

Name:	
ID number:	LiU-ID:
Passed:	Date:

TSBB16 Datorövning A

Samplade signaler

Faltning

Utvecklad av Klas Nordberg

Computer Vision Laboratory, Linköping University, Sweden

24 augusti 2015

Introduktion

Denna övning bygger på de kapitel i kompendiet som hör till föreläsning 1-3 samt lektion 1-2. Den tar upp grundläggande metoder som används för att bearbeta signaler i form av ljud och bild. Ljud som du hör är funktioner av en kontinuerlig tidsvariabel t , men de signaler som en dator (eller digitala system generellt) kan bearbeta är funktioner av en diskret tidsvariabel k . Överföringen av en tidskontinuerlig (samplad) signal till en tidsdiskret signal kallas *sampling* och den motsatta processen kallas *rekonstruktion*.

I den första delen av övningen kommer du att titta både på sampling och rekonstruktion, samt omsampling vilket överför en samplad signal till en annan tidsdiskret signal där samplingen av tidsvariabeln är gjort på ett annat sätt.

I den andra delen undersöks hur signaler kan bearbetas med filter, som förstärker olika frekvenskomponenter på olika sätt.

Bland uppgifterna finns "mästarprov" uppgifter som (eventuellt) ser svårare ut än de är. De kräver att du har förberedd övningen ordentligt och bygger i mycket på att du har gjort och förstått uppgifterna innan.

Extra

Avsnitt som är markerade med "Extra" kan initialt hoppas över och slutföras när de andra övningarna är avslutade om det finns tid kvar.

1 Förberedelseuppgifter

Innan du kommer till datorövningen är det viktigt att du har en klar uppfattning om det tillhörande kursmaterialet.

Nedan finns ett antal förberedelseuppgifter som ska besvaras innan du deltar i datorövningen. Förutom föreläsningar och lektioner behöver du läsa och sätta dig in i de olika övningarna i denna guide för att kunna besvara en del av uppgifterna.

1.1 Samplade signaler

1. Vad är det för fysikalisk storhet som en ljudsignal representerar? Ledning: se kompendiet kapitel 1.

SVAR:

2. Vilket frekvensintervall brukar normalt anges som hörbart för det mänskliga örat? Ledning: kolla Wikipedia.

SVAR:

3. Hur många sampel period måste en tidsdiskret signal ha enligt samplingsteoremet för att väl kunna beskriva en tidskontinuerlig cosinus-signal? Ledning: se kompendiet kapitel 4 och kapitel 6.

SVAR:

4. Om en signal samplas med för låg samplingsfrekvens kan det uppstå vinkningsdistorsion när den rekonstrueras. Vad innebär det? Ledning: se kapitel 6.

SVAR:

5. Med vilken samplingsfrekvens och antal bitar lagras ljud på CD-skivor (per ljudkanal)? Vad innebär det i bitar/sekund som måste läsas från CD:n för att kunna spela upp ljudet? Ledning: kolla Wikipedia.

SVAR:

6. Hur höga frekvenser kan då representeras i musiken utan att vinkningsdistorsion uppstår?

SVAR:

7. Hur många kvantiseringsnivåer har en signal som representeras med b bitar? Hår många nivåer blir det för 16 bitar?

SVAR:

8. En digital bild kan ses som en matris med pixlar, bildelement. Detta är en diskret funktion, men inte av en diskret tidsvariabel k . Hur många och vad för typ av variabler handlar det om i detta fall?

SVAR:

9. Vad innebär kvantiseringsbrus? Ledning: se kompendiet kapitel 6.

SVAR:

10. Hur borde kvantiseringsbrusets standardavvikelse σ beror av antalet bitar, b , som används för varje sampel?

SVAR:

11. Hur beräknas standardavvikelsen σ mellan en ideal diskret signal med n sampel, betecknad s , och en approximation av samma signal, betecknad s_{app} ?

