klipping redusere effektene av spektral helling, fordi klipping flater ut etterfilterspektrumet ved 25 reduisering av de store verdier til den verdi det er satt lokk på, mens verdiene rundt dalene forblir hovedsakelig uforandret.

Når konvertering til den logaritmiske domene ble gjennomført, blir det resulterende klippede størrelsesspektrum, $H_{pr}(k)$, konvertert 745 fra logg-domenen til 30 den lineære domene, for eksempel som følger:

$$H_{pr}(k) = \exp(H_{pr}(k))$$

hvor \exp er den inverse naturlige logaritmefunksjon.

En N -punkts invers hurtig Fourier-transformasjon 750 utføres på $H_{prf}(k)$, hvilket gir en tidssekvens på $f(n)$, hvor $n=0,1,\dots,N-1$, og N er den samme som i FFT-operasjonen 720 omtalt ovenfor. $f(n)$ er således en N -punkts tidssekvens.

På figur 7 er verdiene av $f(n)$ trunkert 755 ved at verdiene settes til null for $n>M-1$, som følger:

$$h(n) = \begin{cases} f(n) & n = 0,1,2,\dots,M-1 \\ 0 & n > M-1 \end{cases}$$

10 hvor M er rekkefølgen av korttids etterfiltret. Generelt gir en høyere verdi av M filtrert tale med høyere kvalitet. Kompleksiteten ved etterfiltret øker imidlertid ettersom M øker. Verdien av M kan velges, idet disse avveiningene tas i betraktning. I en implementering er M sytten.

15 Verdiene av $h(n)$ blir valgfritt normalisert 760 for å unngå brå forandringer mellom rammer. Dette gjøres for eksempel som følger:

$$h_{prf}(n) = \begin{cases} 1 & n = 0 \\ h(n)/h(0) & n = 1,2,3,\dots,M-1 \end{cases}$$

20 Alternativt brukes en annen normaliseringsoperasjon. For eksempel kan den følgende operasjon brukes:

$$h_n(n) = \frac{h(n)}{\sqrt{\sum_{n=0}^{M-1} h^2(n)}}$$

25 I en implementering hvor normalisering gir etterfilter koeffisienter $h_{prf}(n)$ 765, anvendes et FIR-filter med koeffisienter $h_{prf}(n)$ 765 på den syntetiserte tale i tidsdomenen. Således, i denne implementering, settes den første ordens etterfilter koeffisient ($n=0$) til en verdi på én for hver ramme, for å forhindre signifikante avvik i filterkoeffisientene fra en ramme til den neste.

30 **B. Eksempel midtfrekvens forbedringsfiltre**

I enkelte utførelser inkorporerer en dekode, så som dekoderen 270 som er vist på figur 2, et midtfrekvens forbedringsfilter for etterprosessering, eller et slikt filter anvendes på utgangen fra dekoderen 270. Alternativt inkorporeres et slikt filter i eller anvendes på utgangen fra en annen type av audiodekoder eller

5 prosesseringsverktøy, for eksempel en talekodek som er beskrevet et annet sted i den foreliggende søknad.

Som omtalt ovenfor dekomponerer multibåndkodeker et inngangssignal til kanaler med reduserte båndbredder, typisk fordi subbånd er mer håndterlige og fleksible for koding. Båndpassfiltre, så som filterbankene 216 som er beskrevet

10 ovenfor med henvisning til figur 2, blir ofte brukt for signaldekomposisjon før koding. Signaldekomposisjon kan imidlertid forårsake et tap av signalenergi ved frekvensområdene mellom passbåndene for passbåndfiltrene. Midtfrekvens forbedrings (middle frequency enhancement, "MFE") filteret hjelper med dette

15 potensielle problem til ved å forsterke størrelsesspektrumet for den dekodete utgangstale ved frekvensområder hvis energi ble dempet på grunn av signaldekomposisjon, uten i vesentlig å endre energien ved andre frekvensområder.

På figur 2 anvendes et MFE-filter 284 på utgangen fra båndsyntesefilteret (-filtrene), så som utgangen 292 fra filterbankene 280. Hvis bånd n dekodere 272, 274 er vist på figur 6, anvendes følgelig korttids etterfiltret 694 separat på hvert

20 rekonstruerte bånd i en subbånddekode, mens MFE-filtret 284 anvendes på det kombinerte eller kompositt-rekonstruerte signal som inkluderer bidrag fra de flere subbånd. Som påpekt anvendes alternativt et MFE-filter i forbindelse med en dekode som har en annen konfigurasjon.

