



Institutionen för
Elektro- och Informationsteknik
Lunds Universitet – Lunds Tekniska Högskola

Optimal Signalbehandling

Labbhandledning

Martin Stridh
Leif Sörnmo
Bengt Mandersson
2012

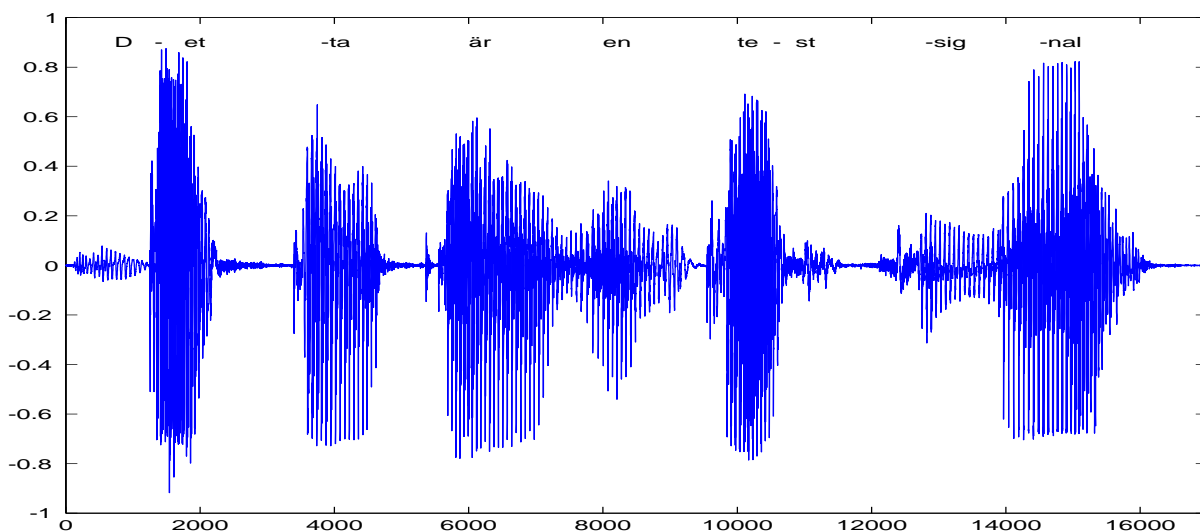
Department of Electrical and Information Technology, Lund University, Sweden

1 Talkodning

1.1 Modell av talsignal

Vid modellering av en talsignal skiljer man huvudsakligen mellan tonande och icke-tonande ljud. De tonande ljuden genereras genom att en luftström pressas genom struphuvudet vilket gör att de då spända stämbanderna börjar vibrera varvid en pulserade luftström bildas. Denna luftström formas sedan spektralt i svalg, munhåla och näshålor vilka tillsammans bildar talapparaten. Ven tänder och läppar deltar vid genereringen av ljudet. Olika talljud erhålls genom att variera talapparatens form. Vid generering av icke-tonande ljud är stämbanderna slappa och luftströmmen blir därför icke-periodisk.

Talsignalen är en icke-stationär signal som varierar mellan periodiska intervall (tonande ljud) och icke-periodiska intervall (icke-tonande ljud). I Fig. 1 visas ett exempel på en talsignal. Betraktar man



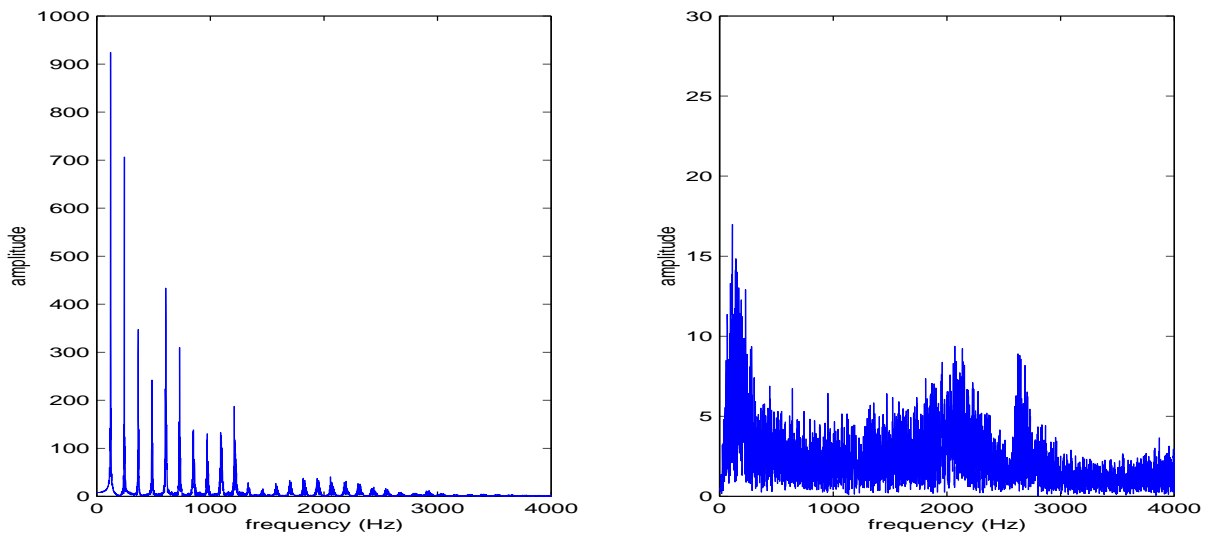
Figur 1: Exempel på talsignal $s(t)$.

frekvensspektrat för de båda typerna av ljud finner man att de tonande ljuden har ett linjespektrum dvs består av ett antal diskreta frekvenser (signalen periodisk) medan de icke-tonande ljuden har kontinuerligt spektrum, se Fig 2.

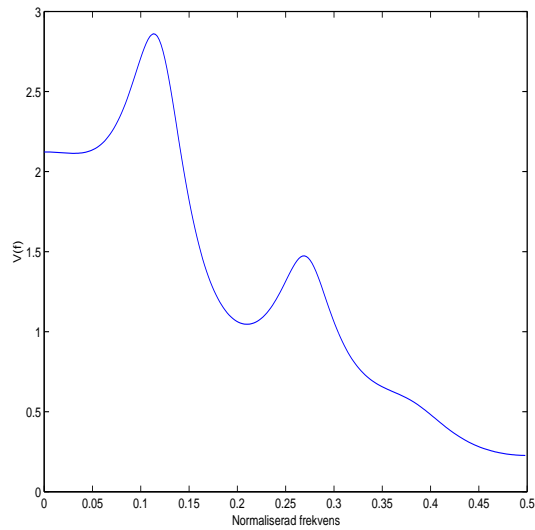
Antag att $E(f)$ är Fouriertransformen av den signal som genereras av stämbanderna. För tonade ljud är detta en periodisk signal vars grundton, F_0 , kallas pitchfrekvens och normalt ligger i intervallet 100 Hz för mansröst till 400 Hz för ljus kvinnoröst. $E(f)$ har lågpasskaraktär och förstärkningen (amplitudfunktionen) avtar med ca 12 dB/oktav (oktav=dubbling av frekvens). För icke-tonande ljud kan denna signal betraktas som brus vars effektspektrum avtar med 6 db/oktav.

Låt vidare $V(f)$ representera talapparatens överföringsfunktion. $V(f)$ är en kontinuerlig funktion med ett antal resonansfrekvenser F_1, F_2, \dots osv som är kopplade till de olika hålrum som bygger upp talapparaten (svalg, näs- och munhålor). Dessa resonansfrekvenser kallas formanter. Genom att variera talapparatens form ändras formanterna och därmed också det tonande ljudets karaktär. Ett exempel på en sådan funktion visas i Fig. 3.

I litteraturen finns åtskilliga modeller av talapparatens överföringsfunktion beskrivna. Vid härledning av en sådan modell betraktas talapparaten ofta som ett antal (p st) sammansatta cylindriska tuber med lika längd l men med olika tvärsnittsytor S_i där $i = 1, \dots, p$. Cylindrarnas väggar anses vara hårda och ljudet utbreder sig som en plan våg. Vid varje övergång erhålls då en framåtgående våg med hastighet $f_i(n)$ och en bakåtgående våg med hastighet $b_i(n)$, se Fig. 4. Motsvarande tryck ges av $R_i f_i(n)$ och $R_i b_i(n)$ där R_i är den i :te tubens akustiska impedans vilken är omvänt proportionell mot tubens



Figur 2: Exempel på spektrum för ljuden till vänster) a (tonande) och till höger) s (icke-tonande).