SVAR:

12. Vad är fördelen med närmsta-granne-interpolation jämförd med bilinjär interpolation?

SVAR:

1.2 Faltning och filter

13. En tidsdiskret signal faltas med ett filter som har 2 koefficienter = $[0.5 \ 0.5]$. Vad blir amplitudkaraktistiken $|H(f)|$ för detta system? Rita en kurva på $|H(f)|$ för f upp till 22050 kHz när $T = 1/44100$ (dvs. 44100 sampel per sekund). Vilken typ av filter är detta? Ledning: se räkneuppgift 4.8.

SVAR:

14. Varför spelar det ingen roll hur amplitudkaraktistiken ser ut över 22,05 kHz i detta fall?

SVAR:

15. Ett tidsdiskret system faltas med ett filter som har 2 koefficienter = $[0.5 \ -0.5]$. Vad blir amplitudkaraktistiken $|H(f)|$ för detta system? Rita en kurva på $|H(f)|$. Vilken typ av filter är detta? Ledning: se räkneuppgift 4.8.

SVAR:

16. Denna typ av filter kan approximativt ses som ett deriverare. Varför? Hur syns det av kurvan för $|H(f)|$? Ledning: se kompendiet avsnitt 4.3.7.

SVAR:

17. Vad blir amplitudkarakteristiken $|H(f)|$ för ett filter med 4 koefficienter $[a \ b \ c \ d]$?

SVAR:

18. Vad är filtrets förstärkning vid frekvenserna $f = 0$ och $f = \text{Nyquist-frekvensen}$? Hur kan koefficienterna c och d uttryckas i a och b för ett LP-filter, dvs. om $H(0) = 1$ och $H(\frac{f_s}{2}) = 0$? Vad blir då amplitudkarakteristiken uttryckt i de två återstående koefficienterna a, b ?

SVAR:

19. Vilken frekvensfunktion $H(\omega)$ har ett system med impulssvaret

$$h(t) = \sqrt{\frac{2}{\pi\sigma^2}} e^{-\frac{1}{2}\frac{t^2}{\sigma^2}} \cos(\omega_0 t)?$$

Skissa både $h(t)$ och funktionen $|H(\omega)|$. Ledning: se kompendiet avsnitt B.8.

SVAR:

20. I det generella fallet, vilken typ av filter motsvarar detta system? Centerfrekvens? Bandbredd?

SVAR:

21. Vilken typ av filter motsvarar detta system om $\omega_0 = 0$ och det får en extra faktor $\frac{1}{2}$? Gränsfrekvens? Skissa frekvensfunktionen.

SVAR:

22. Hur skulle du kunna använda ovanstående information för att åstadkomma ett bandspärrfilter eller ett högpasfilter? Ledning: se kompendiet avsnitt 5.1.1.

SVAR:

23. Hur skulle du välja samplingsfrekvens f_s eller samplingsperiod $T = 1/f_s$ för ett tidsdiskret system som ska implementera frekvensfunktionen som beskrivs i uppgift 19?

SVAR:

2 Matlab-funktioner

En lista med Matlab-funktioner som används i laborationen:

- `zeros`: Skapar en vektor eller matris med alla element = 0.
- `ones`: Skapar en vektor eller matris med alla element = 1.
- `figure`: Skapa ett nytt fönster att rita figurer i.
- `plot`: Rita upp en signal eller funktion. Använd zoom-funktionen (+) för att titta närmare på detaljer i signalen.
- `subplot`: Delar upp ett fönster i flera underfönster som var och en kan användas i `plot`-funktionen.
- `hold`: Lås uppritad figur i ett fönster så att ny ritning kan ske "ovanpå", utan att radera.
- `title`: Skriv ut en rubrik på en figur.
- `sound`: Spela upp ljud i högtalaren.
- `conv`: Faltning mellan två funktioner.
- `pause`: Väntar på att en tangent trycks ned innan programmet fortsätter.
- `sprintf`: Formatera textsträngar för utskrift.
- `axis`: Styr mer i detalj hur en kurva eller bild ritas ut
- `min` och `max`: Beräknar min eller max av värden.

Om du känner dig osäker på hur dessa funktioner används, titta i Matlabs dokumentation av dessa funktioner innan du gör laborationen.