I enkelte implementeringer er MFE-filtret et annen ordens båndpass FIR-filte.

25 Det kaskaderer et første ordens lavpassfilter og et første ordens høypassfilter. Begge første ordens filtre kan ha identiske koeffisienter. Koeffisientene er typisk valgt slik at MFE-filterforsterkningen er ønskelig ved gjennomgangsbånd (økende energi for signalet) og enhet ved stopp-bånd (gjennomføring av signalet uforandret eller relativt uforandret). Alternativt brukes en annen teknikk til å forbedre frekvensområder som

30 har blitt dempet på grunn av bånd-dekomposisjon.

Transferfunksjonen for et første ordens lavpass filter er:

$$H_1 = \frac{1}{1 - \mu} + \frac{\mu}{1 - \mu} Z^{-1}$$

Transferfunksjonen for et første ordens høypassfilter er:

$$H_2 = \frac{1}{1+\mu} - \frac{\mu}{1+\mu} Z^{-1}$$

5 Transferfunksjonen for et annen ordens MFE-filter som kaskaderer lavpass-filtret og høypassfiltret ovenfor er således:

$$H = H_1 \cdot H_2 = \left(\frac{1}{1-\mu} + \frac{\mu}{1-\mu} Z^{-1} \right) \cdot \left(\frac{1}{1+\mu} - \frac{\mu}{1+\mu} Z^{-1} \right) = \frac{1}{1-\mu^2} - \frac{\mu^2}{1-\mu^2} Z^{-2}$$

10 De korresponderende MFE-filterkoeffisienter kan representeres som:

$$h(n) = \begin{cases} \frac{1}{1-\mu^2} & n=0 \\ -\frac{\mu^2}{1-\mu^2} & n=2 \\ 0 & \text{ellers} \end{cases}$$

Verdien av μ kan velges ved eksperiment. For eksempel fremskaffes et
 15 verdiområde av konstante verdier ved analysing av den predikerte spektrumfor-
 vringning som er et resultat av forskjellige konstante verdier. Det er typisk ønskelig å
 velge et verdiområde som ikke overstiger et forhåndsbestemt nivå av predikert
 forvringning. Sluttverdiene blir da valgt blant et sett av verdier innenfor verdiområdet
 ved bruk av resultatene fra subjektive lyttetester. I en implementering, når det brukes
 20 en seksten kHz samplingsrate, og talen er brutt opp i de følgende tre bånd (null til
 åtte kHz, åtte til tolv kHz og tolv til seksten kHz), kan det være ønskelig å forbedre
 området rundt åtte kHz, og μ velges til å være 0,45. Alternativt velges andre verdier
 av μ , særlig hvis det er ønskelig å forbedre et annet frekvensområde. Alternativt
 implementeres MFE-filtret med ett eller flere båndpassfiltre med forskjellig design,
 25 eller MFE-filtret implementeres med ett eller flere andre filtre.

Etter at prinsippene for vår oppfinnelse nå er beskrevet og illustrert med
 henvisning til beskrevne utførelser, vil det medgis at de beskrevne utførelser kan

