

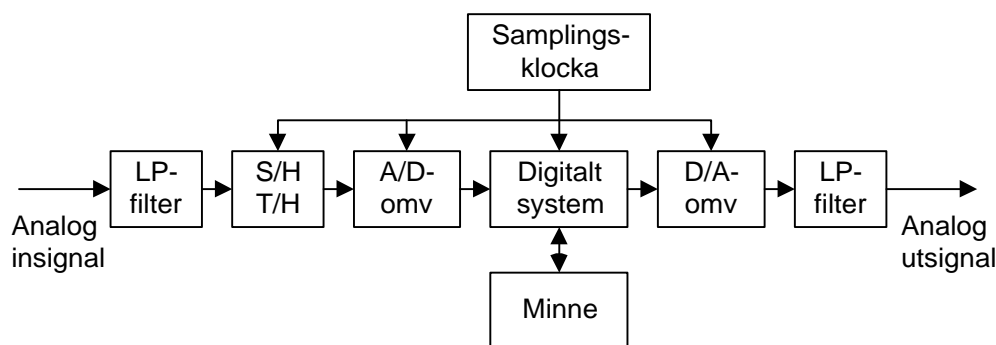
2 Grundbegrepp

Vi kommer i detta kapitel att studera de grundblock som finns i ett typiskt system för tidsdiskret signalbehandling där vi oftast antar att signalerna från början är analoga, vilket betyder att de före signalbehandlingen måste omvandlas till digitala signaler. Vi antar också oftast att vi efter signalbehandlingen åter vill ha analoga signaler varför signalerna måste återomvandlas.

Vi kommer också att se på förutsättningarna för att göra dessa omvandlingar utan att förlora information.

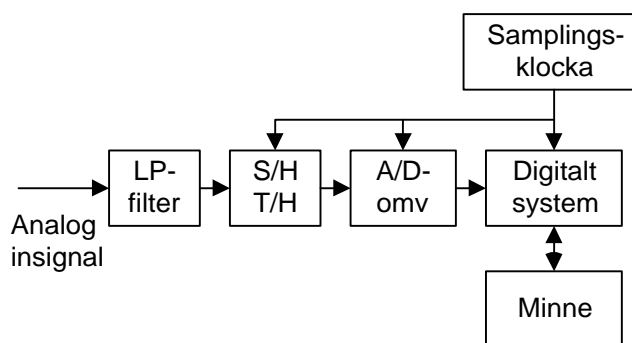
2.1 Typiskt analog/digitalt system

Låt oss se hur ett typiskt system för DSP kan se ut om signalerna från början är analoga och om vi önskar återskapa analoga signaler efter DSP-beräkningarna *Figur 2.1*. Vi kommer att senare i kapitlet förklara de olika blockens funktion och varför de måste ingå i signalkedjan.



Figur 2.1 Generellt digitalt system för analoga in-och utsignaler

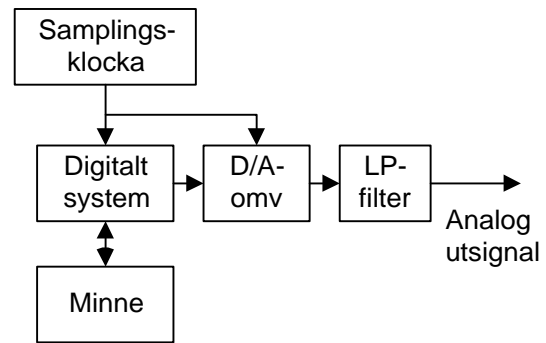
Vi kan också tänka oss att begränsa oss till delar av detta system. Är vi t ex bara intresserade av att analysera signalen, t ex bestämma dess frekvensinnehåll så kommer vi att göra denna analys i DSP-beräkningarna och resultatet presenteras där t ex på en datorskärm. Vi har alltså inget behov av att återskapa en analog signal, *Figur 2.2*.



Figur 2.2 Digitalt system för analoga insignaler

På samma sätt kan vi ha ett delsystem där vi genererar digitala signaler och önskar omvandla dessa till analoga motsvarigheter. Då bortfaller den första omvandlingen från analog till digital signal, *Figur 2.3*. Denna funktion förekommer till exempel i de flesta moderna syntar och andra ljudgenererande apparater.

De två delsystemen kan finnas i samma system men användas vid olika tillfällen, t ex inläsning och lagring vid ett tillfälle och utläsning vid ett annat. I introduktionen framhöll vi att det är lätt att byta funktion i ett DSP-system bara hårdvarudelarna finns på plats.



Figur 2.3 Digitalt system för analoga utsignaler

2.2 Sampling

Då vi studerar en vanlig tidskontinuerlig signal, t ex med hjälp av ett (analogt) oscilloskop, så kan vi vid varje tidpunkt registrera hur signalen uppför sig. Vi låter alltså tiden flyta och ur betraktelsesynpunkt är ingen tidpunkt skild från någon annan, bortsett från att signalen förändrar sig, och vi kan vid vilken tidpunkt som helst ange signalens värde, så gott vi nu klarar av att mäta den. Analoga system, t ex passiva eller aktiva filter, använder sig också av dessa tidskontinuerliga signaler och kan omedelbart reagera på en förändrad insignal. Man kan då ställa sig frågan om det är möjligt att lätta på kraven på att kontinuerligt hålla koll på signalen och om vi i stället kunde nöja oss med att registrera signalens värde då och då? Frågan är alltså om detta är möjligt och hur ofta vi i så fall måste registrera signalen.

Vi skall se att det faktiskt är möjligt att bara mäta då och då utan att förlora information. Hur ofta bör vi då mäta? Det bör säga sig självt att antalet nödvändiga mätningar per tidsenhet varierar från applikation till applikation och är beroende av vilken förändringshastighet (vilket frekvensinnehåll) vi kan förvänta oss i det aktuella fallet. Vi behöver rimligen inte kontrollera kursen lika ofta då vi rör en eka som då vi kör en snabb motorbåt, den hastighet med vilken det kan gå snett (hamna på ett grund) är ju helt olika.

Då vi börjar arbeta med digitala system, t ex datorer, vars funktion bygger på en systemklocka som gör att vi bara kan utföra en operation då vi får en puls från klockan så hamnar vi automatiskt i en situation där klockan har styckat upp tiden och gjort tidsförloppet diskret vilket gör att vi inte kan göra en mätning vid vilken tidpunkt som helst. Önskar vi dessutom göra någon form av signalbehandling så kräver detta några programsatser som kommer att ta viss tid och vill vi klara av signalbehandlingen innan vi tar in nästa mätvärde, vilket vi normalt vill, så måste vi ytterligare utsträcka tiden mellan två mätningar.

I exemplen med båtarna ovan så kontrollerar vi väl kursen då och då men det är inte så viktigt exakt vid vilka tidpunkter kontrollerna sker, bara det sker tillräckligt ofta. Vi kommer i detta fall förmodligen också att kontrollera kursen med ojämna intervall, ute på öppet hav behöver vi inte kolla kursen så ofta som i en smal passage. För att kunna räkna på våra signaler och bygga upp en matematisk modell av våra signaler och system så blir det dock mycket enklare om vi gör våra kontroller med jämna tidsintervall vilket innebär att om vi tidskontinuerligt kan beskriva vår signal med funktionen $x(t)$ så gör vi nu övergången till en tidsdiskret signal genom transformeringen

$$x(t) \rightarrow x[n \cdot T] = x[n]$$

där n är ett heltal och T är tiden mellan två registreringar av signalens värde. Vi tar alltså stickprov på signalen med jämna tidsintervall. Vi säger att vi *sample* signalen med samplingsperioden, samplingsintervallet T , tiden mellan två samplings är T , vilket gör att vi kan uttrycka samplingsfrekvensen f_s , antalet samplings per tidsenhet (normalt sekund), som

$$f_s = \frac{1}{T}$$

dvs samplingsfrekvensen har som andra frekvenser enheten Hertz.

I fortsättningen kommer vi konsekvent att låta samplingsperioden vara underförstådd i våra uttryck och vi skriver som vi nämnde i *Kapitel 1 Introduktion* $x[n]$ i stället för $x[n \cdot T]$.

Lägg märke till att vi använder den tidigare angivna konventionen att använda vanliga parenteser för argumentet hos den tidskontinuerliga signalen medan vi använder hakparenteser för den uppräknade, tidsdiskreta, samplede signalens argument.

Exempel, Bilaga 2.1

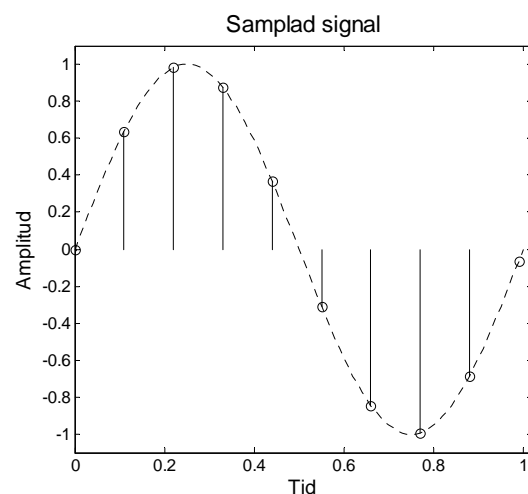
2.2.1 Sample and Hold

Vi kommer alltså att avläsa insignalen en gång per samplingsperiod. Det gör vi genom att låta insignalen gå genom en elektroniskt styrd brytare, där styrningen sköts av samplingsklockan som då sluter brytaren under en kort del av varje samplingsperiod, *Figur 2.4*.