Figur 3: Exempel på överföringsfunktion för talapparaten.

tvärsnitt S_i . Fortplantningen kan anses vara förlustfri och dispersionsfri. Låt utbredningshastigheten vara v . Enligt Fig. 4 gäller då

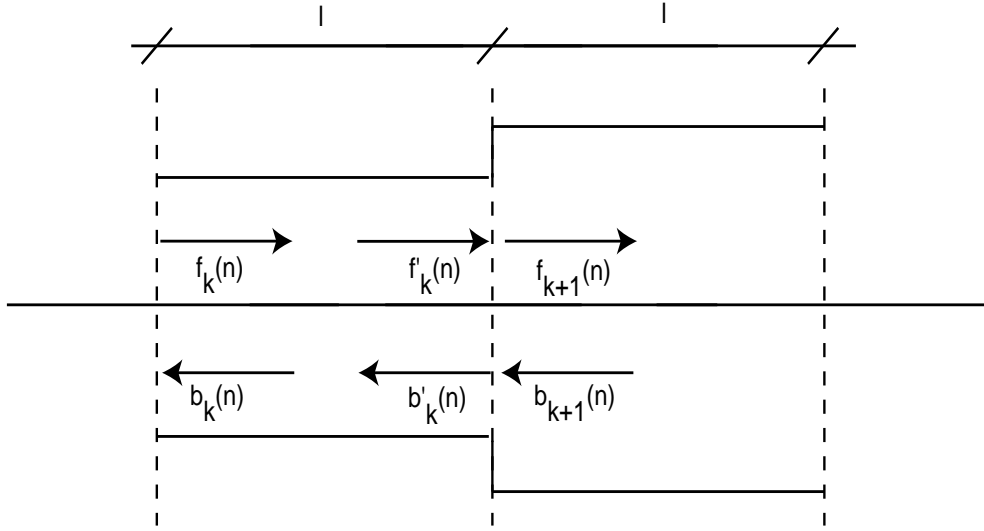
$$f'_i(n) = f_i(n - \frac{l}{v}) \quad (1)$$

$$b'_i(n) = b_i(n + \frac{l}{v}) \quad (2)$$

Med lämpligt val av samplingsintervall fås

$$f'_k(n) = f_k(n - 1) \quad (3)$$

$$b'_k(n) = b_k(n + 1) \quad (4)$$



Figur 4: Den i:te övergången i en tubmodell av talapparaten.

Till vänster om gränssytan gäller för trycket p_i och hastigheten u_i att

$$p_i = R_i(f'_i(n) + b'_i(n)) \quad (5)$$

$$u_i = f'_i(n) - b'_i(n) \quad (6)$$

På samma sätt gäller till höger om gränssytan att

$$p_{i+1} = R_{i+1}(f_{i+1}(n) + b_{i+1}(n)) \quad (7)$$

$$u_{i+1} = f_{i+1}(n) - b_{i+1}(n) \quad (8)$$

Nu gäller att både tryck och hastighet är samma till höger och vänster om en diskontinuitet. Då finner man att

$$R_i f'_i(n) + R_i b'_i(n) = R_{i+1} f_{i+1}(n) + R_{i+1} b_{i+1}(n) \quad (9)$$

$$f'_i(n) - b'_i(n) = f_{i+1}(n) - b_{i+1}(n) \quad (10)$$

vilket kan skrivas om som

$$f_{i+1}(n) = \left(1 + \frac{R_i - R_{i+1}}{R_i + R_{i+1}}\right) f'_i(n) + \frac{R_i - R_{i+1}}{R_i + R_{i+1}} b_{i+1}(n) \quad (11)$$

$$b'_i(n) = -\frac{R_i - R_{i+1}}{R_i + R_{i+1}} f'_i(n) + \left(1 + \frac{R_i - R_{i+1}}{R_i + R_{i+1}}\right) b_{i+1}(n) \quad (12)$$

Inför nu de från optiken bekanta reflexionskoefficienterna

$$\gamma_i = \frac{R_i - R_{i+1}}{R_i + R_{i+1}} \quad (13)$$

och definitionerna av $f'_i(n)$ och $b'_i(n)$ fås

$$f_{i+1}(n) = (1 + \gamma_i) f_i(n - 1) + \gamma_i b_{i+1}(n) \quad (14)$$

$$b_i(n - 1) = -\gamma_i f_i(n - 1) + (1 + \gamma_i) b_{i+1}(n) \quad (15)$$

vilket motsvarar en lattice länk. Modellen för talapparaten kan således beskrivas som en seriekoppling av p stycken latticelänkar. verföringsfunktionen för en sådan modell kan därför skrivas (Jämför Hayes).

$$V(z) = \frac{\text{konst}}{A(z)} = \frac{\text{konst}}{1 + \sum_{i=1}^p a_i z^{-i}} \quad (16)$$

Ljudet påverkas också av läpparna. Låt funktionen $R(f)$ beskriva läpparnas överföringsfunktion. En ofta använd modell är

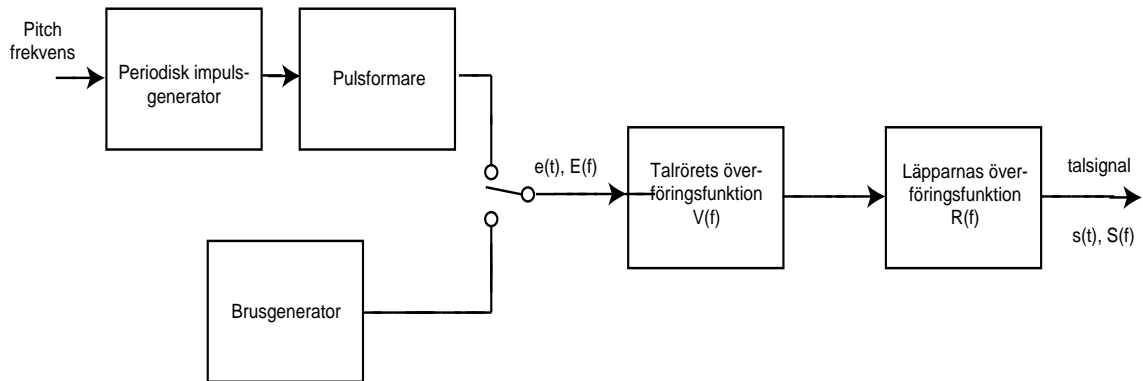
$$R(z) = 1 - z^{-1} \quad (17)$$

vilket är ett högpassfilter vars förstärkning växer med ca 6 dB/oktav

Talsignalens Fouriertransform $S(f)$ kan nu skrivas

$$S(f) = R(f)V(f)E(f) \quad (18)$$

vilket illustreras i Fig. 5. Den övergripande trenden hos $S(f)$ bestäms huvudsakligen av $E(f)R(f)$.



Figur 5: Principskiss av en talmodell.

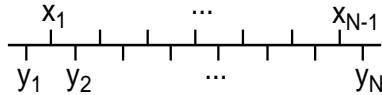
För tonande ljud avtar denna funktion med ca 6 dB/oktav. Förstärks högre frekvenser med ett s.k. pre-emphasis nät erhålles en periodisk signal (linjespektrum i $E(f)$) med någorlunda rakt spektrum vars form väsentligen bestäms av talapparatus överföringsfunktion $V(f)$. De icke-tonande ljuden kännetecknas av ett någorlunda vitt spektrum.

1.2 Kvantisering

En N -punkters kvantiserare avbildar reella tal på en ändlig reell mängd $M = (y_1, y_2, \dots, y_N) \subset \mathcal{R}$ där M definierar kvantiserarens möjliga utsignaler. Värdena y_i kallas kvantiseringsnivåer och mellan dessa finns beslutsnivåer x_i .

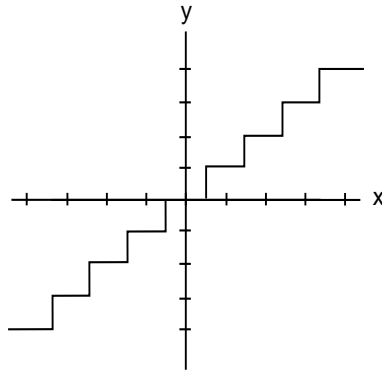
$$x_0 < y_1 < x_1 < y_2 < x_2 < \dots < y_N < x_N \quad (19)$$

Dessa nivåer illustreras i Fig 6 och insignal/utsignalsambandet visas i Fig 7. Oftast är x_0 och x_N



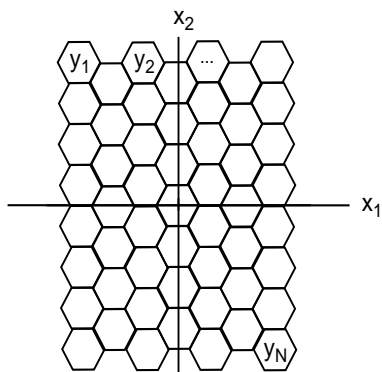
Figur 6: Skalär kvantisering: Kvantiseringsnivåer y_i och beslutsnivåer x_i .

lika med $-\infty$ respektive ∞ . Ovanstående kallas en skalär kvantiserare eftersom ett värde kvantiseras



Figur 7: Skalär kvantisering: insignal(x)/utsignal(y)-samband.

åt gången. Man kan också kvantisera flera värden gruppvis med s.k. vektorkvantisering vilket i det 2-dimensionella fallet illustreras i Fig 8. Där ges en viss kombination av värden i flera parametrar en gemensam kod.