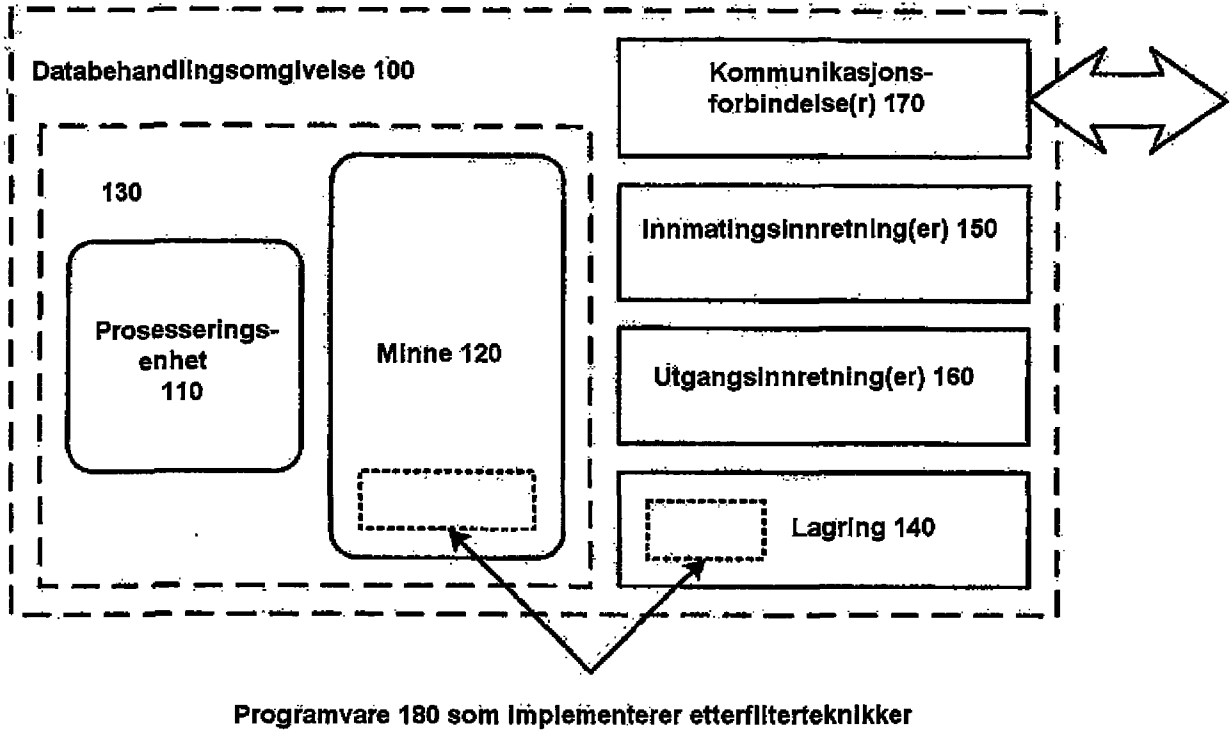
modifiseres i arrangement og detalj uten å avvike fra slike prinsipper. Det skal forstås at de programmer, prosesser eller fremgangsmåter som her er beskrevet ikke er relatert eller begrenset til noen bestemt type av databehandlingsomgivelse, med mindre annet er angitt. Forskjellige typer av databehandlingsomgivelser for generelt eller spesialisert formål kan brukes sammen med eller utføre operasjoner i samsvar med den lære som her er beskrevet. Elementer av de beskrevne utførelser som er vist i programvare kan implementeres i maskinvare, og omvendt.

I lys av de mange mulige utførelser som prinsippene ved vår oppfinnelse kan anvendes på, krever vi som vår oppfinnelse alle slike utførelser som kan komme innenfor omfanget og ideen til de følgende krav og ekvivalenter til dette.