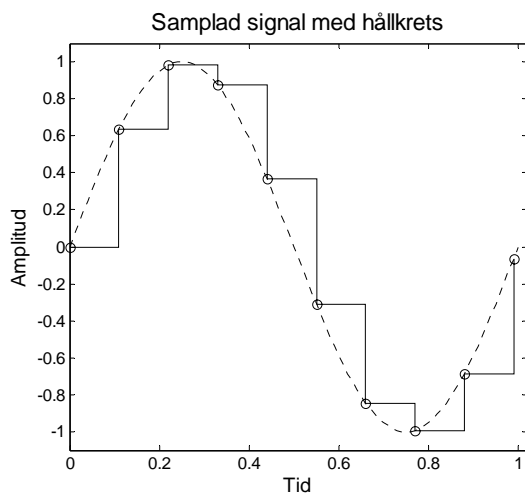
De vanligaste typerna av A/D-omvandlare måste ha en fast nivå på insignalen under den tid då omvandlingen sker. Detta gör att vi måste kunna hålla kvar den avlästa signalnivån under resten av samplingsperioden. Det gör vi genom att låta insignalen ladda upp en kondensator. Vi minns från elläran att denna sker med en tidskonstant $\tau = R \cdot C$, där C är kondensatorns värde medan R är den resistans som sitter i serie med kondensatorn. Har nu objektet som vi mäter på en stor inre resistans så kommer tidskonstanten att bli stor och det tar lång tid att ladda upp kondensatorn helt och vi måste vänta innan signalen blir trovärdig. Vi löser detta genom att sätta en krets med låg utimpedans före brytaren. Kretsen har också hög inimpedans så att den inte belastar mätobjektet. Kretsen kallas för en *buffert*.

På samma sätt kommer den efterföljande A/D-omvandlaren att ladda ur kondensatorn om den inte har hög inimpedans (återigen tidskonstanten $\tau = R \cdot C$). Även här sätter vi in en buffert, i detta fall för att kondensatorn skall 'se' en hög impedans och inte laddas ur under samplingsperioden, *Figur 2.5*.

Den krets som vi har skisserat kallas för en *Sample and Hold-krets* (S/H-krets) och visas i *Figur 2.6*.

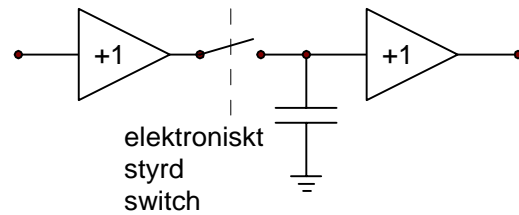


Figur 2.4 Analog insignal och samplad signal

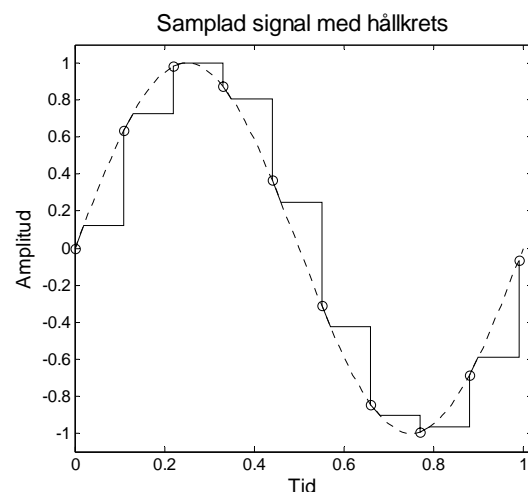


Figur 2.5 Analog insignal och samplad signal från hållkrets

Den ideala S/H-kretsen skall bara sluta brytaren under ett mycket kort ögonblick. I verkligheten kommer den oftast att vara sluten under en lite längre tid och under denna tid kommer kondensatorns laddning att följa inspänningens variation (Figur 2.7). Detta gör att kretsen ibland också kallas en *Track and Hold*-krets.



Figur 2.6 Sample and Hold-krets



Figur 2.7 Analog insignal och signal från Track and Hold-krets

2.2.2 Samplingsvillkor

Vi måste nu ställa oss frågan: Hur ofta måste vi mäta på en signal för att de stickprov, de sampel vi tar skall ge en korrekt beskrivning av signalen? Det kan väl synas naturligt att ju oftare vi mäter ju bättre bild av signalen får vi. Dessutom måste rimligen de frekvenser som ingår i signalen påverka resultatets korrekthet.

Låt oss för enkelhetens skull betrakta en signal som innehåller en enda frekvens, dvs en sinus- eller cosinusformad signal. Det kommer att visa sig att den intuitiva behandlingen och tolkningen av en cosinussignal är enklare än tolkningen av en sinussignal varför vi börjar med den förra. Eftersom cosinus- och sinusfunktioner i grunden är samma sak, de skiljs bara genom en färförskjutning så skall vi sedan få fram ett mer generellt resultat.

2.2.2.1 Cosinusformad signal

Vi har alltså den tidskontinuerliga signalen

$$x(t) = A \cdot \cos(\omega \cdot t) = A \cdot \cos(2 \cdot \pi \cdot f \cdot t)$$

Signalamplituden har ingen inverkan på tidsförloppet så vi kan för enkelhetens skull välja amplituden ett ($A = 1$).

Vi samplar nu denna signal, dvs vi gör signalen tidsdiskret genom att ta signalens värden vid tidpunkterna

$$t = n \cdot T = n \cdot \frac{1}{f_s}$$

där n är ett heltal, T är samplingsperioden (tiden mellan två samplingstillfällen) och f_s är samplingsfrekvensen. Vi får

$$x[n \cdot T] = x[n] = 1 \cdot \cos(2 \cdot \pi \cdot f \cdot n \cdot T) = \cos\left(2 \cdot \pi \cdot n \cdot \frac{f}{f_s}\right)$$

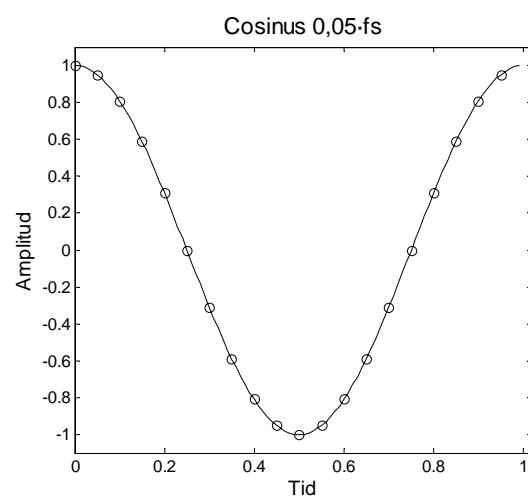
Vi har alltså ett uttryck som inte bygger på den verkliga frekvensen f utan uttrycket bygger på signalfrekvensen relativt samplingsfrekvensen, *den relativa signalfrekvensen* $\frac{f}{f_s}$ och det är en viktig allmän lärdom att det är den relativa och inte den absoluta, verkliga frekvensen som är viktig i samplade system. På samma sätt använder vi inte den absoluta vinkelfrekvensen ω utan *den relativa vinkelfrekvensen*

$$\Omega = \frac{\omega}{f_s} = 2 \cdot \pi \cdot \frac{f}{f_s}.$$

Observera att vi kommer att använda Ω för att beteckna den relativa vinkelfrekvensen medan den verkliga vinkelfrekvensen som tidigare betecknas ω .

Om samplingsfrekvensen är hög i förhållande till signalfrekvensen så kommer vi att göra många registreringar av signalen under en signalperiod och vi kan beskriva signalen noggrant. Lagg åter märke till att samplingsfrekvensen i sig inte är viktig utan det viktiga är förhållandet mellan signalfrekvens och samplingsfrekvens, den relativa signal-

frekvensen, dvs kvoten $\frac{f}{f_s}$.



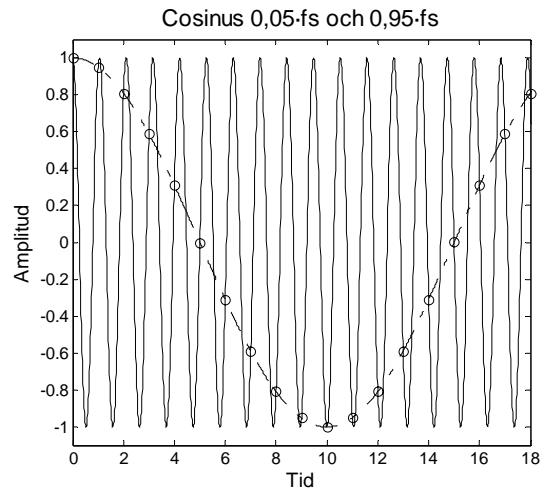
Figur 2.8 Samplad cosinussignal med $f_s = 20 \cdot f_{\text{signal}}$

Figur 2.8 visar förloppet då samplingsfrekvensen är 20 gånger högre än signalfrekvensen.

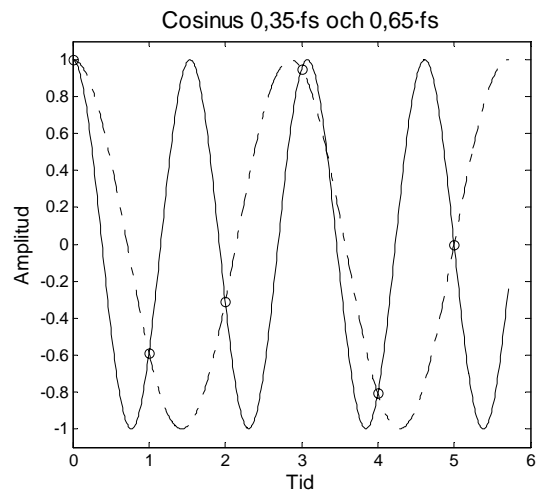
Vi ser att det är inga problem att tolka signalen som en cosinuskurva. Det är också lätt att inse att vi inte kan hitta någon signal med någon närliggande frekvens som skulle gå genom *exakt* samma samplingspunkter, det är väl dock inte otroligt att det skulle kunna gå att passa in någon högfrekvent signal till samma punkter om den har så pass hög frekvens att vi får en eller flera perioder av denna signal mellan varje samplingspunkt och i *Figur 2.9* ser vi ett fall då detta är möjligt, här är signalfrekvensen $0,95 \cdot f_s$. Man kan väl dessutom misstänka att det finns ytterligare signaler som ger samma samplingspunkter om vi fortsätter att öka signalfrekvensen.