Figur 8: 2-dim. vektorkvantisering: Inparametrar x_1, x_2 och kod y

1.2.1 Kvantiseringsdistorsion

Ett fel erhålles alltid vid kvantisering och felet är starkt beroende av kvantiseringsnivåerna y_i och beslutsnivåerna x_i . Felet leder till distorsion och ett mått på denna är det kvadratiske medelfelet som definieras av

$$D = \sum_{i=1}^N \int_{x_{i-1}}^{x_i} (y_i - x)^2 p_X(x) dx \quad (20)$$

där $p_X(x)$ är signalens fördelningsfunktion. Som kvalitetsmått används signal-distorsionsförhållandet

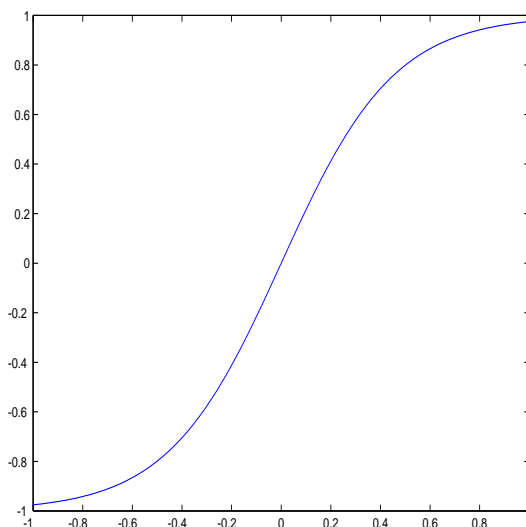
$$SDR = 10 \log_{10} \frac{\sigma^2}{D} \quad (21)$$

1.2.2 Likformig/olikformig kvantisering

Vid likformig kvantisering delas utstyringsområdet $(-V, V)$ upp i N lika stora kvantiseringsintervall dvs $\Delta_i = x_i - x_{i-1} = \Delta$. Storleken av varje sådant intervall blir $\Delta = 2V/N$. Införs bithastigheten $r = \log_2 N$ kan ovanstående skrivas om till $\Delta = 2V/2^r$. En likformig kvantiserare är därmed fullständigt bestämd av N . För att undvika överstyrning väljs V ofta som $V = k\sigma_x$ där ett vanligt val av k är 4. Sannolikheten för överstyrning är då mycket liten.

Ett annan metod är att använda olikformig kvantisering. Vid t.ex. talöverföring kan olika signaler ha vitt skilda nivåer och överföringskvaliteten måste vara acceptabel och helst konstant inom ett relativt brett intervall. Olikformig kvantisering är då att föredra av flera skäl. För det första är i talsammanhang små amplituder vanligare än stora varför man bör välja mindre steg för dessa. Vidare har olika talsignaler ofta högst varierande effektnivåer. Med kompression täcker man lättare ett stort dynamiskt område, vilket utnyttjar att örat är mer känsligt för relativa än absoluta amplitudvariationer och därmed motiverar mindre kvantiseringsteg för låga amplituder och större för höga.

Olikformig kvantisering kan åstadkommas antingen genom att kombinera den likformiga kvantiseraren med en s.k kompressorfunktion, se Fig 9. Kompressorn förstärker låga nivåer mer än höga. Sedan kan utsignalen från kompressorn kvantiseras likformigt. När informationen ska föras tillbaka till den ursprungliga används den inversa kompressorfunktion även kallad expander. Den andra möjligheten är



Figur 9: Exempel på en kompressorfunktion.

att göra själva kvantiseraren olikformig.

En i Europa standardiserad kompressorfunktion är den s.k A-lagen (i USA finns den liknande μ -lagen) i vilken kompressorfunktionen $g(x)$ ges av

$$g(x) = \frac{Ax}{1 + \log(A)} \quad 0 \leq x \leq V/A \quad (22)$$

$$g(x) = \frac{V + V \log(Ax/V)}{1 + \log(A)} \quad V/A \leq x \leq V \quad (23)$$

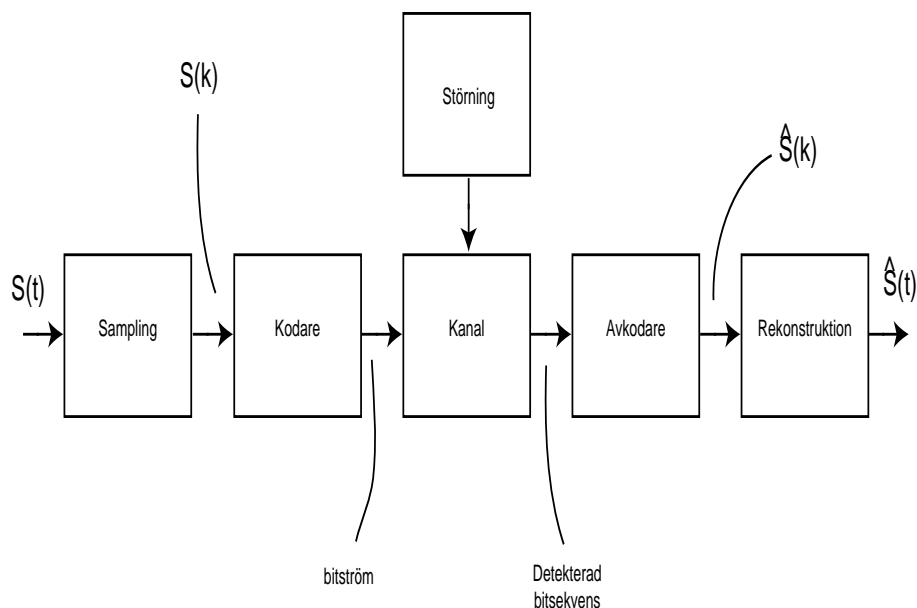
där $g(-x) = -g(x)$ och A ofta ligger mellan 100-150. För att slippa divisioner och logaritmer i hårdvarurealiseringar används ofta A-lagen linjärt approximerad i ett antal segment. Experimentellt har visats att det normalt krävs 12-13 bitar per sampel för talsignaler. Med kompressorteknik räcker 7-8 bitar per sampel.

1.2.3 Optimal kvantisering

Som nämndes ovan uppstår alltid distorsion vid kvantisering. En optimal kvantisering kan beräknas för varje rimlig sannolikhetsfördelning ($p_X(x)$) och beräknas genom derivering av uttrycket för distorsionen i ekv. (20). En algoritm för beräkning av en optimal kvantiserare utvecklades av Lloyd 1957 som är en iterativ algoritm där distorsionen successivt minskas. Optimala beslutsnivåer finns tabellerade för olika fördelningar.

1.3 Kodning av analoga signaler

Transmission av en analog informationssignal är ett klassiskt problem. Fig. 10 visar ett blockschema för ett system för digital överföring av analog information t.ex. tal eller video.



Figur 10: Principskiss över digital transmission av analog informationssignal.

Praktiskt kan $s(k)$ anses vara kvantiserad i en AD omvandlare med stort antal bitar. Kodarens (källkodarens) uppgift är då att transformera $s(k)$ till bitström med en given bithastighet. Avkodaren utför den inversa transformationen. Syftet är att den rekonstruerade signalen ska likna den ursprungliga så mycket som möjligt. Systemet bör vara robust mot bitfel som orskas av störningar på kanalen.

Mellan källkodaren och den tillhörande avkodaren kan man tänka sig olika typer av kanaler. Här kommer vi att tänka oss att kanalen är en sladd men den kan också vara en radiokanal som tex i

mobiltelefoni där det då tillkommer kanalkodning (tex interleaving), modulering, demodulering och kanalavkodning kring kanalen.

Källkodning av analoga signaler kan delas in i tre typer: tidskodning, frekvenskodning och modellbaserad kodning.

1.3.1 Tidskodning

I denna typ av kodning är det tidsvariationer i signalen som representeras digitalt. Den mest grundläggande metoden är *Pulse Code Modulation* (PCM) i vilken varje sampel i signalen kvantiseras till en av 2^p kvantiseringsnivåer där p är antalet bitar som behövs för att representera varje sampel. Både likformig och olikformig kvantisering kan användas.

I många signaler är det stor korrelation mellan konsekutiva sampel och följaktligen är då den genomsnittliga förändringen i varje nytt sampel liten. En metod som utnyttjar redundansen i signalen skulle därför kunna minska bithastigheten. Istället för att skicka över hela amplituden så kan skillnaden mellan på varandra följande sampel skickas över. Denna metod kallas *Differential Pulse Code Modulation* (DPCM). Eftersom skillnaden mellan konsekutiva sampel förväntas bli mindre än amplituden kan färre antal bitar användas till att representera informationen. En ytterligare förbättring av denna metod är att istället för att ta skillnaden mellan ett sampel och dess föregångare så kan skillnaden mellan ett sampelvärde och en prediktion av samma sampel baserat på ett antal föregående användas. Då kan amplituden som ska kodas minskas ytterligare.