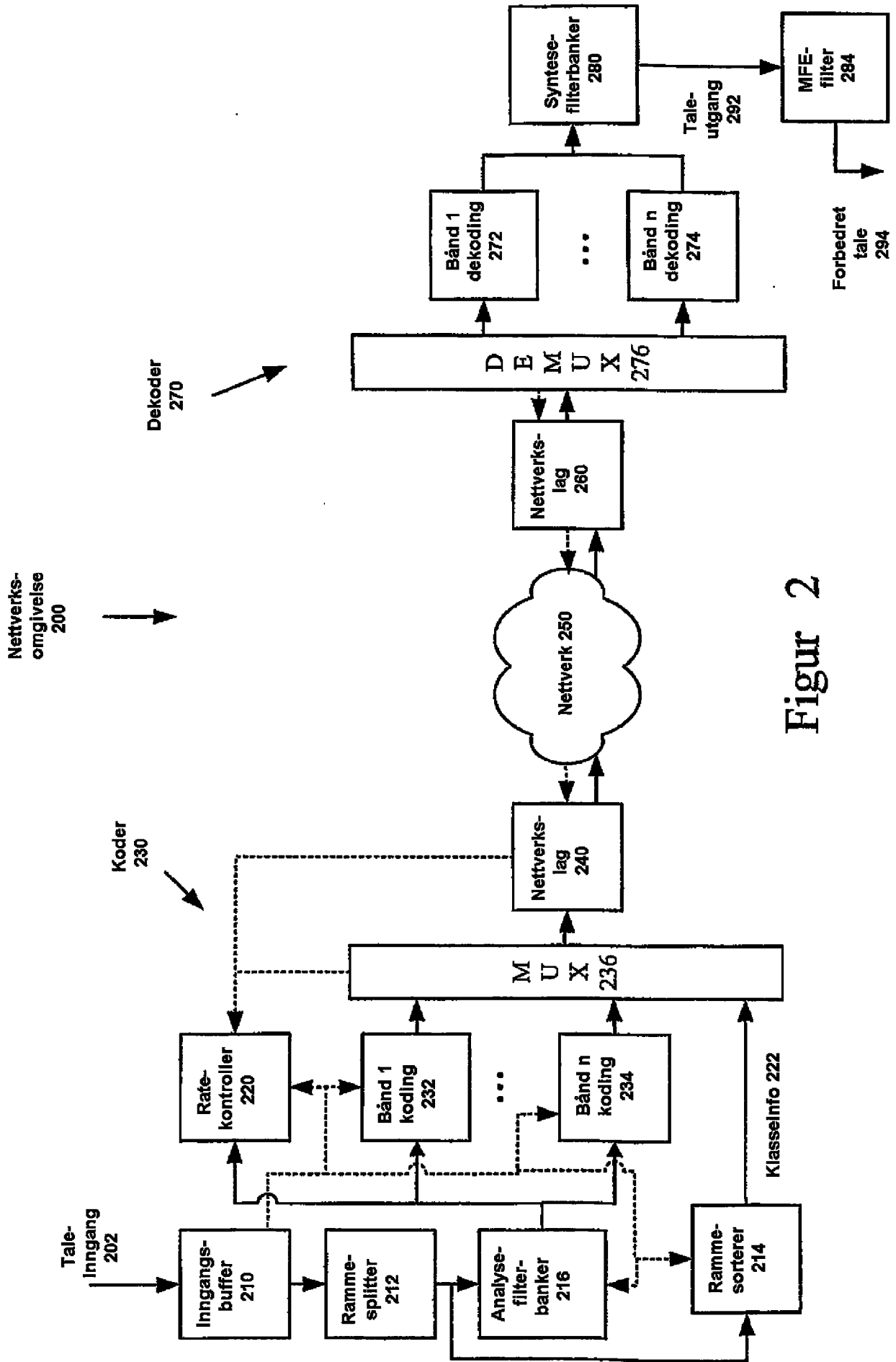
P A T E N T K R A V

1. Datamaskinimplementert fremgangsmåte omfattende:
 - beregning av et sett av filterkoeffisienter for anvendelse på et rekonstruert
5 audiosignal, hvor beregningen av settet av filterkoeffisienter omfatter:
 - gjennomføring (720) av en transformasjon av et sett av initiale tidsdomene-
verdier fra en tidsdomene inn i en frekvensdomene, hvilket produserer et sett av
initiale frekvensdomeneverdier;
 - gjennomføring av én eller flere frekvensdomeneberegninger ved bruk av de
10 initiale frekvensdomeneverdiene for å produsere et sett av prosesserte frekvens-
domeneverdier; og
 - gjennomføring (750) av en transformasjon av de prosesserte frekvensdomene-
verdier fra frekvensdomenen inn i tidsdomenen, hvilket produserer et sett av
prosesserte tidsdomeneverdier;
 - 15 - produsering av et filtrert audiosignal ved filtrering av i det minste en del av det
rekonstruerte audiosignal i en tidsdomene ved bruk av settet av filterkoeffisienter
(694), hvor:
 - gjennomføring (740) av én ene eller flere frekvensdomeneberegninger ved
bruk av de initiale frekvensdomeneverdiene for å produsere et sett av prosesserte
20 frekvensdomeneverdier, omfatter klipping av frekvensdomeneverdiene i frekvens-
domenen, slik at kun de frekvensdomeneverdier som overstiger en maksimumsklippe-
verdi, blir klippet.
2. Fremgangsmåte som angitt i krav 1, hvor det filtrerte audiosignal representerer
25 et frekvens subbånd av det rekonstruerte audiosignal.
3. Fremgangsmåte som angitt i krav 1, hvor beregning av settet av filter-
koeffisienter omfatter:
 - trunkering av settet av tidsdomeneverdier i tidsdomenen.
- 30
4. Fremgangsmåte som angitt i krav 1, hvor beregning av settet av filter-
koeffisienter omfatter prosessering av et sett av lineære prediksjonskoeffisienter.

