Laboration i Fourieranalys, TMA131, vt2000

Signalanalys med snabb Fouriertransform

Den här laborationen har två syften: dels att visa lite på hur den snabba Fouriertransformen fungerar, och lite om vad man bör tänka på när den används; dels att verkligen testa den på ett par exempel.

Laborationen består av flera delar. Naturligtvis är det bäst att göra dem alla, men att göra en del av dem är bättre att inte göra någon.

De olika delarna är

1. Att analysera en ren sinussignal med olika samplingsfrekvens, och olika antal sampel.
2. Att studera överföringsfunktionen för ett linjärt filter, och sedan se hur "brus" passerar filtret
3. Att studera effekter av ickelinjäriteter i filter.

Allmänt om diskret och snabb Fouriertransform

(Detta är en **mycket** kortfattad beskrivning; läs också i boken)
Låt $f(t) : \mathbb{R} \to \mathbb{R}$ vara en signal, och låt

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}\, dt \tag{1}$$

vara dess Fouriertransform. Vi skall först anta att f är *bandbegränsad*, d.v.s. att det finns en konstant Ω så att $\hat{f}(\omega) = 0$ om $|\omega| \geq \Omega$. Samplingssatsen säger då att om man känner $f(t_n)$ i punkterna $t_n = n\frac{\pi}{\Omega}$, d.v.s. i praktiken måste man ha minst två sampel per period för den högsta tillåtna frekvensen. Om det dessutom är så att man bara är intresserad av $f(t)$ i ett begränsat intervall, säg $0 \leq t \leq T$, räcker det alltså att betrakta $N = T\Omega/\pi$ sampelpunkter (antalet måste naturligtvis anpassas så att det blir ett heltal). Om vi med detta menar att funktionen är 0 utanför detta intervall, måste antagandet om bandbegränsning vara felaktigt: ingen funktion som är noll utanför en begränsat intervall kan vara bandbegränsad; men under lämpliga förutsättningar blir inte felet så stort.

Nu antar vi att $f(t) = 0$ utanför intervallet $[0, T]$. Då borde samplingssatsen innebära att det skulle räcka att veta $\hat{f}(\omega_n)$ för $\omega_n = n\frac{2\pi}{T}$, för alla $|\omega_n| \leq \Omega$.
För att snabbt komma fram till den diskreta Fouriertransformen väjer vi N och sätter

$$a_n = f(nL/N) \qquad\qquad \hat{a}_m = \sum_{n=0}^{n=N-1} a_n e^{-i2\pi mn/N} \tag{2}$$

och hoppas, som Folland, att, åtminstone för $|m| << N$ skall $\hat{a}_m$ vara en bra approximation av $\hat{f}(\omega_m)$. Man ser i alla händelser enkelt att $\hat{a}_m$ är periodisk med period N och därför räcker det att betrakta $\hat{a}_m$, $0 \leq m \leq N-1$.

Och precis som for Fouriertransformen finns det en formel för att beräkna a_m om $\hat{a}_m$ är kända:

$$a_n = \frac{1}{N}\sum_{m=0}^{N-1} e^{i2\pi mn/N} \tag{3}$$

Problemet är nu att för man skall få någon noggrannhet i beräkningarna krävs det att N är stort, och då blir det kostsamt att utföra beräkningarna: det är N koefficienter att beräkna, och för varje n skall en summa med N termer beräknas, i allt N^2 operationer. Men det finns ett sätt att göra beräkningarna mycket snabbare, det som kallas snabb Fouriertransform (FFT).

Antag först att N är delbart med 2, $N = 2N_1$. Då kan man skriva

$$\hat{a}_m = \sum_{n=0}^{N_1-1} a_{2n}e^{-i2\pi m2n/2N_1} \quad + \sum_{n=0}^{N_1-1} a_{2n+1}e^{-i2\pi m(2n+1)/2N_1} \tag{4}$$

$$= \sum_{n=0}^{N_1-1} a_{2n}e^{-i2\pi mn/N_1} \quad + e^{-i2\pi m/N}\sum_{n=0}^{N_1-1} a_{2n+1}e^{-i2\pi mn/N_1} \tag{5}$$