Använd möjligheten att placera den Matlab-kod som förekommer i denna laboration i en skriptfil (.m fil) som du kan editera och köra valda delar av i Matlab.

Del I

Samplade signaler

i denna del av laborationen ska du undersöka olika fenomen som uppstår i samband med att tidskontinuerliga signaler samplas, eller samplas om. Signalerna kan vara antingen syntetiska, skapade i datorn, eller ljudsignaler.

Du kommer att spela upp signaler i form av ljud. För att inte störa andra laboranter, koppla in ett headset datorns högtalarutgång. Använd ett förgreningsuttag om ni är två studenter som jobbar tillsammans. Eventuellt kan datorns ljudvolym behöva justeras så att ljudstyrkan blir lagom.

3 Stationära signaler

Till att börja med ska du undersöka syntetiska signaler, sinussignaler, som du skapar själv i Matlab. Matlab-koden som används i denna del av övningen finns i en skript-fil som heter `stat.m`.

Börja med att skapa en stationär signal, en cosinus-signal med frekvensen 1000 Hz, som du sedan kan spela upp. Det blir en tidsdiskret signal, som är samplad med 44.100 Hz och som är 2 sekunder lång.

```
f = 1000; % Signalens frekvens
omega = 2*pi*f; % Signalens vinkelfrekvens
fs = 44100; % Sampelfrekvens
Ts = 1/fs; % Sampelintervall
k = 0:1:(2*fs); % Diskret tidsvariabel över 2 sek
s1 = cos(omega*k*Ts); % Signalen skapas
figure(1);plot(s1,'o-'); % Signalen ritas upp
```

Använd Matlabs zoom-verktyg för att titta på ett kortare intervall av signalen, så att du ser 3-4 perioder av signalen.

FRÅGA: Hur många sampel per period har denna signal? Hur bra stämmer det överens med samplingsteoremet?

SVAR:

Spela upp signalen i datorns högtalare. Observera att du måste ange samplingsfrekvensen f_s för att datorn ska kunna spela upp ljudet på korrekt sätt.

```
sound(s1, fs);
```

Justera högtalarens volymkontroll så att du kan höra ljudet bra (utan att störa dina grannar). Om det inte hörs någonting, kontroller högtalarens anslutningskablar och strömförsörjning. Åtgärda eventuella problem innan du fortsätter.

Skapa ytterligare en signal som har frekvensen 1000 Hz men som en sinus-funktion istället för cosinus, och en cosinus på 2000 Hz och spela upp dem.

```
s2=sin(omega*k*Ts);  
s3=cos(2*omega*k*Ts);  
sound(s1, fs); sound(s2, fs); sound(s3, fs);
```

Rita upp en signal som består av summan av s_1 och s_3 respektive summa av s_2 och s_3

```
figure(1); subplot(3,1,1); plot(s1+s3, 'o-');  
figure(1); subplot(3,1,2); plot(s2+s3, 'o-');
```

Lyssna på dessa två signaler

```
sound(s1+s3, fs); sound(s2+s3, fs); sound(s3, fs);
```

FRÅGA: Hörs det någon skillnad mellan dessa två signaler? Kan du förklara varför?

SVAR:

Rita även en signal som är summan av alla tre: s_1 , s_2 och s_3 och spela upp den tillsammans med en av de övriga:

```
figure(1); subplot(3,1,3); plot(s1+s2+s3, 'o-');  
sound(s1+s3, fs); sound(s1+s2+s3, fs);
```

FRÅGA: Hörs det någon skillnad mellan det första och andra ljudet? Kan du förklara varför?

SVAR:

3.1 Vikningsdistortion

Enligt samplingsteoremet måste en stationär cosinus-signal samplas med minst 2 sampel per period, annars uppstår vinkningsdistorsion. Du ska nu undersöka hur en cosinus-signal ser ut och låter vid olika frekvenser både under och över den som tillåts av samplingsteoremet. För att dina öron, datorns ljudkort och högtalare ska kunna hantera ljudet sänker vi tillfälligt samplingsfrekvensen till 10 kHz

```
fs2=10000;  
Ts2=1/fs2;  
k=0:1:(2*fs2);  
for frek=1000:1000:10000,  
    signal=cos(2*pi*frek*k*Ts2);  
    figure(2);plot(signal,'o-');  
    title(sprintf('Signalens frekvens = %d Hz', frek));  
    sound(signal, fs2);  
    pause;  
end,
```

Använd zoom-funktionen för att titta närmare på en kortare del av signalen, och tryck sedan på en tangent för att gå till nästa ljud med en högre frekvens.