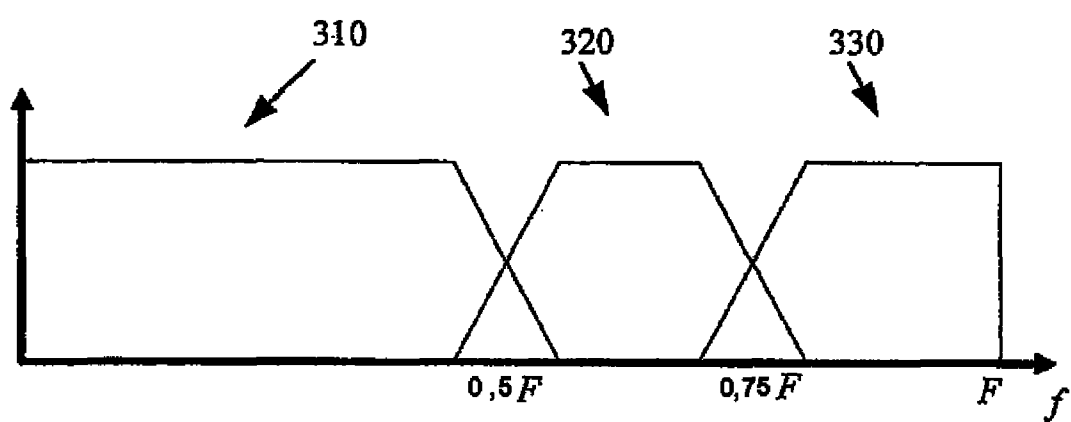
5. Fremgangsmåte som angitt i krav 4, hvor prosessering av settet av lineære prediksjonskoeffisienter omfatter: å sette lokk på et spektrum som er avledet fra settet av lineære prediksjonskoeffisienter.
- 5 6. Fremgangsmåte som angitt i krav 4, hvor prosessering av settet av lineære prediksjonskoeffisienter omfatter redusering av et verdiområde for et spektrum som er avledet fra settet av lineære prediksjonskoeffisienter.
- 10 7. Fremgangsmåte som angitt i krav 1, hvor den ene eller de flere frekvensdomeneberegninger omfatter én eller flere beregninger i en logaritmisk domene.