Många signaler kännetecknas av att dess varians och autokorrelation varierar långsamt i tiden. Dessa signaler kallas kvasistationära. Både PCM och DPCM bygger på att signalerna är stationära. Ett sätt att förbättra PCM och göra den användbar även för kvasistationära signaler är att adaptivt uppdatera utstyringsområdet V (med hjälp av skattad varians) och därmed också ändra kvantiseringsintervallen och på det viset ha bra representation av både starka och svaga signaler. Detta kallas *Adaptiv PCM* (APCM). På liknande sätt kan DPCM göras adaptiv (ADPCM) genom att med jämna mellanrum skatta nya prediktorparametrar t.ex med metoder från Hayes.

En annan metod är *Delta modulation* (DM) vilken är en förenklad variant av DPCM men som bara använder en bit vilken talar om huruvida nästa sampel är större eller mindre än det föregående. I båda fallen blir nästa sampel en kvantiseringsnivå högre eller lägre än det aktuella. För att denna metod ska hinna följa variationer i signalen måste signalen ofta översamplas (dvs t.ex samplas med 5 ggr Nyquistfrekvensen). Denna metod kan göras adaptiv (ADM) genom att uppdatera stegstorleken.

1.3.2 Frekvenskodning

Denna typ av kodning bygger på att signalen delas upp i olika subband (frekvensband). För varje band kodas sedan antingen tidssignalen eller dess spektrala karaktäristik (i parameterform). I *Subband Coding* (SBC) representeras signalen av ett antal subband som sedan tidskodas var för sig med någon tidigare beskriven metod t.ex. APCM. Då kan fler bitar användas för de viktigaste subbanden än för de mindre viktiga.

En annan metod är *Adaptive Transform Coding* (ATC) där den samplade signalen delas in i ett antal tidsintervall och där signalen i varje intervall sedan transformeras till frekvensdomänen och sänds över i parameterform. Olika transformer kan användas t.ex. *Karhunen-Loeve transformen* (KLT), DFT eller *Discrete Cosine Transform* (DCT).

1.3.3 Modellbaserad kodning

Jämfört med ovanstående metoder skiljer sig modellbaserad kodning avsevärt. I denna metod modelleras signalen som ett linjärt filter (se signalmodellering i Hayes) vilket om det exciteras av vitt brus producerar den givna signalen. Istället för att skicka över hela signalen skickas nu istället filterparametrarna över. Ofta resulterar denna metod i stor kompression.

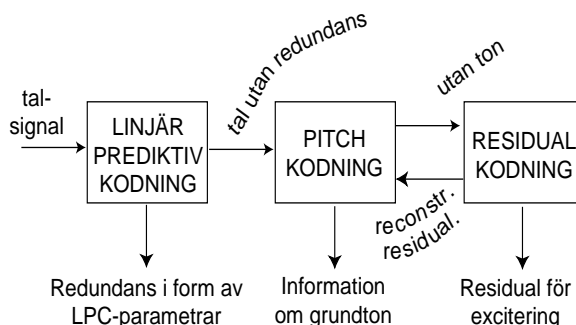
Den vanligaste modellbaserade tekniken kallas *Linear Predictive coding* (LPC) i vilken signalen modelleras som en AR-process (bara poler). När LPC appliceras på talsignaler kallas den ofta för en *vocoder*. Varianter på LPC är *Residual Excited LPC* (RELP) i vilken även filtrets exciteringssignal (residualen) skickas över, *Code-excited LPC* (CELP) där parametrar och residual kodas och skickas över och *Vector-sum Excited LPC* (VSELP). CELP och VSELP använder vektorkvantisering för att koda parametrarna och därmed åstadkomma god kvalitet vid låga bithastigheter.

1.4 Talkodaren

En stor del av vår dagliga kommunikation utgörs av överföring av talsignaler via telefonlinjer och radio- och satellitkanaler. Talsignalen kan anses vara bandbegränsad till <3200 Hz vilket motiverar en samplingsfrekvens på 8000 Hz för alla kodare utom DM. Olika metoder kan jämföras med hjälp av den bithastighet de kräver för att överföra tal med god kvalitet. ADPCM och ADM är effektiva tidskodningsmetoder. ADM kan t.ex. användas ner till 9600 bits/s. Allmänt gäller att bithastigheten med tidskodningsmetoder ligger strax under 16000 bits/s. För hastigheter under 9600 bits/s bör LPC användas. Till exempel skulle i en LPC-talkodare kunna behövas: 1 bit som talar om huruvida ljudet är tonande eller icke tonande. För tonande ljud behöver pitchfrekvensen (ljudets grundton) sändas över. För hyfsad noggrannhet behövs 6 bitar. Dessutom måste en förstärkningsparameter kopplad till pitchfrekvensen skickas med. Denna kan behöva t.ex. 5 bitar om dess värdeområde komprimeras logaritmiskt. Varje prediktor koefficient behöver 8-10 bitar eftersom små variationer i dessa ger stora fel. Med representation i reflexionskoefficienter vilka ligger mellan -1 och 1 kan detta reduceras till 6 bitar. Med filterordning 10 blir det totalt 72 bitar. Då signalen är kvasistationär måste dessa parametrar uppdateras var 15-30 ms vilket totalt resulterar i $2400-4800$ bits/s. Optimalt är det vitt brus som ska excitera dekodern för att rekonstruera talet. Men med tanke på att bara ett fåtal filterparametrar används för att representera signalen förbättras kvaliteten om man även sänder över den signal som ska excitera filtret i dekodern (RELP). Denna brusignal har relativt låg amplitud och kan därför skickas över relativt billigt”.

Kodningstyp	Kvantisering	Kodare	Bithastighet
PCM	Linjär	13 bitar	104000 bits/s
LogPCM	Logaritmisk	7-8 bitar	56000-64000 bits/s
DPCM	Logaritmisk	4-6 bitar	32000-48000 bits/s
ADPCM	Adaptiv	3-4 bitar	24000-32000 bits/s
DM	Binär	1 bit	32000-64000 bits/s
ADM	Binär	1 bit	16000-32000 bits/s
LPC			2400-4800 bits/s

Tabell 1: Olika kodningstyper och den bithastighet de kräver för att representera talsignaler.



Figur 11: Generell struktur i talkodare. Ofta skattas också pitchfrekvensen.

2 Laboration 1: GSM talkodning

2.1 Förberedelseuppgifter

- Repetera följande avsnitt i Hayes Kapitel 4.7.2, 5.2.6.
- Läs avsnitten 'Talkodning' och 'Laboration 1' i detta kompendium.
- Förbered dig på att besvara frågor på ovanstående material muntligen eller skriftligen.
- Förbered dig på att redogöra för GSM talkodarens olika delar.

2.2 Inledning

Global System for Mobile communications (GSM) är ett digitalt mobilkommunikationssystem som snabbt har spridits och tagit marknadsandelar världen över fastän det ursprungligen utvecklades för Europa. Utöver digital överföring innefattar GSM många andra avancerade tjänster och egenskaper t.ex. ISDN kompatibilitet och möjlighet att kommunicera med andra GSM-nät i världen. De avancerade tjänsterna och arkitekturen i GSM har gjort den till en förebild för den framtida tredje generationen av mobiltelefonsystem (efter det analoga t.ex. NMT och det första digitala t.ex. GSM) *Universal Mobil Telecommunication System* (UMTS).

Utvecklingen av GSM inleddes 1982, då Conference of European Posts and Telegraphs (CEPT) startade forskningsgruppen *Groupe Special Mobile* (den ursprungliga betydelsen av GSM). Syftet var att utveckla ett allmänt europeiskt mobiltelefonsystem i 900 MHz-bandet. Vid den här tiden fanns många icke-kompatibla analoga system i Europa. Några av kriterierna för det nya system var:

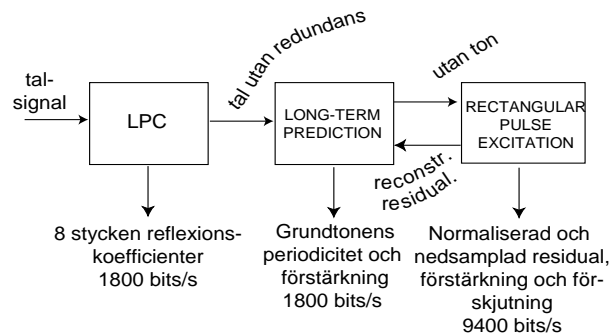
- Bra talkvalitet
- Låg terminal- och underhållskostnad
- Stöd för internationell vidarekoppling till andra GSM-nät
- Handhållna terminaler
- Stöd för nya tjänster
- Spectral effektivitet
- ISDN kompatibilitet

Under 1989, överfördes ansvaret för GSM till *European Telecommunication Standards Institute* (ETSI), och de första rekommendationerna publicerades 1990. Vid den tiden, begärde Storbritannien en rekommendation baserad på GSM men för högre användartäthet och lågeffekt mobilstationer och på 1800 Mhz-bandet. Denna specifikation heter *Digital Cellular System* (DCS1800) och publicerades 1991. Kommersiell användning av GSM-nät startade i mitten av 1991 i flera europeiska länder. I slutet av sommaren 1998, fanns GSM-nät i 120 länder med nära 300 operatörer i Europa, Mellanöstern, Asien, Australien, Afrika, och Sydamerika, och totalt över 100 miljoner användare. Antalet GSM-användare växte samtidigt med 5 miljoner per månad.