Om man tittar efter ser man att den första summan är en diskret Fouriertransform av de N_1 *jämna* koefficienterna a_{2n}, och den andra summan av de N_1 *udda* koefficienterna a_{2n+1}. Om vi skriver a_n^j och a_n^u för de udda respektive jämna koefficienterna, och motsvarande för DFT-versionen, så får vi

$$\hat{a}_m = \hat{a}_m^j + e^{-i2\pi m/N}\hat{a}_m^u \qquad \text{för} \qquad 0 \le m \le N_1 - 1 = N/2 - 1 \tag{6}$$

och (eftersom $\hat{a}_m^j$ och $\hat{a}_m^u$ är periodiska med period N_1)

$$\hat{a}_m = \hat{a}_{m-N_1}^j + e^{-i2\pi m/N}\hat{a}_{m-N_1}^u \qquad \text{för} \qquad N_1 \le m \le N - 1. \tag{7}$$

Kostnaden för att beräkna $\hat{a}_m^j$ och $\hat{a}_m^u$ är i storleksordningen $2N_1^2 = N^2/2$ operationer. Förutom det går det åt en multiplikation och en addition per m, alltså sammanlagt $N^2/2 + N$ operationer (en operation räknas i dessa sammanhang ofta som en addition och en multiplikation). Kostnaden har halverats. Men den stora vinsten gör man om N_1 i sin tur är ett jämnt tal, så att även $\hat{a}_m^j$ och $\hat{a}_m^u$ kan beräknas på liknande sätt, och så vidare. Optimalt resultat får man om $N = 2^k$, sämst får man om N är ett primtal (för då kan man inte göra något alls åt kostnaden).
Av detta skäl vill man ofta välja N som en tvåpotens, och om man har ett antal sampelvärden som ligger nära en tvåpotens kan det se ut som en god ide att helt enkelt fylla på med så många nollor som behövs. Det görs i alla händelser ofta i praktiken ibland. Men i matlab faktoriseras istället N i så många primfaktorer som möjligt, och vinner den beräkningstid som går.

Laborationsuppgift 1

Det enklaste är att göra dessa uppgifter med hjälp av matlab, och jag har anpassat denna beskrivning till det. Men den som är intresserad av mer komplicerade programmeringsuppgifter får gärna göra det på annat sätt.

Denna del av laborationen handlar bara om att bekanta sig med `fft` i matlab. Börja med att starta matlab och läsa manualbladet till `fft`

1. Sampla signalen $f(t) = \sin(10t) + 2\cos(15t)$ N gånger i intervallet $0 \leq t < T$, där $T = 100$, i.e. skapa en vektor `x = [x(1), x(2), ..... , x(N)]`, där `x(k+1)` är det k-te stickprovet ("samplet") av signalen. Observera att numreringen av vektorelement i matlab alltid startar med 1. Beräkna den diskreta Fouriertransformen av `x` med hjälp av funktionen `fft`, och plotta realdel, imaginärdel och absolutbelopp av resultatet. Välj ett antal olika N (stora och små) och se hur resultatet förändras. Förklara utseendet på graferna.

2. Två klassiska sätt att modulera en radiovåg för att överföra till exempel tal är *amplitudmodulation* och *frekvensmodulation*. Låt oss som ett exempel betrakta en bärfrekvens på $\nu = 30$ MHz (alltså kortvågsbandet), och antag att "talet", den signal man vill överföra består av signalen $\phi(t)$. Den modulerade signalen kommer då att ha utseendet

$$f_A(t) = \cos(\nu t)(1 + \alpha\phi(t)) \qquad \text{(am), och} \tag{8}$$

$$f_F(t) = \cos(\nu t + \alpha\phi(t)) \qquad \text{(fm)}\,. \tag{9}$$

I båda fallen är α en konstant som bestämmer modulationsgraden.
Låt nu $\nu = 50$, och låt $\alpha = 0.2$, och antag att

$$\phi(t) = \sin(4t) + 0.5\cos(8t). \tag{10}$$

Plotta den modulerade signalen, och dess spektrum (med hjälp av `fft`) för amplitud- och frekvensmodulation.

Laborationsuppgift 2

En vanlig typ av "signal" är brus. Det innebär att $\phi(t)$ på något sätt är slumpvis fördelat. En samplad signal som vi är intresserade av här kan ha olika beteende. Ett är att $\phi(t_n$ är helt oberoende av varandra, till exempel normalfördelade. Med matlab kan en sådan brussignal (med N sampel) genereras genom

`phi = randn(1,N).`

En annan slags brus uppstår om istället $\psi(t_{n+1}) - \psi(t_n)$ är oberoende slumptal. I matlab kan man skriva t.ex.

```
psi(1) = phi(1);
for j=2:N, psi(j)=psi(j-1)+phi(j); end
```

(För att få ett mer realistiskt brus borde man här egentligen byta `phi(j)` i summan mot `phi(t)*sqrt(dt)`, där `dt`$= \Delta t = t_{n+1} - t_n$, men det ger bara ett fel med en konstant faktor, och för vårt ändamål spelar det ingen roll.