Figur 1

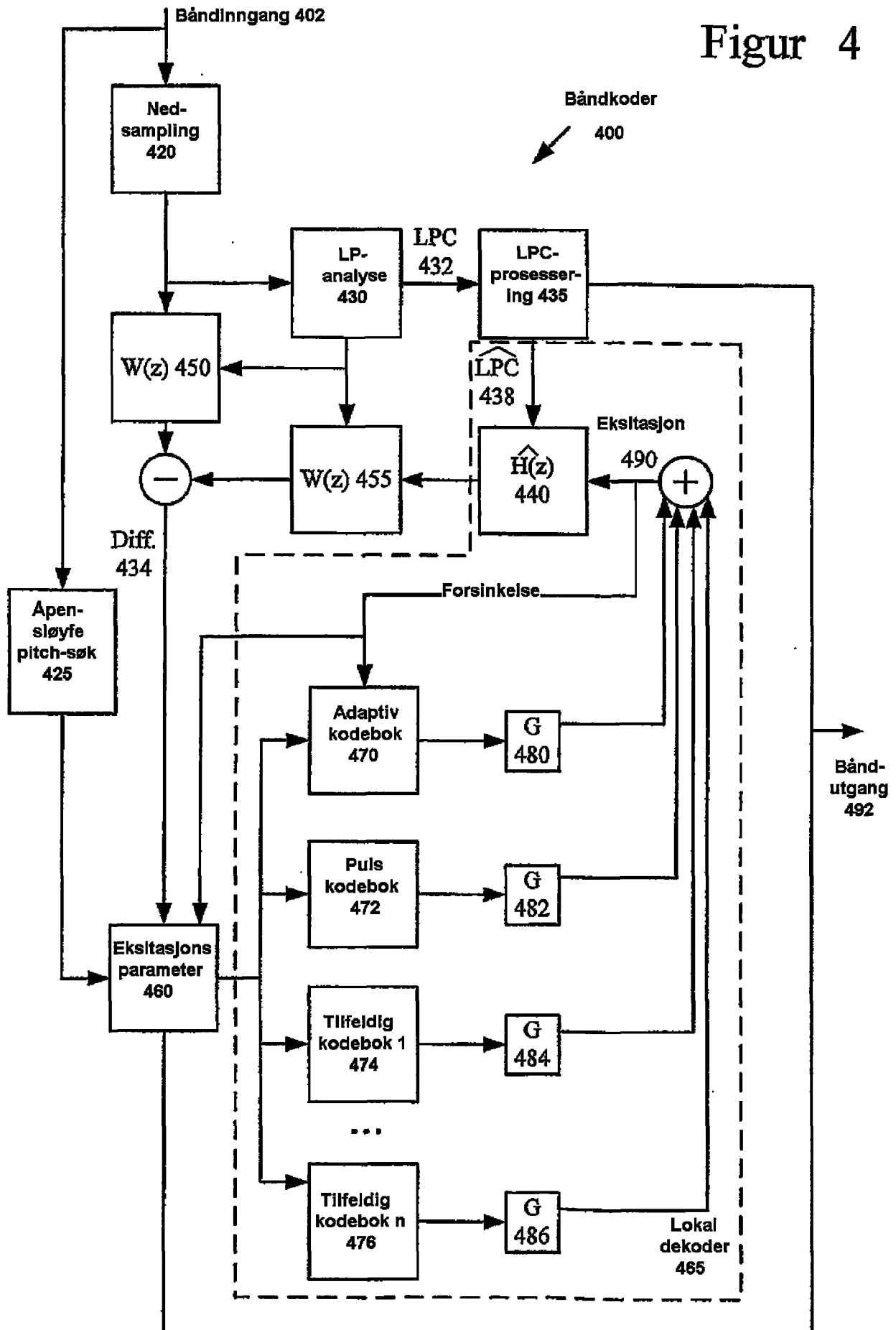


Figur 2