Vi kan alltså misstänka att det rent generellt gäller att om vi har för höga signalfrekvenser i förhållande till samplingsfrekvensen så kan minst två olika frekvenser (en lågfrekvent och en högfrekvent) ge samma samplingspunkter vilket gör att vi inte kan säga vilken av frekvenserna som har givit upphov till samplingspunkterna och som är den korrekta, samplade signalen.

Höjer vi signalfrekvensen i förhållande till samplingsfrekvensen så kommer vi att få färre samplingspunkter under en signalperiod men i *Figur 2.10*, där förhållandet mellan signalfrekvens och samplingsfrekvens är 0,35 så är det faktiskt fortfarande så att vi måste gå en bra bit uppåt i frekvens för att hitta en annan signalfrekvens som ger samma samplingspunkter. Lite experimenterande ger att vi får samma samplingspunkter om vi använder signalfrekvensen $0,65 \cdot f_s$.

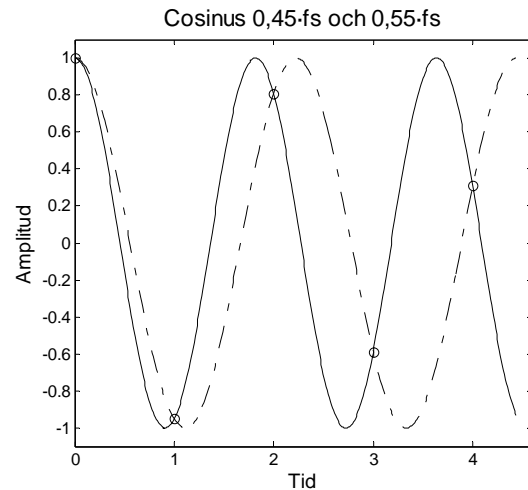


Figur 2.9 Samplad högfrekvent cosinus-signal, frekvens större än $\frac{f_s}{2}$, som ger samma samplingspunkter som en cosinus-signal med $f_s = 20 \cdot f_{\text{signal}}$



Figur 2.10 Samplade cosinussignaler med frekvenser $0,35 \cdot f_s$ och $0,65 \cdot f_s$

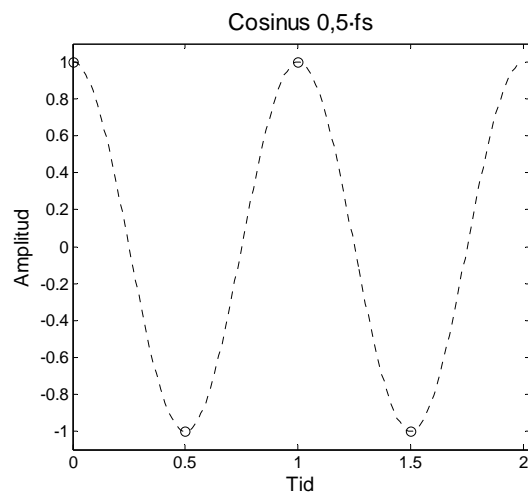
Vi gör också ett experiment med signal-frekvensen $0,45 \cdot f_s$ för att se om vi utifrån dessa tre tester kan hitta något samband mellan den höga och den låga frekvensen som ger samma samplingspunkter. Det visar sig då att vi i detta fall får samma samplingspunkter för signal-frekvensen $0,55 \cdot f_s$, *Figur 2.11*.



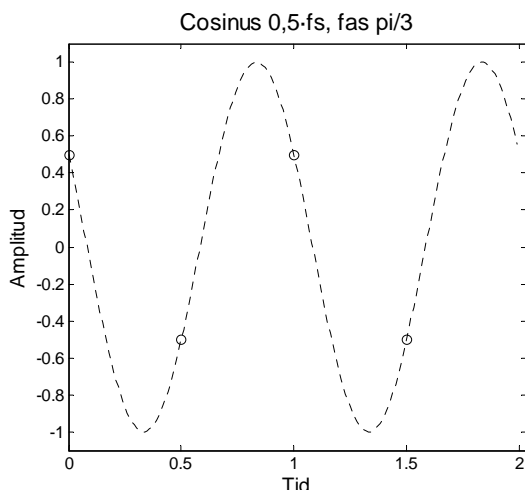
Figur 2.11 Samplade cosinussignaler med frekvenser $0,45 \cdot f_s$ och $0,55 \cdot f_s$

Tar vi nu en samplingsfrekvens som är dubbla signal-frekvensen så får vi *Figur 2.12*, dvs vår samplade signal blir alternerande \pm signalamplituden.

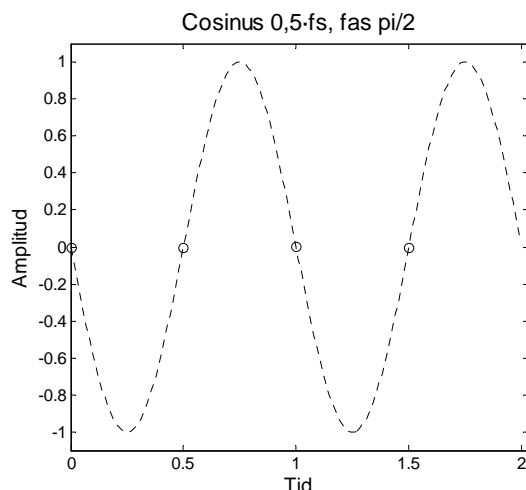
Låt oss här med en gång konstatera att resultatet blir direkt beroende av fasläget hos den samplade signalen, se *Figur 2.12 – 14*. Vi ser i *Figur 2.14*, som visar en sinussignal, att vi i detta fall alltid får samplingsvärdet noll. Slutsatsen måste bli att resultatet för detta förhållande mellan signal-frekvens och samplingsfrekvens inte är tillförlitligt.



Figur 2.12 Sampling, $f_s = 2 \cdot f_{\text{signal}}$, signal samplad i fasläge 0 radianer



Figur 2.13 Sampling, $f_s = 2 \cdot f_{\text{signal}}$, signal
 samplad i fasläge $\frac{\pi}{3}$ radianer



Figur 2.14 Sampling, $f_s = 2 \cdot f_{\text{signal}}$, signal
 samplad i fasläge $\frac{\pi}{2}$ radianer

Vi fortsätter att höja signalfrekvensen och ser på resultatet om $f_{\text{signal}} = 0,55 \cdot f_s$ och finner som vi sagt tidigare att detta ger samma resultat som vi fick för signalen med $f_{\text{signal}} = 0,45 \cdot f_s$ och vid samplingen kommer signalen då att tolkas som om den har den lägre frekvensen, det finns ju ingen anledning för systemet att tro att signalen gör några extra svängar mellan samplingspunkterna.

Fortsätter vi att höja signalfrekvensen så skall vi finna att det finns ytterligare signaler som ger samma samplingspunkter. Vi kommer t ex att få samma samplingspunkter som för signalerna med frekvenserna $f_{\text{signal}} = 0,45 \cdot f_s$ och $f_{\text{signal}} = 0,55 \cdot f_s$ om vi har signalfrekvenserna $f_{\text{signal}} = 1,45 \cdot f_s$ eller $f_{\text{signal}} = 1,55 \cdot f_s$.

Vår slutsats måste av detta måste bli att för att vi skall kunna tolka våra cosinussignaler på ett korrekt och unikt sätt så måste deras frekvenser vara *lägre än* $f_s/2$.

2.2.2.2 Sinusformad signal

Vi sa ovan att förhållandena blir något mer komplicerade då vi har en sinusformad signal. Låt oss ändå se på detta fall. Vi har alltså en sinusformad signal

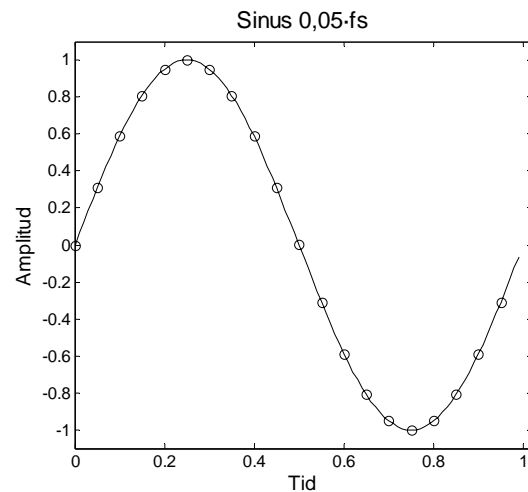
$$x(t) = A \cdot \sin(\omega \cdot t) = A \cdot \sin(2 \cdot \pi \cdot f \cdot t)$$

som vi samplar. Vi använder återigen amplituden 1 ($A = 1$).

$$x[n] = \sin\left(2 \cdot \pi \cdot n \cdot \frac{f}{f_s}\right)$$

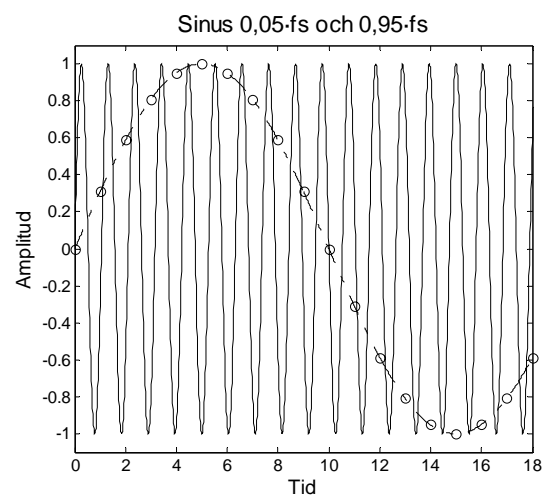
Låt oss se på samma signalfrekvenser (i förhållande till samplingsfrekvensen) som i cosinusfallet.

Vi börjar alltså med en sinussignal med en frekvens som är en tjugondel av samplingsfrekvensen. Vi ser i [Figur 2.15](#) att det även här är ganska lätt att identifiera signalfrekvensen ur de samplade punkterna.