I rekommendationen för GSM fastställdes att talkodaren ska vara av typen *13 kbit/s Linear Predictive Coder - Long Term Prediction - Rectangular Pulse Excitation* (LPC-LTP-RPE) där LPC modellerar dynamiken i talapparaten och därmed reducerar överflödigt information i signalen, LTP skattar grundtonen i talet och RPE beskriver hur filterna ska exciteras när talet ska rekonstrueras. Denna talkodare döptes till *GSM Full-rate* (FR) där FR indikerar att bithastigheten 13 kbit/s krävs.

I denna laboration ska vi titta närmare på de talkodningsprinciper som används i dagens GSM-telefoner.

2.3 Funktionsbeskrivning för GSM talkodare



Figur 12: Principskiss för GSM FR talkodare. Grundmetoden är RELP men med extra kodning av grundtonen (LTP).

Som nämntes ovan används i GSM s.k. LPC-LTP-RPE talkodning. Vi ska nu titta närmare på hur varje del i talkodningsprocessen fungerar och vilka signalbehandlingsmetoder som används. Innan några närmare detaljer beskrivs ska vi se till att ha klart för oss hur de olika delarna i talkodaren hänger ihop och repetera de grundläggande principerna som används. Idn är följande: Huvudsakligen finns två typer av ljud tonande och icke-tonande. För tonande ljud genererar stämbanden ett pulståg. Detta har en grundton och har ett linjespektrum. Förstärkningen i detta spektrum faller med 12 db/oktav. Icke-tonande ljud har ett kontinuerligt effektspektrum som avtar med 6 db/oktav.

Talapparaten har sedan ett spektrum med ett antal resonanstopparesultat från ett antal resonansfrekvenser som de olika delarna av talapparaten har. För tonande ljud ökar läpparna förstärkningen med ca 6 dB/oktav. Icke-tonande ljud påverkas inte så mycket eller har viss ökad förstärkning för ökad frekvens. Det resulterande ljudet är alltså för tonande ljud ett linjespektrum som faller med 6 dB (-12+6) och som har ett antal resonanstopparesultat. För icke-tonande ljud blir resultatet ett kontinuerligt bruspektrum som faller med några dB/oktav.

Samma talkodare ska nu användas för båda dessa ljud. Följande signalbehandling måste nu ske:

- Först ska det fallande spektrat rätas upp inför modellering av dynamiken.
- Därefter skattas talapparatsens spektrum (resonanstoppar) med en AR modell (LPC). Dynamiken filtreras sedan bort med motsvarande prediktionsfilter.
- Nästa steg är att ta bort en eventuell ton i signalen. Detta görs med hjälp av långtidsprediktion (LTP).
- Nu återstår för båda typerna av ljud huvudsakligen brus. Men även detta brus sänds över för att rätta till de fel som modellen gör. Denna signal komprimeras ytterligare med hjälp av RPE.
- Den komprimerade residualsignalen används senare i dekodern för att excitera de båda filterna för att rekonstruera talet.

2.3.1 Insignal

Talsignalens väg från mikrofonen till talkodarens ingång är enligt följande: Signalen samplas till 8Khz och representeras med 13 bitar likformig PCM i tvåkomplement. I en del telefoner finns ett mellansteg där signalen representeras med 8 bitar komprimerad (A-lagen) PCM.

2.3.2 Förbehandling

Det första som händer med signalen i talkodaren är att eventuell offset tas bort från den samplade talsignalen. Detta görs med ett notch filter som undertrycker frekvenser kring 0 Hz.

$$H_{offset}(z) = \frac{1 - z^{-1}}{1 - \alpha z^{-1}} \quad \alpha = 32735 * 2^{-15} \quad (24)$$

Därefter går signalen till ett s.k. FIR pre-emphasis filter. Detta är ett HP filter med en förstärkning på cirka 6 dB/oktav som rätar upp spektrat inför modelleringen vilken blir effektivare då olika frekvenser har ungefär samma styrka. Filtret ges av:

$$H_{preemph}(z) = 1 - \beta z^{-1} \quad \beta = 28180 * 2^{-15} \quad (25)$$

Efter denna förbehandling är talsignalen nu redo att kodas.

2.3.3 Linjär prediktiv kodning

I GSM har man valt att uppdatera modellen för talapparaten för varje intervall om 160 sampel (20 ms). Signalen segmenteras därför till 160 sample icke-överlappande intervall. För varje intervall ska nu en AR-modell av talapparatsens dynamik skattas. Ordning 8 har valts dvs

$$V(z) = \frac{1}{A(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_8 z^{-8}} \quad (26)$$

Ett enkelt sätt att skatta dessa filterparametrar är att först skatta korrelationskoefficienterna $r_s(0) - r_s(8)$. Detta görs i GSM specifikationen med hjälp av följande formel (se även Hayes [4.153]).

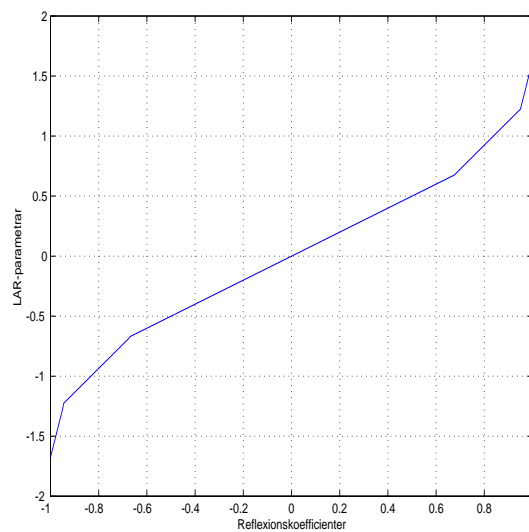
$$r_s(k) = \sum_{i=k}^{159} s(i)s(i-k) \quad k = 0..8 \quad (27)$$

Sedan kan man lätt gå över till $A(z)$ -parametrar med Levinson-Durbin rekursion (Hayes 5.2). Dessa parametrar är emellertid inte lämpliga för kvantisering eftersom de i princip kan vara hur stora som helst. Ska man både kunna täcka stora värden och skillnader mellan små lika viktiga värden krävs slösaktigt många bitar. Reflexionskoefficienter är däremot lämpliga eftersom de ligger mellan -1 och 1. Vergången till reflexionskoefficienter görs med hjälp av Schurs rekursion (Hayes 5.2.6) direkt från korrelation. Eftersom dålig noggrannhet i stora reflexionskoefficienter påverkar resultatet kraftigt vill man här ha tätare kvantisering för värden med absolutbelopp nära 1. En lösning på detta är som vi sett att en kompressor-expander. Vi skapar därför en ny uppsättning parametrar reflexionskoefficienter s.k. Log-Area Ratios (LAR) där stora värden har viktats upp (motsatsen till att vikta upp noggrannhet i mindre värden som diskuterades i avsnittet om olinjär kvantisering). Expanderfunktionen ges av

$$LAR_i = \log_{10}\left(\frac{1 + \gamma_i}{1 - \gamma_i}\right) \quad (28)$$

Denna funktion är dock inte lämplig för implementation eftersom den innehåller logaritm och division. Istället används en linjär approximation av denna funktion (se Fig. 13) för vilken istället addition och multiplikation används vilket är mycket angenämare för hårdvarukonstruktören.

Nu är det dags att kvantisera dessa parametrar innan de skickas vidare till radiosystemet. De olika LAR-parametrarna liksom reflexionskoefficienterna har olika stor betydelse. T.ex. förväntas den första vara mest betydelsefull varför också dess noggrannhet är mest viktig. Både storlek och betydelse avtar ofta sedan efter hand. Därför att det är lämpligt att använda olika antal bitar för att representera de olika parametrarna (de första två får 6 bitar vardera, de följande två 5 bitar vardera, osv). En förutbestämd mappning som är tillgänglig både i kodare och avkodare ser till att värdena ryms i sina värderum.



Figur 13: Expanderfunktion för reflexionskoefficienter som smetar ut värden med belopp nära 1 på ett större intervall.

Dynamiken i signalen är därmed modellerad med hjälp av 8 kvantiserade LAR-parametrar vilka tillsammans kräver 36 bitar för varje segment om 160 sampel. Totalt krävs för detta 1800 bits/s. Dessa ska nu bidra till att rekonstruera signalen i dekodern och skickas därför till radiosystemet.