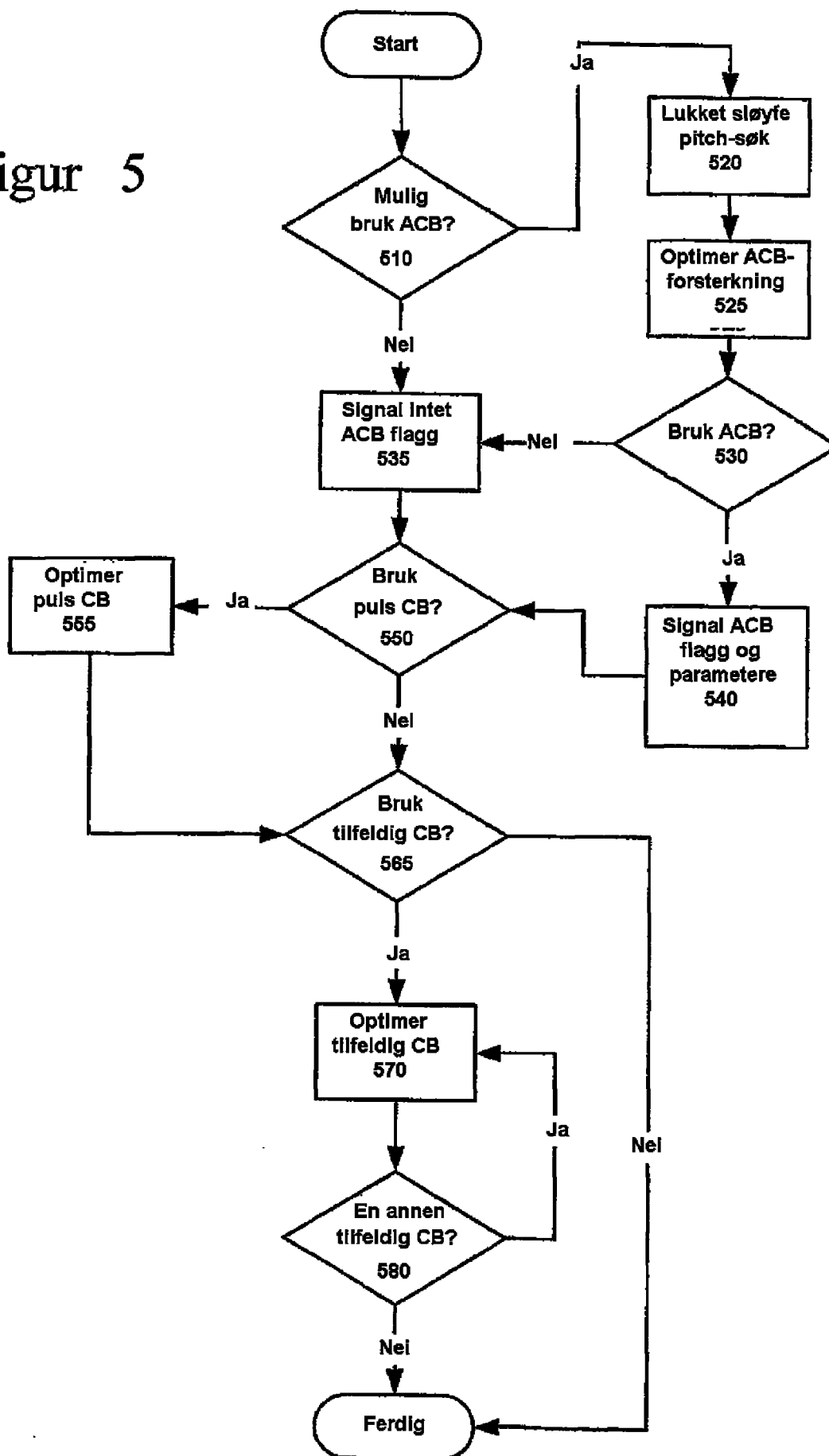
Figur 3

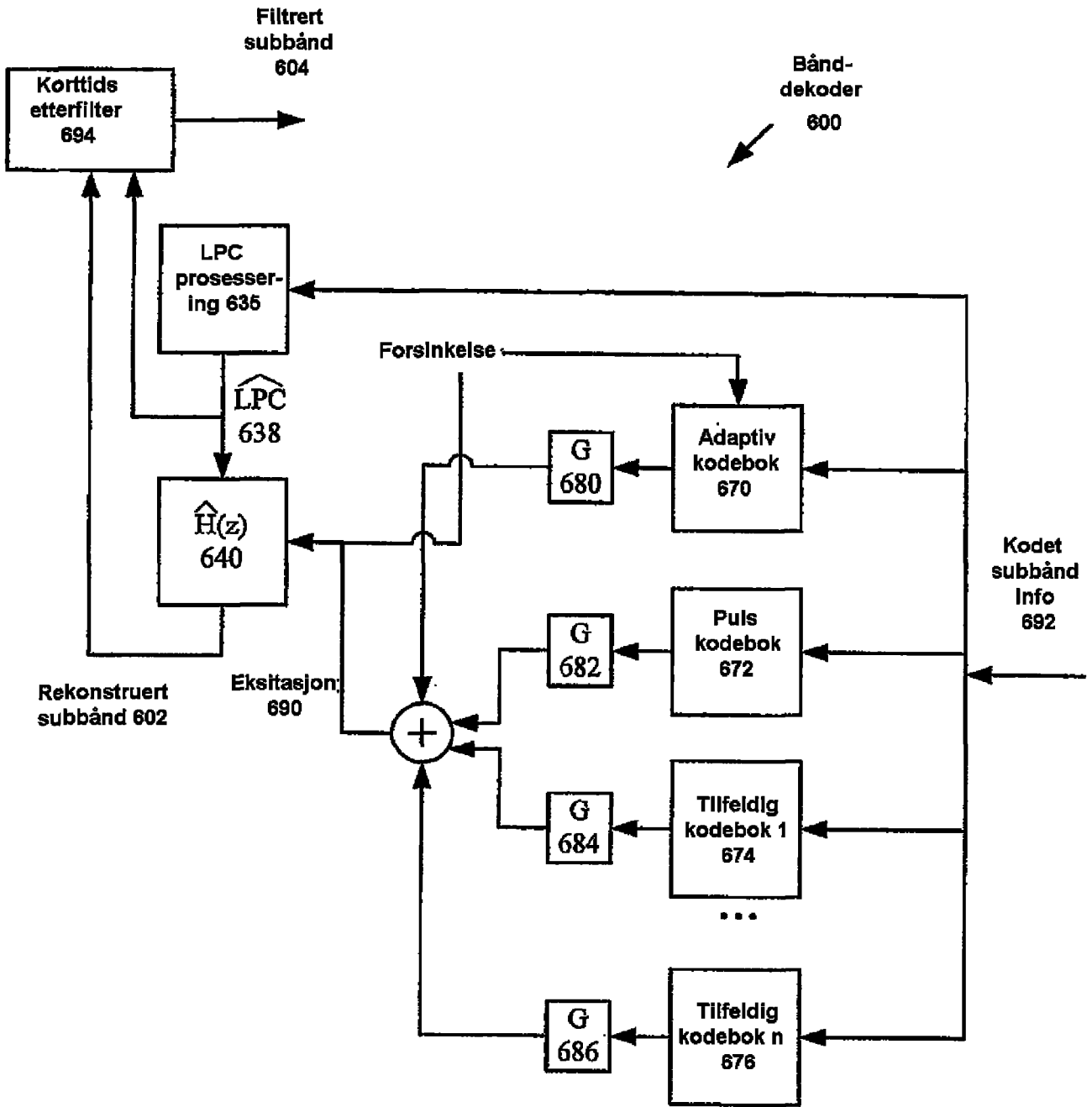
Figur 4



