

## Skoleøving 1

### Grunnleggende trening i bruk av Matlab

#### Oppgave 1 (grunnleggende trening)

- a) **Lag et "analogt" sinusformet signal  $x$ .** Matlab kan kun behandle digitale signaler. Vi må derfor lage en diskret (digital) versjon av det analoge signalet. La oss se på et sinusformede analogt signal med frekvensen 440 Hz. Dette kan skrives slik:

$$x(t) = \sin(2 \pi 440 t)$$

I Matlab må vi punktprøve (sample) dette signalet. Vi velger å punktprøve 8000 ganger pr sekund. Da blir punktprøvingsintervallet  $T_s = 1/8000S = 0.125mS$ . Vi ønsker at signalet skal vare i 5 sekunder. Det tilsvarer 40000 punktprøver som vi må ha liggende i en tabell i Matlab. Denne tabellen representerer signalet. I Matlab omtaler vi denne tabellen som en *vektor*.

I Matlab gjør vi dette slik:

```
Fs = 8000;           % definerer signalets punktprøvefrekvens
Ts=1/Fs;             % Ts=0.125E-3 punktprøvingsintervall
t=0:Ts:5;            % lager en tabell t med alle tidspunktene
```

Så lager vi signalet  $x$ :

```
x=sin(2*pi*440*t);    % legger alle signalverdiene i en tabell x
```

Nå kan vi plotte signalet:

```
plot(t, x);           % plotter signalet i et figurvindu
plot(t(1:1000), x(1:1000)); % plotter de første 1000 verdier
```

Til slutt skal vi høre på lyden i hodetelefon! (Denne må kobles til på PC-ens lydutgang.)

```
soundsc(x, Fs); % sender signalet (lyden) til PCens lydkort
```

Sett på *navn* på aksene og *tittel* på plottet ved å bruke kommandoene *xlabel*, *ylabel* og *title*.

- b) **Samle alle kommandoene i en Matlab skript-fil:**  
Skriv alle kommandoene inn i en fil (klipp og lim) og lagre med filnavn **sinskr.m**  
(Sett inn egne kommentarer som forklarer koden)

```
Fs = 8000;           % dine kommentarer
Ts=1/Fs;             % .. og så videre
t=0:Ts:5;
x1=sin(2*pi*440*t);
plot(t, x);
soundsc(x,Fs);
```

- c) **Lag din egen Matlabfunksjon:**  
Modifiser koden fra punktet over til slik som vist under og lagre denne i en fil med navn **sinfunk.m**

```
function x=sinfunk(f1)
%   x=sinfunk(f1)
%
%   Denne funksjonen lager er sinusformet signal med
%   frekvensen f1. Samplingsfrekvensen er 8000Hz,
%   f1 må derfor være mindre enn 4000Hz.
%
%   Eksempel: x=sinfunk(440)

Fs = 8000;           % dine kommentarer
Ts=1/Fs;             % .. og så videre
t=0:Ts:5;
x=sin(2*pi*f1*t);
plot(t, x);
soundsc(x,Fs);
```

- d) **Til slutt: (kan hoppe over, det er viktigere å gjøre oppgave 2)**

1. Du kan nå kalle funksjonen ved å skrive for eksempel: lyd=**sinfunk(400)** ;  
Hør på lyden i hodetelefon!
2. Eksperimenter med andre frekvenser
3. Prøv å skrive **help sinfunk**, hva skjer?
4. Greier du å plotte en mindre del av signalet (lyd) etter at funksjonen er ferdig med sine beregninger? (Plott for eksempel bare de 100 første verdiene i av signalet.)

Husk å lagre alt du har gjort til seinere bruk!

## Oppgave 2

Vi skal nå undersøke summasjon av to cosinussignaler med middelvei forskjellig fra null.