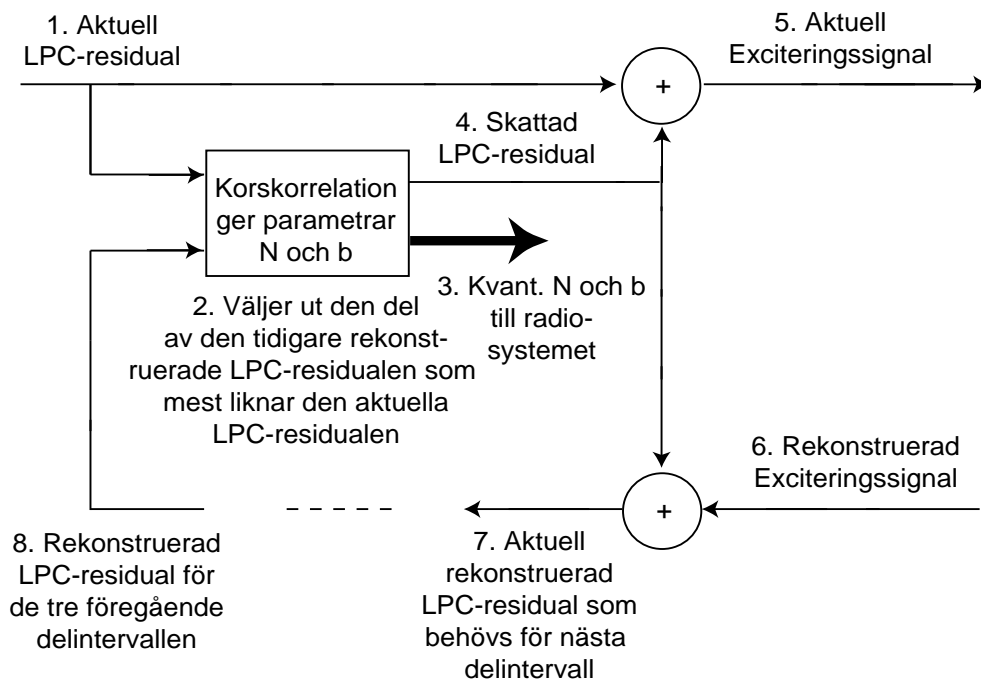
Det som återstår är att ta bort den modellerade informationen ur signalen innan denna går vidare i systemet. Detta görs med ett Lattice filter motsvarande $A(z)$ (Prediktionsfilter). I detta steget är det viktigt att försöka efterlikna det som kommer att hända i dekodern där endast den kvantiserade informationen är tillgänglig. Därför ska de kvantiserade LAR-värdena användas vid denna filtrering. Om så sker kommer residualen efter filtreringen att vara relaterad till de kvantiserade parametrar som sänds över och därmed kommer residual och parametrar att passa ihop vid rekonstruktionen. Residualen kallas från och med nu LPC-residualen.

I denna filtrering ska reflexionskoefficienter användas (Levinson-Durbin i Hayes) och därför måste dessa återskapas från de kvantiserade LAR-parametrarna. LAR-parametrarna mappas tillbaka till sina riktiga värden, körs igenom den inversa expanderfunktionen och blir åter reflexionskoefficienter. Filtreringen görs sedan med de vanliga Lattice formlerna (Hayes 6.2).

För att mjuka till de stegförändringar som uppstår i segmentgränserna viktas det förra segmentets parametrar in något i det 40 första samplen i segmentet. Detta kan göras såvida samma metod används i dekodern.

2.3.4 Pitch-analys

Syftet med denna del är nu att filtrera bort en eventuell grundton i signalen. Det ligger nära till hands att tänka sig att stämbandets frekvens kan variera snabbare än talapparatusens form. Därför delas här varje segment upp ytterligare till 40 sampel (5 ms) segment. För varje delsegment ska nu en korrelationsfaktor N som indikerar en periodicitet i signalen och en tillhörande förstärkning b skattas. LTP-filtret fungerar enligt följande: Betrakta en signal som är skillnaden mellan det som gick in och det som gick ut ur LTP-filtret (dvs det som filtret tog bort eller la till) under de senaste tre delsegmenten. Addera sedan till den brussekvens (excitationssignal) som sändes över för dessa delsegment. Då fås en signal som innehåller dels långtidsinformationen (eftersom det var vad som subtraherades bort av LTP-filtret) och dels en rekonstruerad variant av brussignalen som sändes över till dekodern. Detta är ju i någon mening en rekonstruerad LPC-residual. Parametrarna N och b för det nya delsegmentet bestäms

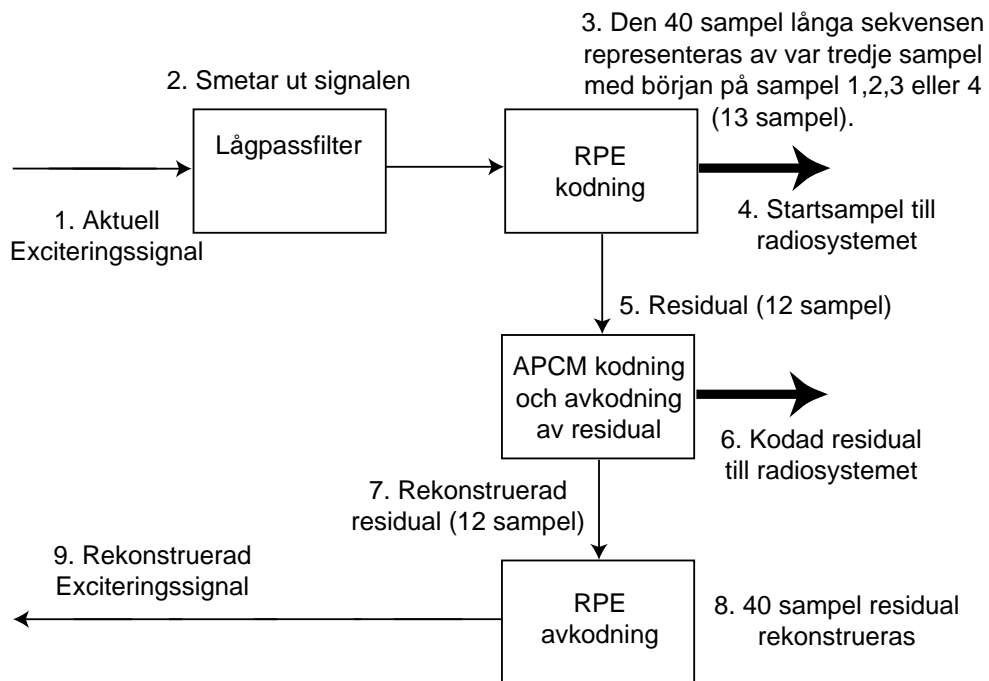


Figur 14: Pitch-analysen (LTP) i GSM FR steg för steg.

nu genom att plocka ut den del av den rekonstruerade LPC-residualen för de tre senaste delsegmenten som mest liknar den nya LPC-residualen och subtrahera en skalad (b) version av den från den aktuella LPC-residualen. Vilken del som passar bäst bestäms genom korskorrelation där N är förskjutningen för vilken korrelationen är störst. Förstärkningen b ser till att utsläckningen blir så bra som möjligt. Den resulterande residualen kallas exciteringssignal eftersom det är den som ska excitera dekodern. Ven här är det de kvantiserade värdena på N och b som används för att skapa residualen. Dessa skickas liksom de kvantiserade LAR-parametrarna vidare till radiosystemet. Förskjutningen N kan anta värden mellan -40 till -120 varför det krävs 7 bitar för att representera dessa. Till förstärkningen används 2 bitar som representerar 4 tabellerade värden (ej likformiga intervall). Totalt krävs här alltså 9 bitar för varje delsegment vilket totalt ger 1800 bits/s.

2.3.5 Kodning av residual

Nu återstår bara att koda exciteringssignalen. Detta görs också för varje delintervall. Till att börja med faltas exciteringssignalen genom ett filter med lågpasskaraktär. Ur den filtrerade signalen skapas nu fyra signaler av längd tretton med början på sampel 1-4 och därefter var tredje sampel. Att plocka ut var tredje sampel bör inte försämra en LP-filtrerad signal. Den av dessa i vilken den totala energin är störst väljs sedan ut att representera residualen. Den tretton bitar långa valda signalen kvantiserar nu med APCM där det största värdet kvantiserar logaritmiskt med 6 bitar (även här används linjär approximation av logaritmen). De tretton bitarna normaliseras sedan till detta största värde och kvantiserar var för sig med 3 bitar. Detta är ett effektivt sätt att få god noggrannhet både för svaga och starka signaler. Denna del av talkodaren behöver nu skicka över vilken av de fyra sekvenserna som valdes. För detta krävs 2 bitar. Normaliseringsvärdet krävde 6 bitar och de tretton residualbitarna 3 bitar vardera. Totalt blir detta 9400 bits/s vilket är den dyrastadeln av de totalt 13 kbits/s som behövs för hela talkodaren.



Figur 15: Residualkodning i GSM FR steg för steg.

2.3.6 GSM FR dekodare

Avkodaren bygger på precis samma principer som kodaren och beskrivs därför inte närmare.