Figur 2.15 Samplad sinussignal med $f_s = 20 \cdot f_{signal}$

Låt oss nu försöka hitta den första hög-frekventa sinussignal som har samma samplingspunkter. Rimligen borde detta vara samma frekvens som i cosinusfallet, dvs $0,95 \cdot f_s$. Vi provar och får [Figur 2.16](#), dvs samplingspunkterna sammanfaller *inte*.



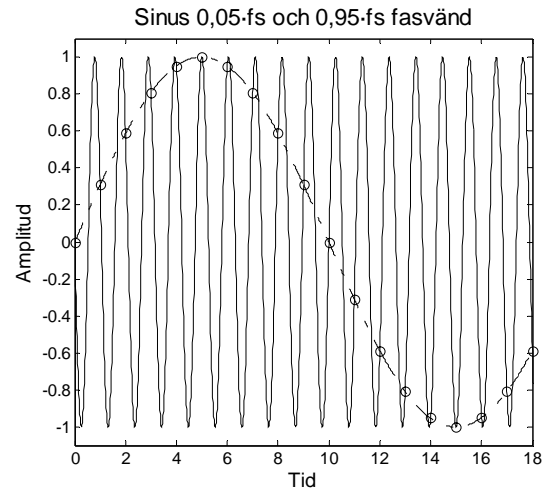
Figur 2.16 Samplad högfrekvent sinus-signal, frekvens större än $f_s/2$, som borde ge samma samplingspunkter som en sinussignal med $f_s = 20 \cdot f_{signal}$

Lite experimenterande visar dock att samplingspunkterna för dessa två kurvor sammanfaller om vi fasvrider en av signalerna, t ex den högfrekventa signalen, 180° , vilket är det samma som att sätta minustecken på signalen

$$x[n] = \sin\left(2 \cdot \pi \cdot n \cdot \frac{f}{f_s} - 180^\circ\right) =$$

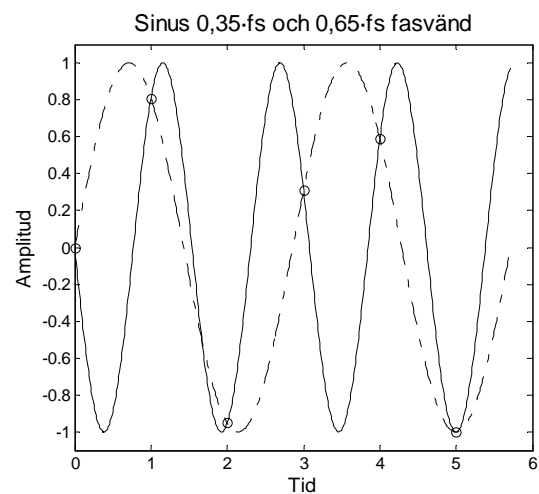
$$= -\sin\left(2 \cdot \pi \cdot n \cdot \frac{f}{f_s}\right)$$

Vi får [Figur 2.17](#).



Figur 2.17 Samplad 180° fasvriden högfrekvent sinussignal, frekvens större än $f_s/2$, som ger samma samplingspunkter som en sinussignal med med $f_s = 20 \cdot f_{\text{signal}}$

Tar vi nu som i cosinusfallet en signal med frekvensen $0,35 \cdot f_s$ så hittar vi även här den första högfrekventa sinussignalen med sammanfallande samplingspunkter vid frekvensen $0,65 \cdot f_s$ men denna signal måste även i detta fall ha negativt tecken, [Figur 2.18](#).



Figur 2.18 Samplade sinussignaler med frekvenser $0,35 \cdot f_s$ och $0,65 \cdot f_s$, högfrekvent signal med negativt tecken

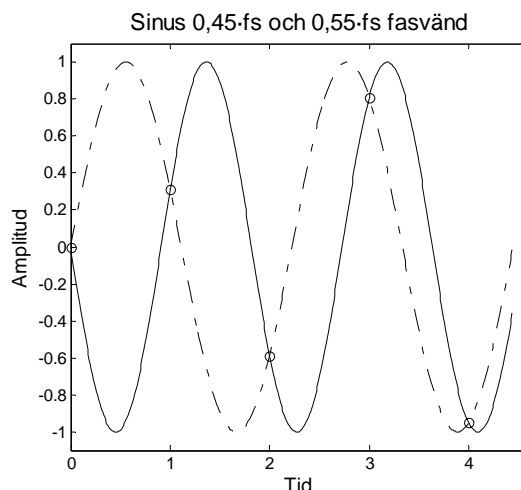
På samma sätt ger signalfrekvenserna $0,45 \cdot f_s$ och $0,55 \cdot f_s$ samma samplingspunkter om den högfrekventa signalen ges ombytt tecken, *Figur 2.19*.

En sinusformad signal med signalfrekvens lika med halva samplingsfrekvensen såg vi i *Figur 2.14*.

Fortsätter vi även här att höja signalfrekvensen så kommer som i cosinusfallet signaler med frekvenserna $1,45 \cdot f_s$ och $1,55 \cdot f_s$ ge samma samplingspunkter som signalen med $0,45 \cdot f_s$ och här behöver den högfrekventa signalen *inte* fasvändas.

Vi kan alltså även för sinusformade signaler konstatera att samplingsfrekvensen måste vara mer än dubbelt så hög som signalfrekvensen för att signalen skall tolkas korrekt.

Vi har därmed formulerat *samlingsvillkoret*



Figur 2.19 Samplade sinussignaler med frekvenser $0,45 \cdot f_s$ och $0,55 \cdot f_s$, högfrekvent signal med negativt tecken

Samplingsfrekvensen måste vara mer än dubbelt så hög som den högsta signalfrekvensen för att samplingen skall ge ett korrekt resultat

$$f_{\text{sampling}} > 2 \cdot f_{\text{signal,max}}$$

Teoremet kallas ibland *Shannons teorem* efter den man som 1949 formulerade villkoret. Villkoret hade dock redan 1928 visats av den svenskättlade amerikanen Harry Nyquist. Ibland ser man i signalbehandlingslitteraturen villkoret

$$f_{\text{sampling}} \geq 2 \cdot f_{\text{signal,max}}$$

men likhet i uttrycket förutsätter att vi använder oändligt många samplingspunkter för att definiera signalen så i praktiken gäller strikt olikhet.

Den högsta tillåtna signalfrekvensen (vid given samplingsfrekvens), alltså $f_s/2$, kallas ibland *vikningsfrekvensen* eller *Nyquistfrekvensen*. Förklaringen till ordet vikning kommer vi strax till.

2.2.3 Vikning (spegling, aliasing)

Vad finns det nu för samband mellan de frekvenser som ger samma samplingspunkter och varför uppför sig sinus- och cosinusformadesignaler olika? Och varför är förhållandena olika vid olika frekvenser (fasvändning respektive inte fasvändning av sinusignalen)?

Låt oss börja med att studera sambandet mellan frekvenserna och mer eller mindre utelämnas fasen så länge.

2.2.3.1 Vikningsfrekvenser

Vi såg tidigare att signalfrekvenserna $0,35 \cdot f_s$ och $0,65 \cdot f_s$ gav samma samplingspunkter i cosinusfallet och även i sinusfallet om vi bytte tecken på den ena signalen vilket är det samma som att ge den 180° fasvridning. På samma sätt fick vi samma samplingspunkter för signalfrekvenserna $0,45 \cdot f_s$ och $0,55 \cdot f_s$. Återigen får vi fasvända den ena signalen i sinusfallet.

Lite tankearbete ger att den lågfrekventa signalen ligger lika långt nedanför halva samplingsfrekvensen som den högfrekventa signalen ligger ovanför $f_s/2$. Vi säger att den högfrekventa signalen *speglar sig* i $f_s/2$. Vi kan alternativt säga att den *viker sig* runt

$f_s/2$ och vi talar om *spiegling* eller *vikning* (på engelska *aliasing*).

Kan vi formulera ett generellt samband? Ja det kan vi. Om den verkliga signalfrekvensen ligger i intervallet $f_s/2 - f_s$ så kommer den att tolkas som om den har frekvensen

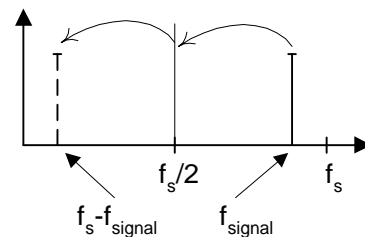
$$f'_{signal} = f_s - f_{signal}$$

dvs den tolkas som liggande i intervallet $0 - f_s/2$,

Figur 2.20.

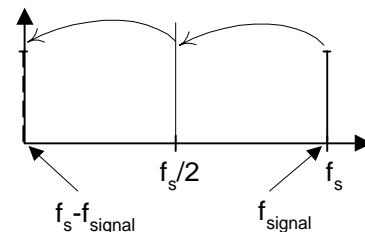
Som vi ser av figuren så får vi det märkliga fenomenet att då vi i detta högfrekventa intervall höjer signalfrekvensen så kommer frekvensen att tolkas som en allt lägre frekvens till dess att fallet $f_{signal} = f_s$ gör att signalen tolkas som om den vore en likspänning, *Figur 2.21.*

Vad gäller nu om vi har signalfrekvenser högre än samplingsfrekvensen? Vi kan visa att signalfrekvenser i intervallet $f_s - 3 \cdot f_s/2$ kommer att tolkas som signaler med frekvenser givna av signalfrekvensen minus samplingsfrekvensen ($f_{signal} - f_s$) och då har den högfrekventa och den lågfrekventa signalen samma fasläge, det krävs ingen fasvändning för vare sig cosinus- eller sinussignaler.