Vi tar utgangspunkt i formelen  $x(t) = A_0 + A_1 \cos(2\pi f_1 t + \varphi_1) + A_2 \cos(2\pi f_2 t + \varphi_2)$

I punkt a) under blir du bedt om å lage en «function» *sumcos2* som kan generere og plote en slik sammensatt funksjon  $x(t)$ . En beskrivelse av funksjonen med tilhørende datastruktur er beskrevet under:

$$[x, t] = \text{sumcos2}(m, fu1, fu2, tstart, tstop)$$

der

- x er genererte funksjonsverdier
- t er tilhørende tidsverdier
- m er middelvei
- fu1 er en liten vektor med parameterne  $[A_1 \phi_1 f_1]$  for funksjon 1
- fu2 er en liten vektor med parameterne  $[A_2 \phi_2 f_2]$  for funksjon 2
- tstart er tidsverdi for start
- tstop er tidsverdi for stopp

Eksempel på bruk: `[x, t] = sumcos2(5, [10 pi/3 50], [3 4*pi/3 150], 0, 0.040)`

Et forslag til fornuftig datastruktur er illustrert under:

m

fu1

$A_1$	$\phi_1$	$f_1$
-------	----------	-------

fu2

A <sub>2</sub>	phi <sub>2</sub>	f <sub>2</sub>
----------------	------------------	----------------

tstart

tstop

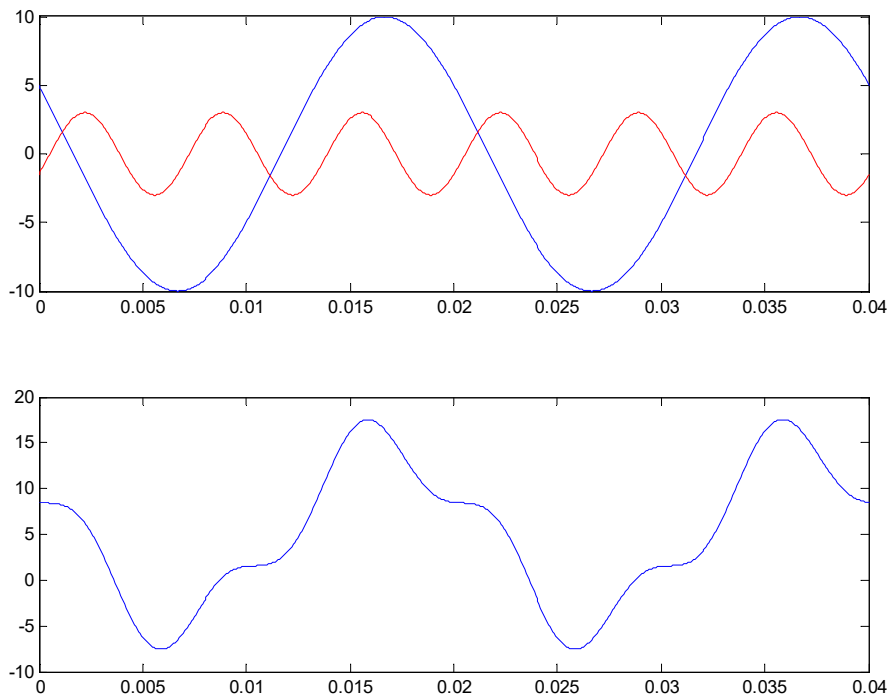
x (mange funksjonsverdier)

[illegible]