2.4 Tillgängliga MATLAB-filer

```
function ut=convert(in,fs_in,fs_ut,bitar_in,bitar_ut)

% Denna funktion samplar om signalen "in" som har samplingsfrekvens
% "fs_in" och "bitar_in" bitars upplösning till signalen "ut" med
% samplingsfrekvens "fs_ut" och "bitar_ut" bitars upplösning.
%
% Exempel
%
% ut=convert(in,fs_in,fs_ut,bitar_in,bitar_ut)

function [LPCres,KvantLAR]=LPCenc(sig,AntalPar);
%
% Denna funktion tar bort redundans i talsignalen och skapar en LPC-residual.
% Den borttagna informationen finns nu i de kvantiserade filterparametrarna.
%
% sig      - talsignal
% AntalPar - Antal filterparametrar (LAR)
% KvantLAR - De kvantiserade LAR-parametrarna
% LPCres   - Den kvarvarande signalen.
%
% Exempel
%
% [LPCres,KvantLAR]=LPCenc(sig,AntalPar);

function utsig=LPCdec(KvantLAR,RecLPCres);
%
% Denna funktion lägger till den redundans som finns kodad i LAR-parametrarna
% till den rekonstruerade LPC-residualen och
% skapar därmed den rekonstruerade talsignalen.
%
% utsig    - rekonstruerad talsignal
% KvantLAR - De kvantiserade LAR-parametrarna
% RecLPCres - Den rekonstruerade LPC-residualen
%
% Exempel
%
% utsig=LPCdec(KvantLAR,RecLPCres);

function [Excsig,RecExcsig,KvantN,Kvantb,KvantM,KvantMax,KvantExc]=LTPenc(LPCres);
%
% Denna funktion kodar LPC-residualen i grundtonrelaterade parametrar och residual
%
% KvantN    - Förskjutning mellan två toppar i signalen
% Kvantb    - Amplitudförstärkning jämfört med föregående topp
% KvantM    - 0-3 talar om startsampel för de 12 bitar som representerar de 40
```

```

%          ursprungliga (var tredje)
% KvantMax - Talar om vilket utstyrningsområde som gäller vid APCM av residualen
% KvantExc - Värde för de 12 normaliserade samplen
% LPCres   - LPC-residual dvs talsignal från vilken korttidsredundans är borttagen med LPC
% Excsig   - LPC-residual från vilken tonen är bortplockad
% RecExcsig - Detta är Excsig återskapad från den information som finns i de kvantiserade paramet.
%
% Exempel
%
% [Excsig,RecExcsig,KvantN,Kvantb,KvantM,KvantMax,KvantExc]=LTPenc(LPCres);

function RecLPCres=LTPdec(KvantN,Kvantb,KvantM,KvantMax,KvantExc,Res);

% Denna funktion rekonstruerar LPC-residualen från de översända kvantiserade parametrarna
%
% KvantN   - Förskjutning mellan två toppar i signalen
% Kvantb   - Amplitudförstärkning jämfört med föregående topp
% KvantM   - 0-3 talar om startsampel för de 12 bitar som representerar de 40
%           ursprungliga (var tredje)
% KvantMax - Talar om vilket utstyrningsområde som gäller vid APCM av residualen
% KvantExc - Värde för de 12 normaliserade samplen
% RecLPCres - Rekonstruerad LPC residual. Dvs återskapad exciteringssignal med adderad ton
% Res      - 'res' om KvantExc ska användas och 'vit' om brus ska användas.
%
% Exempel
%
% RecLPCres=LTPdec(KvantN,Kvantb,KvantM,KvantMax,KvantExc,Res);

function ut=prefilt(in);
%
% Exempel
%
% ut=prefilt(in);

function ut=postfilt(in);
%
% Exempel
%
% ut=postfilt(in);

```

2.5 Laborationsuppgifter

Starta WaveStudio. Spela in ca 1 sekund av ett tonande ljud (Ä”) och en sekund av ett icke-tonande (S”) genom att:

1. Tryck på =”record” och därefter .
2. Säg vart och ett av ljuden i mikrofonen under en dryg sekund.
3. Tryck sedan .
4. Markera en del av signalen som ser någorlunda stationär ut.
5. Gå därefter in i menyn *Edit* och kopiera *Copy*.
6. Gå in under *File* och skapa en ny fil, *New*.
7. Klistra in signalen med *Paste* under *Edit*.
8. Spara filerna som “a.wav” resp. “s.wav” i mappen *c:\myfiles*.

Starta Matlab och skriv *initosb*.

Uppgift 1

Läs in dina ljud-filer i MATLAB med kommandot *wavread*. Dessa filer är samplade med CD-kvalitet dvs 44.1 kHz och 16 bitars upplösning. Sampla om dem till 8 kHz med 13 bitars upplösning med hjälp av funktionen *convert*. Välj därefter ut ca 1 sekund av vardera ljuden (T.ex. med *signal=signal(1:8000);*).

Uppgift 2

Förbehandla (enligt avsnitt Förbehandling) signalerna med funktionen *prefilt*. Plotta 200 sampel av signalerna före och efter filtret med kommandot *plot*. Vad gör förfiltret? Lyssna på de båda signalerna före och efter filtrering (*sound([signalföre;signalefter]);*).

Uppgift 3

Gör nu LPC-analys med funktionen *LPCenc*. Använd åtta filterparametrar. Plotta LPC-residualen och de kvantiserade LAR-parametrarna. r åtta parametrar ett lämpligt antal? r det någon skillnad mellan det tonande och icke-tonande ljuden? Lägg nu till LPC-residualen sist i vektorn i *sound*-kommandot och lyssna för de båda ljuden. Vad har filtrerats bort?

Uppgift 4

Gör därefter LTP-analys av LPC-residualen med hjälp av funktionen *LTPenc*. Plotta exciteringssignalen och dess kvantiserade motsvarighet *RecExcSig*. Vilken grundton har ditt A-ljud? Vilken information finns i *Kvantb* och *KvantMax*? Lyssna på exciteringssignalen dvs det som är kvar efter LTP-filtrering. Jämför detta med den rekonstruerade exciteringssignalen dvs den som dekodern ser.

Uppgift 5

Plotta 200 sampel av varje signal (före förfilt., efter förfilt, LPC-res, Exc-sig och rekonstruerad exc-sig) i en figur (*subplot*). Kommentera skillnader mellan A och S. Lyssna också på alla signalerna i rätt ordning.

Uppgift 6

Nu är det dags att börja rekonstruera talsignalen. Avkoda de kvantiserade parametrarna och rekonstruera LPC-residualen från dessa. Använd funktionen *LTPdec*. Lyssna på denna och jämför med de tidigare signalerna.

Uppgift 7

Syntetisera nu talsignalen med *LPCdec*. Lyssna på resultatet. Kommentera.

Uppgift 8

Efterbehandla signalen med funktionen *postfilt*.

Uppgift 9

Jämför tidsignal för signalen före och efter talkodning. Lyssna på alla signaler i de olika stegen.

Uppgift 10

Prova att excitera dekodern med vitt brus dvs sänd över talet med 3600 bits/s dvs använd alternativet 'vit' när du använder *LTPdec*. Gör därefter om efterföljande steg.

3 Laboration 2: Spektralanalys

3.1 Förberedelseuppgifter

- I Hayes' bok finns många metoder för spektralanalys. Granska dessa och förbered dig på att principiellt förklara hur de olika metoderna fungerar.
- Försäkra dig om att du förstår MATLAB-koden i filerna *pergram*, *mper* och *welch* (bokens kommando periodogram är omdöpt till pergram ty Matlab innehåller nu ett eget kommando periodogram)

3.2 Inledning

I denna laborationen ska vi undersöka hur olika metoder för spektralanalys användas för att få fram information från ett antal olika okända signaler. Vi börjar med enkla signaler och sedan tittar vi på signalerna i de olika stegen i GSM. Lite mer om GSM finns i kapitel 4 för den intresserade.

3.3 Tillgängliga signal-filer

Utöver de MATLAB-filer som användes i laboration 1 finns ett antal signaler som vi ska titta på.

```
function Px=pergram(x,n1,n2)
% Periodogram enl. Hayes bok.

function Px=mper(x,win,n1,n2);
% Modifierat Periodogram enl. Hayes bok.

function Px=welch(x,L,over,win);
% Welch metod enl. Hayes bok.

afib.mat
% EKG-signal som innehåller förmaksflimmer
```

Dessutom alla filer från Lab 1.

3.4 Laborationsuppgifter

Uppgift 1

Spela in ett A-ljud och en kort mening (typ 4-5 ord) i Wavestudio och spara på samma sätt som i Laboration 1.

Uppgift 2

Sampla om dem till 8 kHz med 13 bitars upplösning med hjälp av funktionen *convert*. Välj därefter ut ca 1 sekund av vardera ljuden A-ljudet och 1.5-2 sekunder av meningen (T.ex. med *signal=signal(1:8000);*).