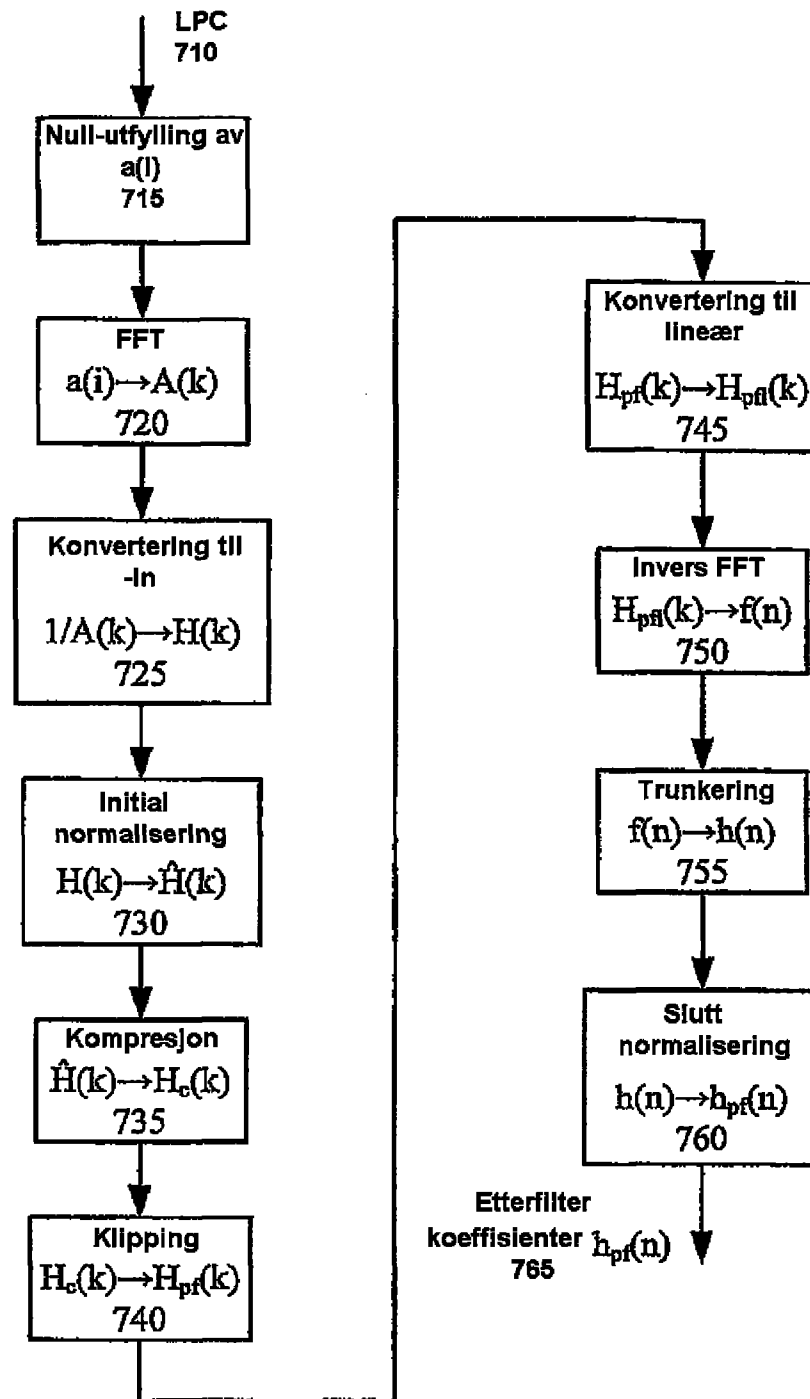
5/7

Figur 5





Figur 6



Figur 7