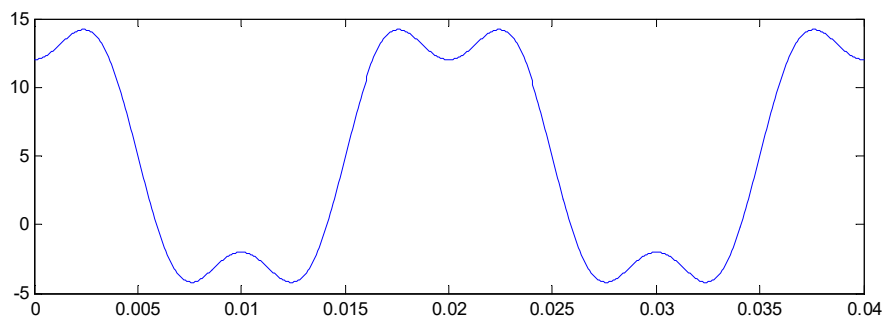
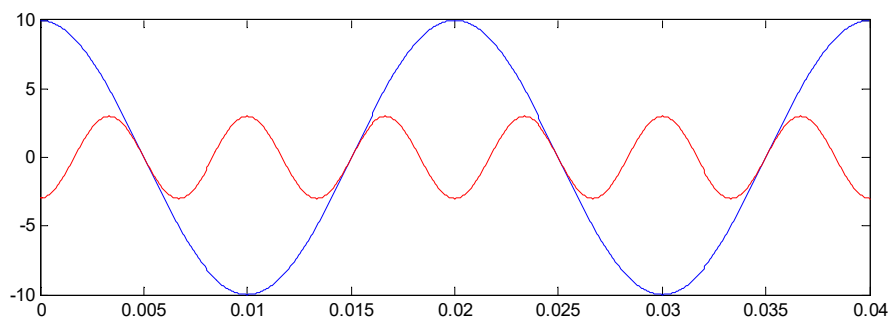
t (mange tidsverdier)

[illegible]

Under vises et eksempel på et plot generert av en slik funksjon. Funksjonen genererer plot som viser både de to sinusformede funksjonene hver for seg samt den sammensatte funksjonen:



- Du skal nå lage en slik «function» *sumcos2*. Hvis du vil ha hjelp til å lage funksjonen så kan du ta utgangspunkt i eksemplet nedenfor.
- Bruk funksjonen du har laget og plot summen av to cosinusfunksjoner der frekvensen er den samme (50Hz), men se hva som skjer med summen når amplitude og fase endres for den ene funksjonen. La middelveiden være null.
- Bruk funksjonen du har laget og plot summen av to cosinusfunksjoner der frekvensene er henholdsvis 50Hz og 150Hz. **Hva blir periodetiden?** Prøv å variere de andre parameterne (ikke frekvensene) og observer hva som skjer.
- La frekvensene være henholdsvis 100Hz og 150Hz. **Hva blir periodetiden nå?** Prøv å variere de andre parameterne (ikke frekvensene) og observer hva som skjer.
- Prøv å regulere amplituden og fasen til funksjonene til du får bildet under. Frekvensene skal være 50Hz og 150Hz. Noter amplitudene og fasene når du har funnet noe som er mest mulig likt.



Vis fram til veileder / lærer hva du har fått til!

Eksempelfunksjon på neste side:

```

function [ x, t ] = sumcos2(m, fu1, fu2, tstart, tstop)
%
% [ x, t ] = sumcos2(m, fu1, fu2, tstart, tstop)
% der
% - x er funksjonsverdier
% - t er tidsverdier
% - m er middelfverdi
% - fu1 er en liten vektor med [A1 phi1 f1]
% - fu2 er en liten vektor med [A2 phi2 f2]
% - tstart er tidsverdi for start
% - tstop er tidsverdi for stopp
%
% Eksempel: [x, t] = sumcos2(5, [10 pi/3 50], [3 4*pi/3 150], 0, 0.040)

f2 = fu2(3);           % tar utgangspunkt i funksjonen med høyest frekvens
T2 = 1/f2;
t_inc = T2/100;        % lager 100 punkter pr periode for å få pen kurve

t = tstart:t_inc:tstop;

xm = m;                % middelfverdi
x1 = fu1(1)*cos(2*pi*fu1(3)*t + fu1(2)); % cosfunksjon nr 1
x2 = fu2(1)*cos(2*pi*fu2(3)*t + fu2(2)); % cosfunksjon nr 2
x = xm + x1 + x2;      % lager summert funksjon

subplot(2,1,1)         % plotter enkeltfunksjoner
plot(t,x1,'b');
hold on;
plot(t,x2,'r');
hold off;

subplot(2,1,2)         % plotter sum
plot(t,x)

end

```