Uppgift 3

Testa funktionerna *pergram*, *mper* och *welch* på de båda signalerna. Förklara utseendet för de olika metoderna. Eftersom större delen av informationen ligger under 1000 Hz så kan du sampla ner signalerna en faktor 4(*decimate*).

Uppgift 4

Kan du karaktärisera ljuden med hjälp av det spektra som du tagit fram. Vad är det för grundläggande skillnad mellan dessa signaler som gör att frekvensanalys är mer eller mindre meningsfullt?

Uppgift 5

Ladda in signalen *afb.mat*. Kör din Welch-fil. Vad kan du säga om denna signalen? Signalen är samplad med 1000 Hz. Biologiska signaler ligger sällan över 20 Hz. Prova därför att sampla ner den med en faktor 50 *decimate* så att bara information under 10 Hz kommer med. Vad ser du?

Uppgift 6

Modifiera din Welch-fil så att den istället för att medelvärdesbilda alla spektra från de olika tidpunkterna placerar dessa vektorer parallellt i en matris. Då gör filen vad som kallas för tid-frekvens analys. Består förmaksflimret i hjärtsignalen av flera komponenter?

Uppgift 7

Analysera med den nya filen åter A-signalen och meningen. Fås nu någon information om A-ljudet resp meningen? Beskriv några vokaler i frekvenstermer.

Uppgift 8

Kör nu A-signalen och meningen (8 kHz) genom talkodaren. Sampla sedan ner signalerna i de olika stegen i kodaren (insignal,...,RecExcsig) till 2kHz och titta på signalerna med din tid-frekvensanalys-fil. Beskriv vad du ser.

4 Talkodning för tredje generationens mobiltelefoner

4.1 Inledning

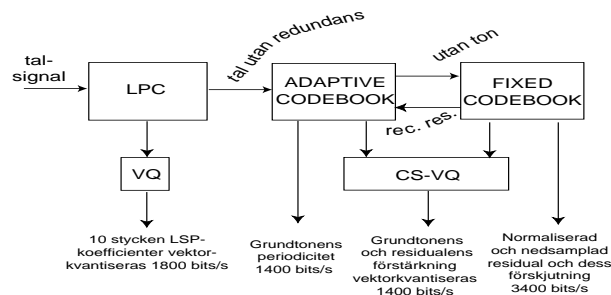
Det är nu tio år sedan de första rekommendationerna för GSM talkodning publicerades. Mycket har hänt inom talkodningsområdet sedan dess och flera talkodningsalgoritmer stöds nu av GSM. Den senaste s.k. *Enhanced Full-rate* (EFR) vilken föreslogs av Nokia under 1996 introducerades i t.ex. Ericssons SH888 (Där stod "ny talteknologi" i reklamen). Enligt amerikanska tester har dess röstöverföringskvalitet varit minst lika bra som det fasta nätets. Den täcker dessutom bättre in talfrekvenserna hos kvinnor och barn än t.ex. FR. Bithastigheten för denna är 12.2 kbits/s dvs ungefär samma som för FR. GSM stödjer också s.k. *Half-rate* (HR) talkodning vilken behöver ungefär halva bithastigheten.

Detta avsnitt beskriver de olika stegen i en av de talkodare som var föreslagna att användas i UMTS-telefonerna när dessa introduceras i början av 2000-talet. Det existerade flera olika förslag på talkodningsstandarder innan valet föll på en.

Som ny standard valdes *Adaptive Multi Rate*, AMR. Adaptive Multirate innebär att datahastigheten anpassas efter hur bra kanalen är. I det mycket brus på kanalen används en låg datahastighet för att det som sänds säkert ska komma fram, medan om kanalen har ett högt signal-brus-förhållande så kan man lägga med mer information i den översända talsignalen. EFR var kandidat till standarden. En tredje huvudkandidat till standarden bygger på samma grundprincip som EFR nämligen *Algebraic CELP* (ACELP). Denna kräver bara 8 kbits/s. ACELP utnyttjar många av de principer som vi har sett i GSM FR men använder vektorkvantisering och överför 16-bitars tal istället för 13-bitars som i FR/EFR. Det är denna talkodare vi ska titta på i detta avsnitt.

4.2 Funktionsbeskrivning för 8 kbit/s ACELP

Denna nya typ av talkodning har i princip samma uppbyggnad som GSM FR men använder andra och mer noggranna metoder i alla tre byggblocken (se Fig 16). ACELP betyder att residualen kodas Algebraiskt och CELP att LPC-parametrarna vektorkvantiseras dvs att olika kombinationer av värden i de olika parametrarna ges en gemensam kod (därför kodexciterad linjär prediktion).



Figur 16: tabell

4.2.1 Insignal

Istället för 13 bitar likformig PCM används här 16 bitars ljudkvalitet. Att överföra tal med 16 bitars kvalitet ökar kraven på noggrannhet och borde öka antalet bitar som krävs för att representera detta men som vi ska se kan detta tjänas in på andra sätt.

4.2.2 Förbehandling

I detta steg skalas signalen ner för att minska risken för överstyrning. Dessutom filtreras offset bort med ett högpassfilter med gränshfrekvensen 80 Hz.

4.2.3 Linjär prediktiv kodning

För att ytterligare förbättra kvaliteten används nu 10 filterparametrar i LPC-analysen istället för 8. Dessutom är intervallen hälften så långa (80 sampel, 10 ms). Detta betyder att LPC-parametrarna uppdateras dubbelt så ofta. Att minska bithastigheten verkar för tillfället tungt.

Tidigare baserades korrelationsberäkningen i LPC-analysen endast på det aktuella 20 ms intervallet medan de i den nya metoden baseras på 30 ms varav 5 ms ligger framför det aktuella intervallet. Ett fönster (typ Hamming) viktas sedan upp informationen i det aktuella intervallet. Viktigt att notera är att all framförhållning introducerar delay i systemet. I telefonsammanhang kan endast mycket korta delay accepteras.

Korrelationsparametrarna omvandlas här till $A(z)$ -polynom istället för reflexionskoefficienter. Dessa omvandlas sedan till s.k. Linjespektrum (LS) frekvenser. Dessa LS-frekvenser är en representation av $A(z)$ där alla parametrar ligger på enhetscirkeln och därför kan representeras av frekvenser. För att ytterligare reducera den information som ska sändas över MA-predikteras de aktuella LS-frekvenserna från parametrarna i de fyra föregående intervallen. Skillnaden mellan de beräknade och de predikterade LS-parametrarna vektorkvantiseras sedan 10-dimensionellt (olika kombination av värden i de tio parametrarna ges olika koder). Inom varje intervall används sedan ett medelvärde av dessa parametrar och de för det föregående intervallet under första halvan av intervallet medan endast de aktuella parametrarna används under den andra. För varje intervall om 10 ms skickas med denna metod 18 bitar till radiosystemet. Bithastigheten för den förbättrade LPC-analysen med 2 ytterligare parametrar, högre kvalitetskrav och dubbelt så tät uppdatering blir därmed 1800 bits/s vilket är samma som i GSM FR.

4.2.4 Pitch-analys

För att bestämma pitchfrekvensen används här en viktad talsignal. Denna viktning bygger på LPC-parametrarna och lyfter fram pitchfrekvensen. Skattningen av pitchfrekvensen sker i två steg. Först skattas den ungefärligt med dålig noggrannhet med autokorrelation i den viktade signalen dvs inte med korskorrelation med hjälp av en rekonstruerad sänd signal som i FR. Först därefter korskorreleras den med den rekonstruerade signalen men nu med 3-faldig noggrannhet inom ett litet intervall kring den preliminära grundtonen. På samma sätt som tidigare subtraheras pitchfrekvensens bidrag till signalen med en pitchförstärkningsparameter. Pitchfrekvensen uppdateras för varje 5 ms intervall precis som i FR. För vartannat delintervall används 9 bitar (absolut angivelse) och för vartannat 5 bitar (relativt det föregående). Denna metod kallas Adaptive Codebook och kräver 1400 bits/s. Pitchförstärkningsparametern kodas senare tillsammans med en förstärkningsparameter i residualkodningen.

4.2.5 Kodning av residual

Residualkodningen skiljer sig ganska mycket från den som användes i GSM FR där var tredje sample fick representera residualen. Detta krävde många bitar vilket vi inte har råd med här. Här väljs 4 sampel (av 40) ut att representera residualen. Dessa kan bara anta värdena +1 och -1. En gemensam förstärkningsparameter anpassar sedan hela amplitud så att den representerar residualen så bra som möjligt. De 4 sampel som valts ut kan inte placeras helt godtyckligt i tiden. En s.k. Fixed Codebook säger att den första måste ligga på plats 1 + en multipel av 5, den andra på plats 2 + en multipel av 5 osv. Tecken och plats måste här skickas över. För detta krävs 17 bitar per 5 ms vilket blir 3400 bits/s.

De båda förstärkningsparametrarna vektorkvantiseras och kräver 7 bitar/5 ms dvs 1400 bits/s. Totalt blir detta 8 kbits/s.