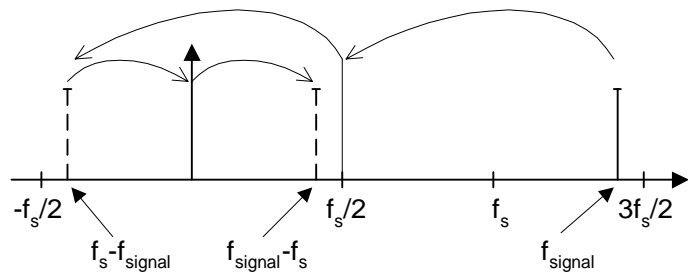
Figur 2.20 Vikning av signal med frekvens i intervallet

$$f_s/2 < f < f_s$$



Figur 2.21 Samplad signal med $f_{signal} = f_s$

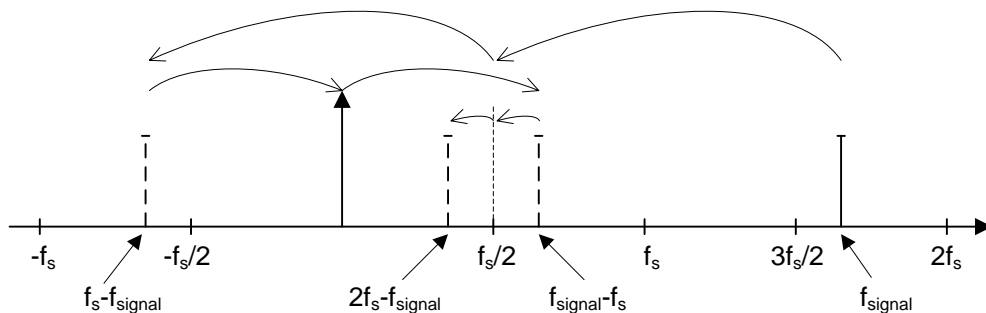
Vi kan som i [Figur 2.22](#) göra tolkningen att frekvensen viks ner till en negativ frekvens $f_s - f_{signal}$ i frekvensintervallet $-\frac{f_s}{2}$ till 0, för att sedan vikas ytterligare en gång till frekvensen $f_{signal} - f_s$ i intervallet $0 - \frac{f_s}{2}$.



Figur 2.22 Vikning av signal med frekvens i intervallet $f_s/2 < f < 3 \cdot f_s/2$

På samma sätt kan vi göra

tolkningen att signaler i intervallet $3 \cdot \frac{f_s}{2} - 2 \cdot f_s$ kommer att vikas ner till den negativa frekvensen $f_s - f_{signal}$ i intervallet $-f_s$ till $-\frac{f_s}{2}$ för att sedan vikas upp till frekvensen $f_{signal} - f_s$ i intervallet $\frac{f_s}{2} - f_s$ för att slutligen tolkas som liggande vid frekvensen $2 \cdot f_s - f_{signal}$ nedanför halva samplingsfrekvensen via ny vikning, [Figur 2.23](#).



Figur 2.23 Vikning av signal med frekvens i intervallet $3 \cdot \frac{f_s}{2} < f < 2 \cdot f_s$

Vi kan generellt formulera sambandet så att om vi har en signalfrekvens f_{signal} som är större än $\frac{f_s}{2}$ så kommer den att tolkas som speglingsfrekvensen f'_{signal} enligt

$$f'_{signal} = \begin{cases} r \cdot f_s & \text{om } r \leq 0,5 \\ f_s - r \cdot f_s & \text{om } r > 0,5 \end{cases} \quad r = \text{decimaldelen vid division mellan } f_{signal} \text{ och } f_s$$

Vi ser att resonemanget gäller även om signalfrekvensen är lägre än $\frac{f_s}{2}$, men då har vi det triviala fallet att signalfrekvens och speglingsfrekvens är samma sak. Signalens amplitud påverkas inte av vikningen. En alternativ tolkning av frekvensspeglingen finner du i [Bilaga 2.2](#).

2.2.3.2 Fasvinkel

Vi har sett att sinus- och cosinusfunktioner påverkar fassen olika vid vikning och måste nu fundera över vad detta kan bero på.

Vi vet från matematiken att cosinus är en jämn funktion och den brukar beskrivas som en funktion som saknar fasvridning medan sinusfunktionen är udda med fasvridning 90° , *Figur 2.24*.

Vi ser detta ur att en allmän signal kan beskrivas som

$$A \cdot \cos(\omega \cdot t) + j \cdot B \cdot \sin(\omega \cdot t)$$

dvs cosinusfunktionen är rent reell medan sinusfunktionen är rent imaginär (fasvridning 90°).

Vi har alltså sambandet

$$A \cdot \sin(\omega \cdot t) = A \cdot \cos(\omega \cdot t + 90^\circ)$$

Vi såg att då en sinussignal med frekvens i intervallet $f_s/2 - f_s$ samplades så bytte den vikta signalen tecken, dvs den fasvrids 180° . Lite tankearbete gör att vi kan inse att sinusfunktionens fasvridning på 90° helt enkelt byter tecken, den totala fasvridningen är ju då 180° . Cosinusfunktionen får också ombytt tecken på fassen men här har vi ju fasvridningen 0° så detta har ingen inverkan. De båda signaltyperna behandlas alltså på samma sätt men på grund av deras olika fasinnehåll så blir resultaten olika.

Skulle vår signal, med frekvens i intervallet $f_s/2 - f_s$ nu ha någon annan fasvridning

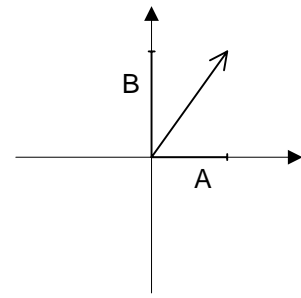
$$x(t) = A \cdot \cos(\omega \cdot t + \Phi)$$

så kommer den samplade signalen att vikas ner till en frekvens i intervallet $0 - f_s/2$ och denna signal kommer då att ha fasvridningen $-\Phi$.

Vad gäller signaler med frekvenser i intervallet $f_s - 3 \cdot f_s/2$ så såg vi ovan hur vi kunde göra tolkningen att signalen veks två gånger vilket betyder att vi får två teckenbyten på fassen och vi är tillbaka där vi började och den lågfrekventa signalen i intervallet $0 - f_s/2$ får samma tecken på fassen som den ursprungliga signalen hade.

2.2.3.3 Sammanfattning

Vi kan sammanfatta sambanden mellan signalernas belopp och fasvinkel enligt



Figur 2.24 Allmän signal

$$\left\{ \begin{array}{l} |f_{\text{vikt signal}}| = |f_{\text{ursprunglig signal}}| \\ \langle f_{\text{vikt signal}} \rangle = \begin{cases} \langle f_{\text{ursprunglig signal}} \rangle & \text{om } r \leq 0,5 \\ -\langle f_{\text{ursprunglig signal}} \rangle & \text{om } r > 0,5 \end{cases} \end{array} \right. \quad \begin{array}{l} r = \text{decimaldelen vid division} \\ \text{mellan } f_{\text{ursprunglig signal}} \text{ och } f_s \end{array}$$

Dessa erfarenheter kommer till nytta då vi i nästa kapitel studerar signaler i frekvensplanet.

En mer matematisk behandling av sampling och vikning finns i [Bilaga 2.3](#).

[Exempel Bilaga 2.4](#)

2.2.3.4 Antivikningsfilter

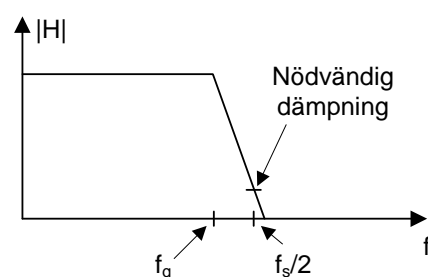
Samplingsvillkoret gör att vi måste se till att signaler med frekvenser högre än $f_s/2$ aldrig kan ta sig in i systemet, dvs de får inte nå S/H-kretsen. Det undviker vi genom att placera ett lågpasfilter före S/H-kretsen (se [Figur 2.1](#)). Filtret brukar kallas för ett *antivikningsfilter*. Lägga märke till att filtret ligger före A/D-omvandlaren, dvs där vi har analoga signaler och vi måste använda ett analogt filter.

Vi bör alltid ha med ett antivikningsfilter även om vi vet att den aktuella signalen inte innehåller frekvenskomponenter större än $f_s/2$ eftersom signalen trots detta alltid kan innehålla störningar, t ex brus, med högre frekvenser och dessa kommer då att vikas ner till samma frekvensområde som signalen har, dvs frekvenser lägre än $f_s/2$.

Signaler som har drabbats av vikningsdistorsion, dvs har samplats med för låg samplingsfrekvens, kan **ALDRIG återställas** eftersom det inte finns något sätt att avgöra om delsignaler härstammar från korrekt samplade signaler eller om de är vikta, falska komponenter och dessa två delar, sanna och falska delsignaler, kommer att adderas och sammanblandas.

Antivikningsfiltret måste då ha en gränshfrekvens

som är lägre än $f_s/2$ och eftersom ett filter (speciellt ett analogt filter) inte på något enkelt sätt kan ges en brant övergång mellan pass- och spärrband så måste vi tillåta ett visst övergångsband. Dessutom är det omöjligt att realisera ett filter med oändlig dämpning i spärrbandet vilket gör att viss vikning kommer att förekomma. Vi får bara se till att filtret är tillräckligt skarpt och därmed dämpar tillräckligt mycket i spärrbandet. Hur mycket som är tillräckligt avgörs av applikationen, [Figur 2.25](#).



Figur 2.25 Antivikningsfilter

Det finns möjligheter att via så kallad översampling använda ett enklare analogt filter och göra en del av filtreringen digitalt. Vi kommer inte att behandla detta i grundkursen.

Exempel Bilaga 2.5

2.2.4 Undersampling

Under speciella förutsättningar kan vi tillåta oss att använda en samplingsfrekvens som är lägre än signalfrekvensen, vi talar då om *undersampling*. Vilka är då dessa förutsättningar?