- Skapa en brussignal `phi(n)` och `psi(n)` enligt ovanstående. Plotta tidsserierna, och studera även spektrum med hjälp av FFT-algoritmen (i Matlab, till exempel).

- Med hjälp av spektralanalys kan man försöka hitta signaler i brus. På kurshemsidan finns en länk till en fil som innehåller sampel från en brusig signal, samplad över ett tidsintervall på 10 sekunder. Hämta hem den, läs in den i matlab, och undersök om det finns något i den som liknar en signal. Ange i så fall tydliga frekvenser, och deras relativa amplitud. **OBS**. Datafilerna genereras individuellt till var och en som hämtar den. Följ instruktionerna för nedhämtning. I laborationsrapporten skall identfikationstalet `ftal` anges.

- Ett enkelt filter kan modelleras med en differentialekvation

$$v''(t) + 2kv'(t) + \omega^2 v(t) = f(t); \tag{11}$$

  här är $f(t)$ insignalen, och $v(t)$ utsignalen. Denna deluppgift består i att mata in den bruset $\phi(t)$ som beskrivits ovan och plotta spektrum för utsignalen. I matlab kan man göra det på detta sätt:

  – Konstruera en matlabfunktion, en fil `odedef.m`, som innehåller definitionen av differentialekvationen, enligt

```
function out1 = odedef(t,y)
global tlist flist; % tlist är vektorer som innehåller
                    % sampeltidpunkter och sampel-
värden för insignalen
a=0.3;  % dämpning
b=110.0;  % omega^2
out1 = [y(2); ...
        interp1(tlist,flist,t) ...
                     % detta skapar en interpolerad
                     % funktion av (tlist, flist)
         -a*y(2)-b*y(1)];
```

  – Skapa sen en matlab skriptfil till exempel så här:

```
% globala deklarationer
global tlist flist
N = 10000;
Tmax = 10.0;
dt = Tmax / N;
tlist = dt : dt : Tmax;
% skapa en insignal ...
flist = 10*randn(1,N);
[ T, v ] = ode45('odedef', tlist, [0, 0]);
```

  Efter körning av denna fil, innehåller `v(j,1)` lösningen vid tid t_j, och `v(j,2)` innehåller lösningens derivata.

Laborationsuppgift 3

Denna del av laborationen handlar om ickelinjäriteter och hur dessa påverkar spektrum.

- En förstärkare skall idealt sett vara linjär, utsignalen $v(t)$ skall vara proportionell mot insignalen $f(t)$:

  $$v(t) = Kf(t)\,. \tag{12}$$

  En riktig förstärkare kan tänkas istället ge en utsignal på formen

  $$v(t) = Kf(t) + bf(t)^2 + cf(t)^3 \tag{13}$$

  Låt $f(t) = \cos(10t)$ eller $f(t) = \cos(10t) + \sin(13t)$, och studera utsignalens spektrum för olika värden på b och c. Gör också en teoretisk analys för att förklara resultatet.

- Vi går tillbaka till filtrets differentialekvation, men modifierar den en aning:

  $$v''(t) + bv(t)(1 + \eta v(t)^2) = A\cos(\omega t) \tag{14}$$

  I denna ekvation har dämpningen tagits bort, men en en ickelinjär (kubisk) term har tillkommit. Ekvationen brukar kallas Duffingekvationen. Studera spektrum av lösningen $v(t)$ som ovan, för olika kombinationer av b, η, ω och A. Studera också vad som händer om högerledet ersätts av en summa av två cosinusfunktioner med olika frekvens. Försök att motivera resultatet teoretiskt.

Rapport och bedömning av laboration

Laborationen är inte obligatorisk, men kan ge högst 5 bonuspoäng vid tentamen. Även en delvis utförd laboration ger viss poäng. För att få betyg 5 i kursen som helhet krävs, förutom de vanliga kraven vid tentamenstillfället, även att laborationen har utförts.

Rapporten skall innehålla en beskrivning av den utförda laborationen, utskrifter av plottar i den mån som behövs för att förklara resultaten, samt de beräkningar som behövs för att motivera resultaten.

Rapporterna får gärna vara handskrivna *under förutsättning att de skrivs med läslig handstil!*.

Rapporterna skall vara inlämnade **absolut senast fredag i tentamensveckan**.