FRÅGA: Vad är det du ser och hör? Är det i överensstämmelse med samplings-teoremet?

SVAR:

4 Naturliga ljud

Matlab-koden för denna del av övningen finns i skript-filen `natural.m`.

Ett antal signaler som består av ord uttalade av olika personer finns lagrade i filen `Ljud.mat`. Ladda in filen i Matlab, rita upp signalerna och spela upp ljuden.

```
load Ljud
figure(3);
for p=1:10,
    subplot(10,1,p);plot(ljud1{p});
    sound(ljud1{p},fs);
    pause(1);
end,
```

Ljudklippen är inspelade med CD-kvalitet, dvs 16 bitar per sampel och 44100 sampel per sekund. Förstora fönstret som signalerna ritades upp i för att se dom tydligt. Du kan även använda zoom-verktyget för att titta på detaljer i signalerna.

4.1 Kvantisering

Ändra kvantiseringen genom att “kasta bort” (eller sätta =0) de minst signifikanta bitarna i varje sampel, så att de b mest signifikanta bitarna återstår. Exempelvis, för $b = 15$ bitar:

```
b=15;
figure(4);
for p=1:10,
    kv_ljud=liu_quantize(ljud1{p},1.0,b);
    subplot(10,1,p);plot(kv_ljud);
    sound(kv_ljud,fs);
    pause(1);
end,
```

Prova att minska antalet bitar per sampel, b , och gör en bedömning hur många bitar som behövs för att inte kvantiseringsbruset ska bli besvärande.

FRÅGA: Hur många bitar behövs?

SVAR:

Prova att gå ända ned till bara 1 bit per sampel (inne i for-loopen). Ändra även intervallet som kvantiseringen sker inom till $\pm 0,5$ så att ljudets variationer kommer med bättre:

```
kv_ljud=liu_quantize(ljud1{p}, 0.5, 1);
```

FRÅGA: Kan du fortfarande urskilja vad som sägs?
--

SVAR:

Tag ett av ljuden, exempelvis det sista, och beräkna hur standardavvikelsen σ av kvantiseringsbruset beror av antalet bitar b som används. Plotta σ som funktion av b , titta även på funktionen i en logaritmisk skala.

```
forb=1:16
    kv_ljud=floor(ljud1{10}*2^(b-1))*2^(1-b);
    sigma(b)=norm(ljud1{10}-kv_ljud);
end,
figure(5);subplot(2,1,1);plot(sigma);
subplot(2,1,2);plot(log(sigma));
```

FRÅGA: Hur skulle du beskriva sambandet mellan kvantiseringsbruset och antalet bitar? Är detta det förväntade? (Se förberedelseuppgift 10)
--

SVAR:

Kontrollera om sambandet stämmer även för några andra ljudsignaler.

4.2 Nedsampling

Vid nedsampling reduceras antalet sampel per tidsenhet genom att endast vart n -te sampel i den tidsdiskreta signalen behålls medan de övriga kastas bort. Det betyder att den nedsamlade signalen har n gånger färre sampel än den ursprungliga signalen, dvs. dess samplingsfrekvens är f_s/n , där f_s är samplingsfrekvensen för den ursprungliga tidsdiskreta signalen. Om samma antal bitar per sampel bibehålls kommer nedsamplingen att producera en signal som kräver n gånger färre bitar per tidsenhet vid lagring eller överföring.