Vi har ovan sett att signaler som ligger i intervallet $k \cdot f_s$ till $k \cdot f_s + \frac{f_s}{2}$, där k är ett

heltal, kommer att speglas och tolkas på ett sätt medan signaler i intervallet $k \cdot f_s + \frac{f_s}{2}$ till $(k + 1) \cdot f_s$ kommer att tolkas på ett annat sätt. Vet vi nu att det bara finns signaler i *ett*

av dessa intervall och i *inga* andra intervall, inte ens i intervallet $0 - \frac{f_s}{2}$, så kan vi lätt

bestämma vilken frekvens den registrerade lågfrekventa, speglade signalen egentligen har och vi kan göra denna omtolkning i vår beräkning. För att vi skall vara säkra på att signalen bara innehåller frekvenskomponenter i önskat område så får vi ersätta antivikningsfiltrets lågpassfilter med ett bandpassfilter som bara släpper igenom önskat frekvensband.

Metoden är bara användbar då vi samplar signalen för att analysera den i den tidsdiskreta världen. Skulle vi återskapa den analoga signalen via D/A-omvandling så skulle vi ju inte få rätt frekvens utan vi skulle få den lågfrekventa speglingsfrekvensen.

Undersampling är vanlig i mätinstrument som används för att mäta på smalbandiga högfrekventa signaler, t ex radiosignaler runt en bärvåg.

2.3 Talrepresentation

Vi börjar nu närma oss övergången från analog till digital signal och före detta kommer vi att tala lite grann om hur tal kan representeras i digital form. Vi har här två huvudgrupper av tal nämligen fixtal och flyttal. I de flesta fall ger flyttal större talområde och noggrannare resultat men det kräver också mer avancerade processorer och som i så många andra fall så får vi väga prestanda mot pris.

Lägg märke till att de A/D- och D/A-omvandlare vi senare kommer att tala om använder sig av fixtalsrepresentation även om beräkningarna i DSP-systemet sker med flyttal.

2.3.1 Ordlängd

Oberoende av representation så arbetar vårt digitala system med någon systemberoende *ordlängd*, dvs varje värde representeras av ett visst antal bitar. Vanliga ordlängder är

- 8 bitar, vanligt i enklare mikroprocessorer
- 12 bitar, ganska vanligt i mätsammanhang
- 16 bitar, mer avancerade mikroprocessorer, musik på CD
- 20 bitar, används i mer noggranna A/D- och D/A-omvandlare
- 24 bitar, den största ordlängd man brukar hitta hos A/D- och D/A-omvandlare
- 32 bitar, ytterligare mer avancerade mikroprocessorer
- 64 bitar, ännu mer avancerade mikroprocessorer

Det är vanligt att digitala system använder fler bitar för sina interna beräkningar än vad som finns hos omvandlarna, detta för att behålla beräkningsnoggrannheten genom systemet. Detta förklarar varför tabellen anger 32 bitar som en använd ordlängd medan A/D- och D/A-omvandlarna som mest 'bara' ger 24 bitar.

2.3.2 Fixtal

För fixtal gäller att vi i systemet har låst var i talet decimalpunkten ligger eller det kanske är bättre att tala om *binalpunkt* eftersom vi här jobbar med binära tal. Vi har två vanliga typer.

2.3.2.1 Heltal

Här jobbar vi med heltal och använder vanlig binär räkning. Binalpunkten ligger alltså till höger om LSB (Least Significant Bit, minsta signifikanta bit). Vi har då signalintervallet 0 till $2^{\text{antal bitar}} - 1$ om vi bara arbetar med positiva tal medan vi har intervallet $-2^{\text{antal bitar} - 1}$ till $2^{\text{antal bitar} - 1} - 1$ om vi arbetar med både negativa och positiva tal. I det senare fallet beror gränserna lite av hur vi representerar negativa tal. Heltal är inte så vanliga i signalbehandlingssammanhang.

2.3.2.2 Fraktionella tal

Då vi arbetar med fraktionella tal kan vi bara representera tal med storlek mindre än ett. Använder vi bara positiva tal så ligger binalpunkten till vänster om MSB (Most Significant Bit, mest signifikanta bit) i det binära ordet och för ett fyra bitars ord har vi

$$b_3b_2b_1b_0 \sim b_3 \cdot 2^{-1} + b_2 \cdot 2^{-2} + b_1 \cdot 2^{-3} + b_0 \cdot 2^{-4}$$

Använder vi både positiva och negativa tal så har vi i stället för ett fyra bitars ord

$$b_3b_2b_1b_0 \sim \text{teckenbit} + b_2 \cdot 2^{-1} + b_1 \cdot 2^{-2} + b_0 \cdot 2^{-3}$$

Oftast använder vi dessutom 2-komplement för negativa tal, se nedan.

Att vi bara kan representera tal mindre än ett leder ofta till att uttryck måste *skalas* för att konstanter skall hamna i arbetsområdet.

Fraktionell representation är vanlig i signalprocessorer med fixtalsrepresentation.

2.3.3 Flyttal

Flyttal representeras av mantissa och exponent, dvs *mantissa* · *talbas*^{exponent}. Det gör att vi här kan representera ett mycket större talområde än vad vi kan göra med fixtal men det leder samtidigt till mer komplicerade beräkningsenheter (ALU, aritmetisk logisk enhet) än vad fixtal kräver.

2.3.4 Negativa tal

Det förekommer tre olika metoder för att representera negativa tal och dessa kan förekomma både vid fix- och flyttalsberäkningar.

2.3.4.1 Teckenbit

Här låter vi talets mest signifikanta bit ange talets tecken (oftast nolla för positivt tal och etta för negativt tal) medan resten av bitarna representerar amplituden. Grundidén är enkel men kan ibland komplicera våra beräkningsfunktioner.

2.3.4.2 Ett-komplement

Här får positiva tal representeras av sin ursprungliga binära form, MSB måste vara en nolla (0), medan negativa tal representeras genom att alla bitar i talet (som representerar talets amplitud) inverteras vilket gör att MSB är ett (1) för negativa tal. Metoden är inte speciellt vanlig.

2.3.4.3 Två-komplement

Metoden är snarlik ett-komplement med den skillnaden att efter att vi har inverterat det negativa talets bitar så adderar vi ett till LSB (least significant bit, minst signifikanta bit). Detta är den helt dominerande metoden att representera negativa tal och den har vissa beräkningsmässiga fördelar, bland annat kan vi subtrahera genom att addera det första talet och det andra talets tvåkomplement. Vi behöver alltså ingen subtraherare utan bara en adderare och en krets som skapar två-komplement. Vi går inte in mera på detta.

Exempel Bilaga 2.6

2.4 Analog/digitalomvandlare (A/D)

Vår samplade signal är nu tidsdiskret men den har fortfarande en analog amplitud efter S/H-kretsen, dvs kretsens kondensator kan anta vilken spänningsnivå som helst (inom systemets arbetsområde). För att överföra signalen till det digitala DSP-systemet måste vi nu även diskretisera dess amplitud, vi måste omvandla den till digital form. Detta sker i en A/D-omvandlare. Det finns ett antal olika typer av A/D-omvandlare, som integrerande omvandlare och omvandlare med succesiv approximation, vi kommer inte att gå närmare in på deras uppbyggnad men däremot diskutera en del av deras egenskaper.

2.4.1 A/D-omvandlarens grundfunktion

Som namnet säger skall en analog/digitalomvandlare göra omvandlingen mellan en analog och en digital signal. Detta betyder i praktiken att en analog spänning (A/D-omvandlarens insignalen är normalt sett en spänning även om den via en givare har omvandlats från någon annan storhet, t ex nivå, tryck etc) skall omvandlas till ett binärt ord. Det binära ordet består av ett begränsat antal bitar, n , vilket innebär att den digitala signalen har 2^n nivåer som skall representera olika spänningar hos insignalen. Lägg märke till att vi här arbetar

med fixtal och inte flyttal och att våra nivåer är relativt maximal spänning i arbetsområdet, inte relativt någon spänningsskala i volt.

Hur stor spänning motsvarar då varje nivå hos den digitala signalen? Vilken *upplösning* har omvandlaren? Detta beror på insignalens arbetsområde och på antalet bitar hos A/D-omvandlaren. Vi talar om två olika typer av A/D-omvandlare, *unipolära* och *bipolära* omvandlare.

2.4.1.1 Unipolära A/D-omvandlare

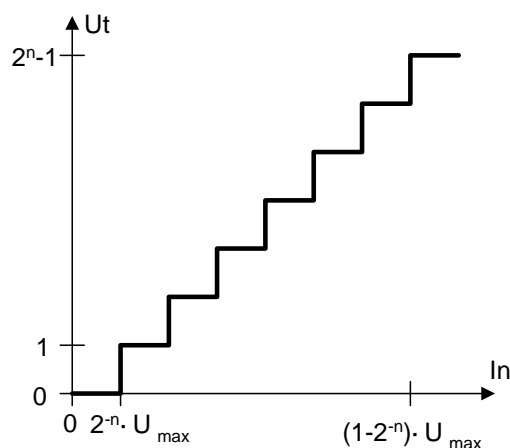
En unipolär omvandlare kan bara hantera positiva insignaler (tal utan tecken) och insignalen kan då finnas i området $0 - U_{\max}$. Omvandlaren får då en upplösning Δ enligt

$$\Delta = \frac{U_{\max}}{2^n}$$

och upplösningen är då den minsta spänningsskillnaden som vi kan representera. Lägg märke till att ett ord med bara nollor motsvarar spänningen 0 volt medan ett ord med bara ettor *inte* representerar spänningen U_{\max} utan spänningen $U_{\max} - \Delta$. Vi kan alltså inte representera

insignalens största värde, med fler bitar kommer vi dock närmare denna nivå (i volt) samtidigt som upplösningen blir bättre, *Figur 2.26*. Man bör undvika att säga större upplösning eftersom det kan vara svårt att veta vad man menar, fler steg eller större steg?

Unipolära omvandlare är vanliga i styr- och reglersammanhang då negativa signaler inte kan förekomma, t ex nivåer i tankar. De kan också förekomma i bildsammanhang där bildpunkternas färg- eller gråskala brukar beskrivs av positiva tal (ofta heltal).



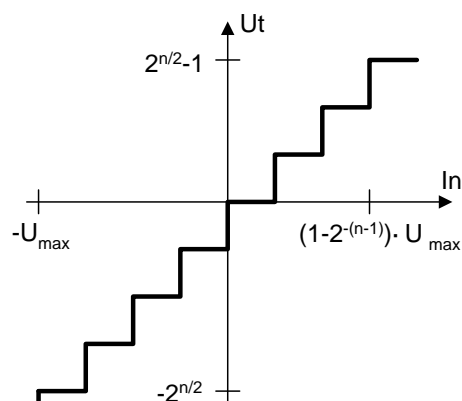
Figur 2.26 Unipolär A/D-omvandlare

2.4.1.2 Bipolära A/D-omvandlare

Bipolära omvandlare hanterar både positiva och negativa signaler, i de flesta fall lika stora områden på båda sidor om nollan, och vi har arbetsområdet $-U_{\max}$ till U_{\max} , dvs totalt $2 \cdot U_{\max}$, som ger upplösningen

$$\Delta = \frac{U_{\max} - (-U_{\max})}{2^n} = \frac{2 \cdot U_{\max}}{2^n}$$

Samma resonemang som för den unipolära omvandlaren gör att ett binärt ord med bara nollor representerar spänningen $-U_{\max}$ medan ett ord med bara ettor representerar spänningen



Figur 2.27 Bipolär A/D-omvandlare

$U_{\max} - \Delta$. Lagg märke till att den negativa nivån är större (i magnitud) än den positiva, [Figur 2.27](#).

Bipolära omvandlare används för olika typer av växelspanningar, t ex ljudsignaler.

Exempel Bilaga 2.7 - 8

2.4.1.3 Codec

I kommunikationssammanhang används ganska ofta omvandlare som inte har linjär omvandling från analoga till digitala signaler och tillbaka igen utan man använder en A/D-omvandlare med logaritmisk överföringsfunktion och en D/A-omvandlare med exponentiell överföringsfunktion. Metoden ger små upplösningsteg vid låga amplituder och större upplösningsteg vid stora amplituder. Denna fördelning är i en del fall bättre än den linjära fördelningen, till exempel då kommunikationen består av tal eftersom vi uppfattar ljud med logaritmiska skalor. Man säger att signalen är *komprimerad* (komprimerad/expanderad).

Filter, samplingskrets, A/D-omvandlare och D/A-omvandlare samt ett serieinterface brukar vara integrerade i en enda krets som kallas *codec* (coder/decoder).

Vi kommer inte att gå in på denna typ av A/D-, D/A-system.

2.4.1.4 Omvandlingsfel

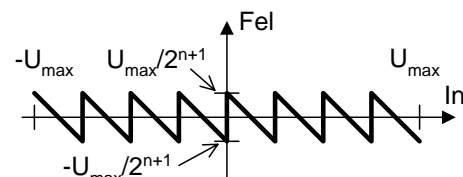
Eftersom vi omvandlar den analoga signalen till ett antal diskreta steg, vi kvantiserar den, så kommer det alltid att finnas analoga signaler som inte beskrivs korrekt eftersom de har nivåer som ligger mellan dessa steg, vi får alltså fel i vår beräkning.

Om vi avrundar värdena så kommer vi att få det största felet om signalnivån ligger precis mitt emellan två kvantiseringssteg, dvs vi får

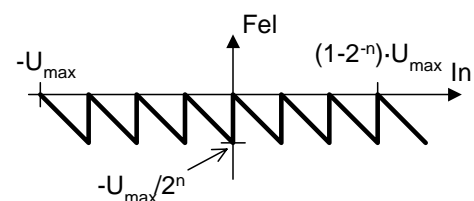
$$\text{Maximalt kvantiseringsfel} = \frac{\Delta}{2}$$

[Figur 2.28](#)

Trunkerar vi i stället värdet till närmast underliggande värde (som är vanligast) så blir det maximala felet lika med upplösningen Δ och vi får detta fel då signalnivån ligger strax under en nivå hos A/D-omvandlaren, [Figur 2.29](#).



[Figur 2.28](#) Omvandlingsfel hos avrundande A/D-omvandlare



[Figur 2.29](#) Omvandlingsfel hos trunkerande A/D-omvandlare

2.4.1.5 Kvantiseringsbrus

Beräkningsfelet brukar ofta kallas kvantiseringsbrus (quantisation noise) och vi brukar ange det via förhållandet mellan maximal signalstorlek och felets storlek. Förhållandet kallas *signal/kvantiseringsbrusförhållande* (eller slarvigt signal/brusförhållande, eller på engelska *Signal to Quantisation Noise Ratio, SQNR*).

För en unipolär omvandlare motsvaras den största signalen av 2^n bitar medan största felet är en halv bit (vid avrundande omvandlare)

$$SQNR_{unipolär} = \frac{U_{\max}}{\Delta/2} = \frac{U_{\max}}{\frac{U_{\max}/2^n}{2}} = \frac{2^n}{1/2} = 2^{n+1} \sim 20 \cdot \log(2^{n+1}) = 20 \cdot (n+1) \cdot \log(2) \approx 6 \cdot (n+1)$$

För bipolära omvandlare blir den största signalen det samma som den största positiva eller negativa amplituden och den motsvaras då av 2^{n-1} bitar och vi får

$$SQNR_{bipolär} = \frac{U_{\max}}{\Delta/2} = \frac{U_{\max}}{\frac{2 \cdot U_{\max}/2^n}{2}} = \frac{2^{n-1}}{1/2} = 2^n \sim 20 \cdot \log(2^n) = 20 \cdot n \cdot \log(2) \approx 6 \cdot n$$

n	4	8	12	16	20	24
$SQNR$	24,1	48,2	72,2	96,3	120,4	144,5

Tabell 2.1 Signal/brusförhållanden (SQNR) vid olika ordlängder för bipolär A/D-omvandlare

Lägg märke till att vi har beräknat SQNR vid maximal inamplitud. Har vi en lägre inamplitud så kommer felet fortfarande att vara lika stort medan signalamplituden sjunker vilket innebär att SQNR sjunker.

Exempel Bilaga 2.9

Ibland använder man i stället signal- och bruseffekt för att beräkna SQNR. För en bipolär omvandlare är bruseffekten

$$P_n = \frac{\Delta^2}{12}$$

och signaleffekten

$$P_s = \frac{U_{\max}^2}{2}$$

som ger

$$SQNR_{bipolär} = 1,5 \cdot 2^{2 \cdot n} \sim 6,02 \cdot n + 1,76 \text{ dB}$$

Bilaga 2.10

2.4.2 A/D-omvandlartyper

Det finns ett antal olika typer av A/D-omvandlare som vi inte skall gå närmare in på men vi skall i alla fall nämna deras grundfunktioner.

2.4.2.1 Integrerande A/D-omvandlare

Funktionen bygger i sin enklaste form (enkelramp) på att man räknar hur lång tid det tar för den analoga spänningen att ladda upp en kondensator. Högre spänning ger längre uppladdningstid och därmed större räknarvärde. Man kan få bättre resultat genom att räkna tiden för både en upp- och en urladdning (dubbelramp) och denna metod är vanligast.

Omvandlingen tar lång tid men har den förtjänsten att den via en integrerande funktion (upp- och urladdning) undertrycker störningar hos signalen. Den långa omvandlingstiden gör denna typ av omvandlare ovanlig i signalbehandlings-sammanhang. Typen är mycket vanlig i digitala multimetrar.

2.4.2.2 Uppräknande A/D-omvandlare

Här låter vi en räknare räkna upp insignalen till en D/A-omvandlare vars analoga utsignal jämförs med den analoga insignalen. Då de båda signalerna blir lika stora stoppas uppräknningen och vi har funnit det digitala värde som motsvarar insignalens amplitud. Metoden förekommer sällan eftersom uppräknningen gör att omvandlingstiden kan variera kraftigt (liten insignal ger få stegs uppräkning och därmed kort omvandlingstid medan stor insignal ger många stegs uppräkning och därmed lång omvandlingstid) vilket gör det svårt att använda den i samplade sammanhang där vi har en bestämd tid (samplingperioden) för omvandling och DSP-beräkning.

2.4.2.3 A/D-omvandling via successiv approximation

Funktionen är ganska lik den uppräknande omvandlaren men i stället för att styra D/A-omvandlaren från en räknare så använder vi logik som testar bit för bit, från MSB till LSB, och sätter bitarna så länge insignalen är större än D/A-omvandlarens utsignal. Här har vi god kontroll över omvandlingstiden, den kommer alltid att ta lika lång tid för varje bit och den totala omvandlingstiden blir direkt proportionell mot antalet bitar i det digitala ordet. Vi gör ju *en* jämförelse per bit.

Denna omvandlartyp är den klart dominerande, speciellt för omvandlare med bättre upplösning, dvs fler bitar och metoden kallas *successiv approximation*.

2.4.2.4 Flashomvandlare

I flashomvandlaren har vi satt upp en kedja av 2^{n-1} jämförare (komparatorer) med hjälp av vilka vi jämför insignalen och direkt får ut det digitala ordet via ett logiskt nät.

Metoden är den snabbaste av dem vi beskrivit men det är svårt och dyrt att bygga dessa omvandlare för många bitar, det leder ju till många komparatorer som kräver precision i uppbyggnad och intrimning av deras jämförelsenivåer. Detta leder till att flashomvandlare förkommer upp till 10-12 bitars ordlängd.

2.4.3 Omvandlarspecifikationer

Vi skall lite kort nämna vilka egenskaper hos en A/D-omvandlare som det kan vara lämpligt att känna till. Egenskaperna ger sig ofta till känna som olika typer av fel.

2.4.3.1 Omvandlingstid

Tiden anger hur lång tid en A/D-omvandling tar. Den totala omvandlingstiden beror dessutom av den tid som på engelska kallas settling time (se nedan), vi kan kalla den stabiliseringstid.

2.4.3.2 Settling time

Stabiliseringstiden anger den tid som går efter en omvandling innan utsignalen har stabiliserat sig så att utsignalen är korrekt.