Vid behov kan de sampel som kastats bort approximeras på olika sätt, exempelvis

interpolation. Två olika metoder för interpolation ska undersökas:

- **Närmsta-granne-interpolation:** de sampel som ska rekonstrueras sätts till värdet av det närmsta kända sampel från den nedsamlade signalen.
- **Linjär interpolation:** det sampel som ska rekonstrueras ska ligga på en rät linje genom de två närmsta kända samplen (till höger respektive vänster).

Välj nedsamlingsfaktorn n som en jämn 2-potens: $n = 2^p$ där p är ett heltal. Exempelvis, en nedsampling med faktor 4 och rekonstruktion med närmsta-granne-interpolation:

```
n=2^2;
rekfilt=ones(1,n);
figure(7);plot(rekfilt,'Linewidth',2);
figure(6);
for p=1:10,
    signal=ljud1{p};
    nedsampl=zeros(1,length(signal));
    nedsampl(1:n:end)=signal(1:n:end);
    rekon=conv(nedsampl,rekfilt);
    subplot(10,1,p);plot(rekon,'o-');
    sound(rekon,fs);
end,
```

FRÅGA: Hur stor nedsamlingsfaktor (jämn 2-potens) kan användas innan det blir ett besvärande störning av ljudet?

SVAR:

Gör samma undersökning med linjär interpolation:

```
n=2^2;
rekfilt=(n-abs(-n:(n-1)))/n;
...
```

FRÅGA: Blir det någon skillnad jämfört med närmsta-granne-interpolation?

SVAR:

Undersök hur σ , standardavvikelsen av skillnaden mellan den ursprungliga signalen och den rekonstruerade signalen, beror av nedslingsfaktorn n .

```
signal=ljud1{10};
sigma=[];
for p=1:10,
    n=2^p;
    rekfilt=(n-abs(-n:(n-1)))/n;
    nedsampl=zeros(1,length(signal));
    nedsampl(1:n:end)=signal(1:n:end);
    rekon=conv(nedsampl,rekfilt,'same');
    sigma(p)=norm(signal-rekon);
end,
figure(7);subplot(2,1,1);plot(sigma,'o-');
subplot(2,1,2);plot(log(sigma),'o-');
```

FRÅGA: Hur kan du på enklast sätt beskriva sambandet mellan σ och n ?

SVAR:

4.3 Linearitet och distorsion

Du ska nu undersöka ett tidsdiskret system som inte är linjärt. Systemet tar helt enkelt varje sampel och avbildar det med en olinjär funktion f , vilket inför en slags distorsion på signalen.

$$f(x) = \text{sign}(x) \cdot \|x\|^\gamma$$

Här är γ en parameter som kontrollerar distorsionen. Notera att för $\gamma = 1$ är $f(x) = x$.

Plotta distorsionsfunktionen f

```

gamma=1.0;
x=-1:0.01:1;
distf=sign(x).*(abs(x).^gamma);
figure(6);plot(x,distf);

```

och lyssna på hur den påverkar de olika ljudsignalerna

```

for p=1:10,
    s=ljudl{p};
    dist_s=sign(s).*(abs(s).^gamma);
    subplot(10,1,p);plot(dist_s);
    sound(dist_s,fs);
    pause(1);
end,

```

Prova olika värden på γ .

FRÅGA: Inom vilket intervall kring 1 kan γ variera utan det uppenbart hörs att ljudet är förvrängt?

SVAR:

FRÅGA: På vilket sätt motiverar detta resultat att det för en del system är viktigt just att de är linjära och inte olinjära?

SVAR:

Extra

Prova att utsätta cosinus-signalen s_1 , som skapades i första övningen, för distorsionsfunktionen f .

FRÅGA: Hur upplever du att signalens tonhöjd ändras för olika värden på γ ?

SVAR:

4.4 Mästarprov: Bilder

Du ska nu göra samma typ av undersökning på bilder. Bilder i gråskala (ibland kallade “svartvita bilder”) brukar ofta digitaliseras med 8 bitar per pixel, vilket ger en god kvalitet. Matlab-koden för denna övning ligger i skript-filen `bilder.m`.