Omvandlings- och stabiliseringstiden anger nu den totala omvandlingstiden och skall vi hinna med omvandlingarna så får inte samplingsperioden, T , vara kortare än denna tid. I praktiken kommer vi dessutom att behöva lite beräkningstid mellan varje samplingsfall vilket gör att samplingsperioden måste vara ytterligare längre, dvs maximal samplingsfrekvens sjunker.

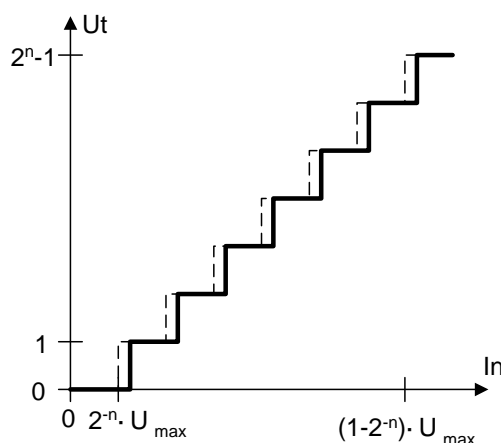
2.4.3.3 Noggrannhet

Noggrannheten anger hur trovärdigt det omvandlade värdet är och anges i upplösningssteg Δ , eller i LSB (som är samma sak) som det brukar stå i datablad. Vi kan t ex se noggrannheten $\pm 1,5$ LSB. ibland anges flera noggrannheter som beror på hur lång stabiliseringstid vi anger.

2.4.3.4 Offsetfel

Offset innebär att alla spänningar är lite förskjutna med samma (positiva eller negativa) värde U_{offset} . Vilket för en bipolär omvandlare innebär att den inte har arbetsområdet $-U_{\text{max}}$ till U_{max} utan i stället området

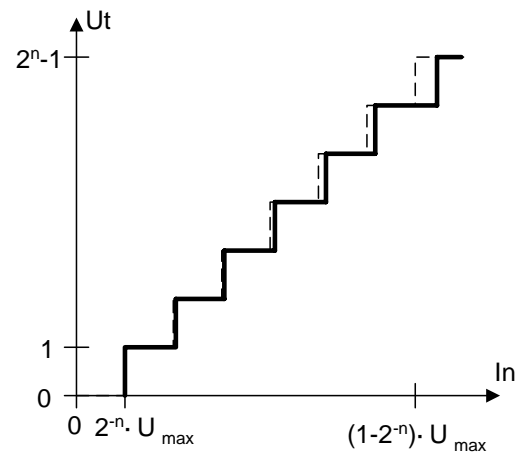
$-U_{\text{max}} + U_{\text{offset}}$ till $U_{\text{max}} + U_{\text{offset}}$, [Figur 2.30](#).



Figur 2.30 Offsetfel

2.4.3.5 Förstärkningsfel

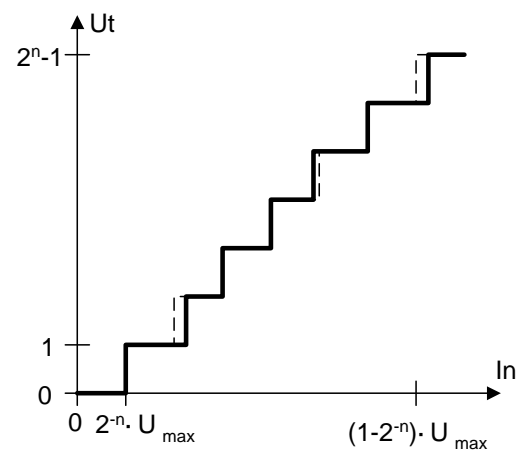
Med förstärkningsfel menar vi att upplösningen inte är den önskade upplösningen Δ utan lite större eller mindre än detta, [Figur 2.31](#).



Figur 2.31 Förstärkningsfel

2.4.3.6 Linjaritet

Med linjaritetsfel menar vi att de olika omvandlingsstegen Δ inte är lika stora, [Figur 2.32](#).



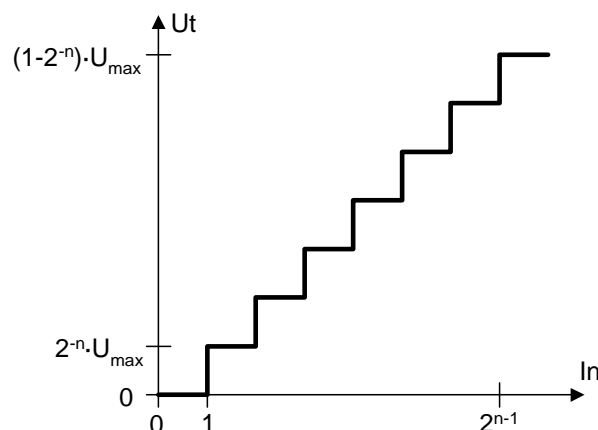
Figur 2.32 Linjaritetsfel

2.5 Digital/analogomvandlare (D/A)

D/A-omvandlaren skall omvandla det binära ordet till en analog spänning. Eftersom det binära ordet med n bitar bara kan ge 2^n olika nivåer så är utsignalen egentligen inte analog (amplitudkontinuerlig) utan den kan bara anta 2^n olika diskreta nivåer, *Figur 2.33*. Lägga märke till att vi även här använder fixtal.

De flesta D/A-omvandlare genererar utspänningen med hjälp av ett nät av resistanser som styrs av de digitala bitarna och av en operationsförstärkare. Omvandlingstiden är normalt kort men det kan ta en viss tid innan utsignalen har stabiliserat sig.

Omvandlingsprocessen beskrivs av samma data och fel som gällde för A/D-omvandlaren.



Figur 2.33 D/A-omvandlad utsignal

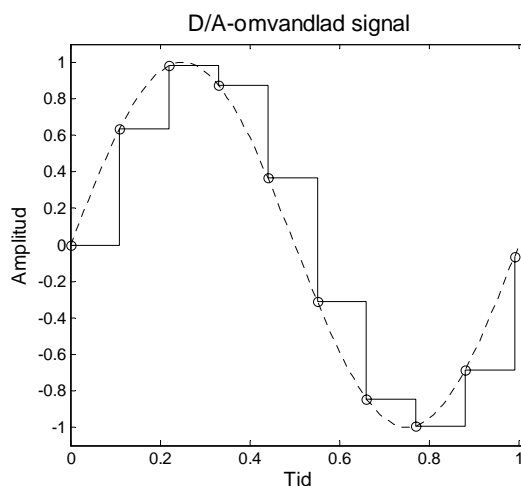
2.6 Utjämningsfilter

Vi nämnde ovan att utsignalen från D/A-omvandlaren bara kan anta 2^n diskreta nivåer och vi ser i *Figur 2.34* att vi får en kantig utsignal med 'trappsteg'. Detta är oftast inte acceptabelt utan vi vill ha en signal med mjuka övergångar och inte dessa abrupta steg. Vi inser att diskontinuiteterna kommer att ha grundfrekvensen lika med samplingsfrekvensen f_s eftersom de kommer en gång per samplingsperiod. Samtidigt gäller, om vi uppfyller samplingsvillkoret, att signalen bara kan innehålla signalfrekvenser upp till (strax under) halva samplingsfrekvensen och vi kan filtrera bort diskontinuiteterna med ett lågpasfilter med en gränshfrekvens strax under $f_s/2$. Vi kallar filtret för ett *utjämningsfilter*

(smoothing filter) Lägga märke till att filtret måste komma efter D/A-omvandlaren när vi har en analog signal, om än med diskreta nivåer, så vi måste använda ett analogt filter (*Figur 2.1*).

Vi ser att vi har ungefär samma krav som på antivikningsfiltret före A/D-omvandlaren och kan utnyttja samma dimensionering även om det i praktiken måste bli två separata filter om in- och utmatning skall ske i realtid.

Även här kan vi utnyttja översampling för att flytta över en del av filtreringen till den digitala världen och därmed få enklare analoga filter. Vi går inte in på detta.



Figur 2.34 D/A-omvandlad signal

2.7 Olika former av signaler

Ovanstående beskrivningar av olika delar av vårt system gör att vi kan särskilja fyra olika varianter av signaler i vårt system

- amplitud- och tidskontinuerlig (före S/H-kretsen)
- amplitudkontinuerlig och tidsdiskret (före A/D-omvandlaren men efter S/H-kretsen)
- amplitud- och tidsdiskret (i DSP-systemet)
- amplituddiskret och tidskontinuerlig (efter D/A-omvandlaren men före utjämningsfiltret)

När vi talar om en analog signal menar vi normalt en amplitud- och tidskontinuerlig signal medan vi med en digital signal menar en amplitud- och tidsdiskret signal.

I fortsättningen kommer vi nästan uteslutande att ägna oss åt samplade, dvs tidsdiskreta signaler och system. Vi kommer bara i förbigående att nämna effekter som beror på att våra signaler och system är amplituddiskreta i våra dator- eller hårdvarusystem och resonemangen kommer att ske som om vi hade amplitudkontinuerliga signaler och system, dvs som om vi hade obegränsad noggrannhet i våra beräkningar.

2.8 Typiska data för digitala system

I [Tabell 2.2](#) nedan anger vi data för ett antal olika förhoppningsvis välkända system. Exemplet har en tonvikt på ljudsystem.

Applikation	Samplingsfrekvens	Ordlängd
Telefoni/GSM	8 kHz	8 bitar
CD	44,1 kHz	16 bitar
Digital bandspelare, DAT	48 kHz	16 bitar
MiniDisc	44,1 kHz	16 bitar
Enklare ljudkort	44,1/48 kHz	16/20 bitar
Bättre ljudkort	88,2/96 kHz	20/24 bitar
Avancerad studio	96/192 kHz	24 bitar
Digital radio, DAB	48 kHz	18 bitar

Tabell 2.2 Data för typiska digitala system

Exemplen anger de samplingshastigheter och ordlängder som används vid själva omvandlingen. I några fall (GSM, MiniDisc och DAB) används även olika former av komprimering för att minska antalet översända bitar.