Börja med att läsa in en bild från fil och visa upp den i en figur.

```
im=double(imread('cameraman.png'));  
figure(10);colormap(gray);  
imagesc(im);
```

Kvantisera varje pixel den med b bitar och visa upp resultatet. Prova med olika värden på b .

```
b=8;  
imq=liu_quantize(im,256,b,'unipolar');  
figure(11);colormap(gray);  
imagesc(imq);
```

FRÅGA: Hur många bitar behövs det för varje pixel för att det inte ska börja se ut som att bildens intensitet är förvrängd?
--

SVAR:

Del II

Faltning

5 Små filter

Du kommer nu att undersöka hur operationen faltning påverkar signaler. Matlab-koden för denna del av övningen finns i en skript-fil som heter `filter.m`

Börja med att sätta upp lite parametrar och generera en syntetisk signal s_1 , med frekvensen f :

```
f = 10000; % Signalens frekvens
fs = 44100; % Sampelfrekvens
Ts = 1/fs; % Sampelintervall
k = 0:1:(0.01*fs); % Diskret tidsvariabel över 0.01 sek
s1 = cos(2*pi*f*k*Ts); % Signalen skapas
```

Definiera ett filter med 2 koefficienter = $[0,5 \ 0,5]$, och rita upp dess amplitudkaraktistik.

```
frek=0:22050;
filt=[0.5 0.5];
H=abs(cos(pi*frek*Ts));
figure(12);plot(frek,H);
```

Rita upp signalen, falta signalen med filtret och rita upp utsignalen s_2 från filtret:

```
figure(11);
subplot(2,1,1);plot(s1,'o-');
s2 = conv(s1,filt,'same');
subplot(2,1,2);plot(s2,'o-');
```

Prova att ändra signalen frekvens f , mellan 0 och 22050 Hz.

FRÅGA: Stämmer den uppritade amplitudkaraktistiken med de amplituder du observerar för filtrets utsignal s_2 vid olika frekvenser f ?

SVAR:

Byt filter till ett med koefficienter $= [0, 5 - 0, 5]$, bestäm dess amplitudkaraktistik, och gör om samma undersökning:

```
filt=[0.5 -0.5];
```

H=...

FRÅGA: Stämmer den uppritade amplitudkaraktistiken med de amplituder du observerar för filtrets utsignal s_2 vid olika frekvenser f ?

SVAR:

Kör om skriptet igen, med samma filter som senast och med $f = 5000$ Hz. Detta filter är approximativt ett deriverande filter.

FRÅGA: Hur syns det att filtret har en deriverande funktion på signalen? För vilka frekvenser kommer approximationen att vara bra?

SVAR:

6 Lite större filter

Mästarprov: bestäm värden för koefficienterna i ett FIR-filter av längd 4 så att det blir ett LP-filter med ett smalare passband än vad som går att åstadkomma med ett filter av längd $n = 2$. Använd resultaten från förberedelseuppgifterna 17 och 18.

FRÅGA: Vilka 4 koefficienter har du valt för filtret?

SVAR:

7 Faltning för detektion

Extra

Mästarprov **Bygger på avsnitt 6.5 och avsnitt B.8 i kompendiet.** Skapa två olika vektorer \mathbf{p}_1 och \mathbf{p}_2 med var sin puls i form av impulssvaren som hör till den frekvensfunktion $H(\omega)$ som beskrivs i förberedelseuppgift 19, med samma σ men olika centerfrekvens ω_0 . Gör det genom att sampla motsvarande impulssvar enligt beskrivningen kompendiet. Använd längden $n = 50$ för varje puls, och välj σ så att pulsen “får plats” inom den angivna längden.

Visa att en signal $x[k]$ som består av de två pulserna, med några föregående, mellanliggande och efterföljande sampel satta $= 0$, ger en tydlig detektion av de två pulserna när $x[k]$ är insignal till två matchande filter, ett för respektive puls. Tydlig detektion innebär: varje filter ger en signifikant utsignal för den “egna” pulsen och ≈ 0 för den andra pulsen.

FRÅGA: Hur valdes de matchande filterna?

SVAR:

FRÅGA: Hur valde du σ ? Hur valdes de två centerfrekvenserna?

SVAR: