

# Spektrala Transformer Övningsmaterial

# Innehåll

<b>1</b>	<b>Komplexa tal &amp; Sampling</b>	<b>2</b>
<b>2</b>	<b>Filter</b>	<b>6</b>
<b>3</b>	<b>Z-transform</b>	<b>11</b>
<b>4</b>	<b>Fourierserier</b>	<b>13</b>
<b>A</b>	<b>Svar</b>	<b>15</b>
<b>B</b>	<b>Matlabövningar</b>	<b>31</b>
<b>C</b>	<b>Pythonövningar</b>	<b>35</b>
<b>D</b>	<b>Formelblad</b>	<b>40</b>
<b>E</b>	<b>Extentor med lösningar</b>	<b>44</b>

Problemsamlingen bygger på bidrag från Laura Enflo, Giampiero Salvi, Lasse Björkman (anpassade från kompendiet *Övningar i digital signalbehandling*), Jonas Beskow, Kalin Stefanov, Bajibabu Bollepalli och Patrik Jonell

# Kapitel 1

## Komplexa tal & Sampling

### Komplexa tal

Om man endast använder den reella tallinjen är det inte möjligt att lösa ekvationer av typen  $x^2 = -5$ . Det komplexa talplanet har inga sådana begränsningar. Lösningen till ekvationen  $x^2 = -1$  är  $x = \sqrt{-1} = i$ . Ett komplext tal kan skrivas på formen  $z = a + bi$  och dess konjugat  $\bar{z} = a - bi$ . Längden eller absolutvärdet av  $z$  är  $|z| = |a + bi| = \sqrt{a^2 + b^2}$ .

Observera att  $i$  skrivs som  $j$  inom främst elektrotekniken, eftersom den matematiska beteckningen annars skulle kunna förväxlas med beteckningen för ström.

1. Förenkla så långt som möjligt:

(a)

$$(3 + 4j) - (2 + 3j)$$

(b)

$$5 * 2j$$

2. Förenkla så långt som möjligt:

(a)

$$(2 + j)(2 - j)$$

(b)

$$(2 + j)^2$$

(c)

$$\frac{(1 - j)}{(3 + j)}$$

3. Visa att

$$z\bar{z} = |z|^2$$

4. Förenkla så långt som möjligt:

(a)

$$|(3 + 3j)(2 - 2j)|$$

(b)

$$|(1 - 3j)(2 + j)|$$

Komplexa tal kan också skrivas på polär form, där x- och y-koordinaterna bestäms av  $x = r\cos\theta$  och  $y = r\sin\theta$ .  $r$  definieras som  $r = |z|$ .  $\theta$  definieras som vinkeln mellan x-axeln och Ortsvektorn för  $z$  och kallas därför för argumentet för  $z$ . För att göra argumentet entydigt bestämt kan man välja att  $-\pi < \theta \leq \pi$ . Den polära formen för  $z$  blir sålunda  $z = r(\cos\theta + j\sin\theta)$ . Eftersom Eulers formel säger att  $\cos\theta + j\sin\theta = e^{j\theta}$  kan komplexa tal även uttryckas på exponentialform, det vill säga  $z = re^{j\theta}$ . Man kan visa att formeln  $[e^{j\theta}]^n = e^{jn\theta}$  gäller för komplexa tal (kallas De Moivres formel).

5. Hitta argumentet  $\theta$  som ligger på intervallet  $-\pi < \theta \leq \pi$  om:

(a)

$$z = 1$$

(b)

$$z = 1 + j$$

(c)

$$z = -1 + \sqrt{3}j$$

6. Skriv på exponentialform

(a)

$$z = -j$$

(b)

$$z = 1 - j$$

(c)

$$z = \sqrt{3} + j$$

7. Skriv på så enkel form som möjligt

(a)

$$\frac{5j}{2 + j}$$

(b)

$$\frac{5}{(1 - j)(2 - j)(3 - j)}$$

(c)

$$(1 - j)^4$$

8. Visa att

$$(-1 + j)^7 = -8(1 + j)$$

9. Om  $z_2 \neq z_3$ , visa att

$$\frac{|z_1|}{|z_2 + z_3|} \leq \frac{|z_1|}{||z_2| - |z_3||}$$

Ledning: Triangelolikheten och dess bevis.

10. Visa att

$$\cos^2 \theta = \frac{1 + \cos 2\theta}{2}$$

Ledning:

- Gå till komplex form och tillbaka.
- Utnyttja Eulers formel:

$$\cos a = \frac{e^{ja} + e^{-ja}}{2}$$

11. Visa att

$$e^{j(\alpha-\beta)} + e^{j(\alpha+\beta)} = e^{j\alpha} 2 \cos \beta$$

12. Uttrycket

$$z(\theta) = e^{j\theta} + \frac{1}{2}e^{j2\theta}$$

beskriver en kurva i komplexa talplanet. Låt  $\theta$  gå från 0 till  $2\pi$  och skissa kurvan (beräkna t.ex. en punkt varje  $\pi/4$ ). Rita sedan upp kurvan i Matlab och jämför.

## Sampling

För att omvandla en signal från analog till digital form krävs två steg: först sampling och sedan kvantisering. Vid sampling används en viss samplingsfrekvens  $f_s$  och signalen, till exempel  $x(t)$ , omvandlas från att vara tidskontinuerlig till att bli tidsdiskret, vilket kan uttryckas som  $x(t) \rightarrow x(nT_s)$ , där samplingsperioden  $T_s = 1/f_s$ . När man beskriver tidsdiskreta signaler använder man normerade frekvenser  $\nu_0 = f_0/f_s = f_0T_s$ .

13. Signalen  $x(t) = \sin(2\pi f_0 t)$  med frekvensen  $f_0 = 2000$  Hz samplas med samplingsfrekvensen  $f_s = 8000$  Hz.

(a) Beräkna den normerade frekvensen  $\nu_0$ .

(b) Skriv ett uttryck för den samplade signalen  $y(n)$ .

14. Signalen  $x(t) = \cos(2\pi f_0 t)$  skall samplas. Rita upp den samplade signalen för fem olika frekvenser  $f_0$  (enligt nedan) när samplingsperioden är  $T_s = 1$ :

(a)  $f_0 = 0.1$

(b)  $f_0 = 0.2$

(c)  $f_0 = 0.3$

(d)  $f_0 = 0.5$

(e)  $f_0 = 0.7$

(f) Finns det några likheter mellan alternativen c) och e)?

Frekvenser utöver halva samplingsfrekvensen  $f_s$  ger upphov till vinkningsdistorsion. För att undvika detta måste signalen antingen sakna energi som ligger över den s k Nyquistfrekvensen ( $f_s/2$ ) eller annars låter man signalen före samplingen passera ett antivinkningsfilter som tar bort de oönskade frekvenserna. Det går även att sampla med en annan, högre frekvens och sedan återgå till den tänkta samplingsfrekvensen efter digital filtrering.

15. Vi har signalen  $x(t) = 2\sin(2\pi f_0 t)$  som samplas med  $f_s = 8000$  Hz. För vilken/vilka av nedanstående frekvenser kommer vinkningsdistorsion att uppkomma?

(a) 1000 Hz

(b) 3000 Hz

(c) 5000 Hz

(d) 7000 Hz

MATLAB PROBLEMS
-----------------

16. Given  $\omega$  in the interval  $-2\pi \leq \omega \leq 2\pi$  (*tip: use small step for  $w$ , e.g.  $\pi/100$* ), plot  $\sin(\omega)$  and  $\cos(\omega)$ .

17. Given  $\omega$  in the interval  $-2\pi \leq \omega \leq 2\pi$  (*tip: use small step for  $w$ , e.g.  $\pi/100$* ), plot the real and imaginary parts of  $e^{j\omega}$ .

18. Given  $\alpha = \frac{\pi}{2}$  and  $\beta = \frac{\pi}{18}$ , plot the real- and imaginary parts and the magnitude of the exponentially decaying signal  $x(t) = e^{(j\alpha - \beta)t}$  in the interval  $0 \leq t \leq 25$  (*tip: use small step for  $t$ , e.g. 0.1*).

19. Given  $\alpha = \frac{\pi}{2}$  and  $\beta = \frac{\pi}{18}$ , plot the real- and imaginary parts and the magnitude of the exponentially growing signal  $x(t) = e^{(j\alpha + \beta)t}$  in the interval  $0 \leq t \leq 25$  (*tip: use small step for  $t$ , e.g. 0.1*).

# Kapitel 2

## Filter

1. Compute the frequency response of the filter:

$$y[n] = x[n] - 0.8x[n - 1]$$

2. Find the zeros of the transfer function for the filters:

(a)

$$y[n] = x[n] - 0.25x[n - 2]$$

(b)

$$y[n] = x[n] + 0.25x[n - 2]$$

3. Find the time-discrete feed-forward filter with three terms that has the frequency response:

$$H(\omega) = \frac{(e^{j\omega} - 0.5e^{j\pi/3})(e^{j\omega} - 0.5e^{-j\pi/3})}{(e^{j\omega})^2}$$

*Hint:*  $\cos(\pi/3) = 0.5$ .

4. Find the impulse response of the filter with frequency response:

$$H(\omega) = (1 + e^{-j\omega})^2$$

5. A filter has the following impulse response:

$$h(n) = [.5, .5]$$

Compute the output sequence of the filter, if the input sequence is

(a)  $x(n) = [1, 0, 1, 0, 1, 0]$

(b)  $x(n) = [1, 1, 1, 2, 2, 2]$

(c)  $x(n) = [1, 2, 3]$

\* testa i MATLAB! (problem 16)

6. A filter has the following impulse response:

$$h(n) = [3, 2, 1]$$

Compute the output of the filter, if the input is

(a)  $x(n) = [1, 0, 0, 1, 0, 0]$

(b)  $x(n) = [3, 2, 1]$

(c)  $x(n) = [1, 2, 3]$

\* testa i MATLAB!

7. Rita följande tidssekvenser.

(a)  $\delta[n]$

(e)  $u[n - 4]$

(b)  $\delta[n - 2]$

(f)  $u[n] - u[n - 4]$

(c)  $\delta[n] + \delta[n - 2]$

(g)  $u[-n]$

(d)  $u[n]$

(h)  $u[4 - n]$

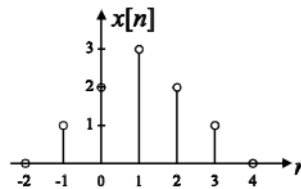
8. Givet tidssekvensen  $x[n]$  i nedanstående figur. Rita följande tidssekvenser.

(a)  $x[n - 1]$

(c)  $x[-n]$

(b)  $x[n + 3]$

(d)  $x[2 - n]$



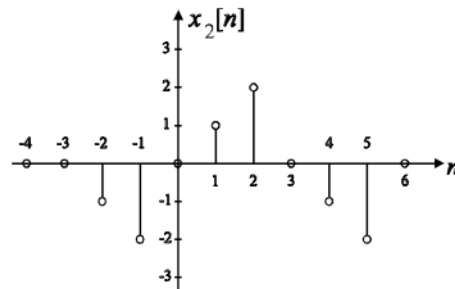
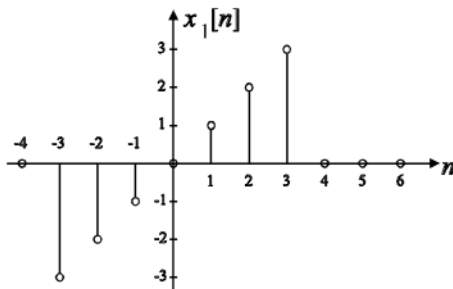
9. Givet tidssekvenserna  $x_1[n]$  och  $x_2[n]$  nedanstående figurer. Rita följande tidssekvenser.

(a)  $y[n] = x_1[n] + x_2[n]$

(c)  $y[n] = x_1[n] \cdot x_2[-n]$

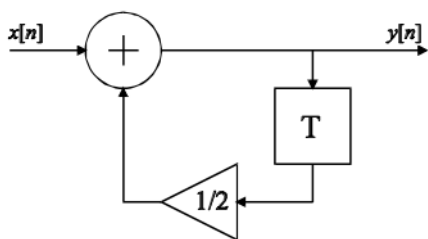
(b)  $y[n] = x_1[n] \cdot x_2[n]$

(d)  $y[n] = x_1[n] \cdot x_2[4 - n]$

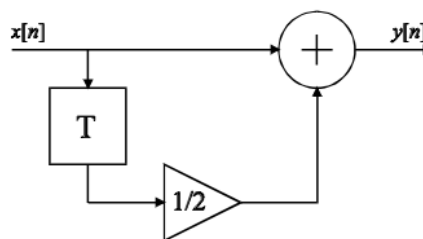




10. Bestäm ett samband mellan insignal  $x[n]$  och utsignal  $y[n]$  för nedanstående tidsdiskreta kretsar.



(a) 1:a ordningens IIR-filter



(b) 1:a ordningens FIR-filter

11. Rita en realisering till var och en av de tidsdiskreta kretsar som beskrivs av nedanstående differensekvationer.

- (a)  $y[n] - \frac{1}{2}y[n-1] = x[n-1]$   
 (b)  $y[n] = x[n] + \frac{1}{2}x[n-2]$   
 (c)  $y[n] - \frac{3}{4}y[n-1] + \frac{1}{8}y[n-2] = x[n]$

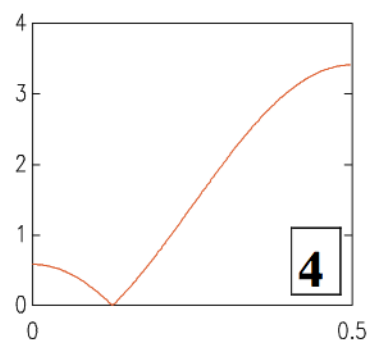
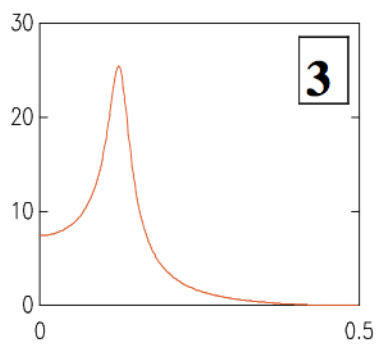
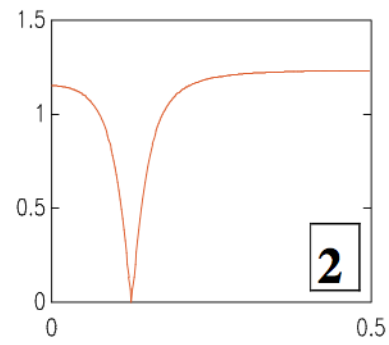
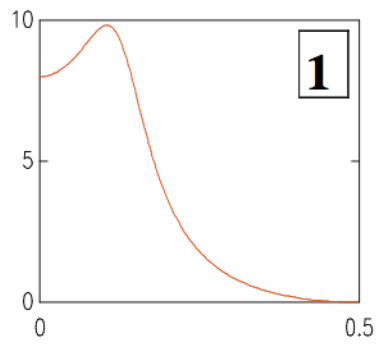
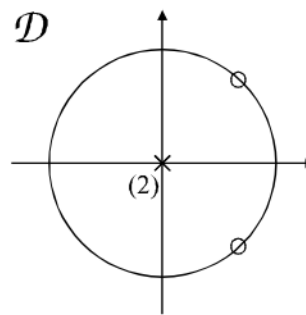
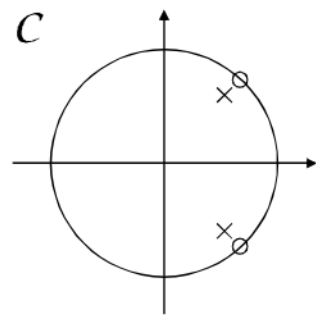
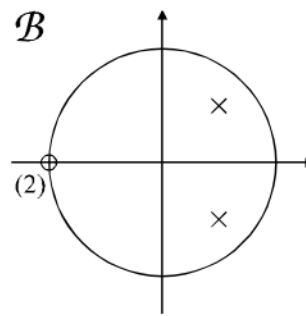
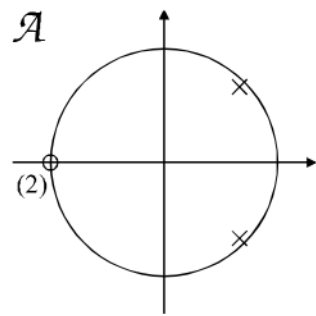
12. Bestäm faltningen mellan  $h[n] = (\frac{1}{4})^n \cdot u[n]$  och  $x[n] = u[n]$ .

13. En tidsdiskret krets med två poler och två nollställen har sina poler placerade i  $p_{1,2} = 0.9 \cdot e^{\pm j \frac{\pi}{4}}$ . Skissa amplitudfunktionen då.

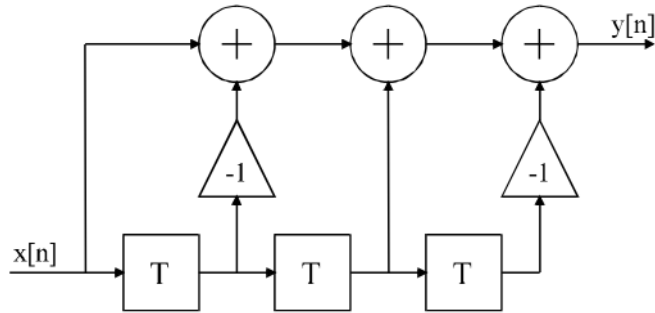
- (a)  $n_{1,2} = -1$   
 (b)  $n_{1,2} = +1$   
 (c)  $n_{1,2} = \pm 1$   
 (d)  $n_{1,2} = \pm j$   
 (e)  $n_{1,2} = e^{\pm j \frac{\pi}{4}}$

\* testa i MATLAB! (problem 17)

14. Nedan ges pol-nollställesdiagram till fyra olika tidsdiskreta kretsar. Para ihop vart och ett med motsvarande amplitudfunktion.



15. Givet nedanstående FIR-filer.



- Bestäm kretsens poler och nollställen
- Bestäm och rita kretsens amplitudfunktion  $|H[e^{j\omega}]|$
- Bestäm och rita kretsens fasfunktion  $\arg\{H[e^{j\omega}]\}$
- Vid vilka frekvenser blir  $|H[e^{j\omega}]| = 0$
- Viken typ av filter är det

MATLAB PROBLEMS

- 16.** Given an impulse response:

$$h(n) = [.5, .5]$$

Compute the output sequence of the filter, if the input sequence is

- $x(n) = [1, 0, 1, 0, 1, 0]$
- $x(n) = [1, 1, 1, 2, 2, 2]$
- $x(n) = [1, 2, 3]$

*Hint: Use 'conv' function in MATLAB*

- 17.** En tidsdiskret krets med två poler och två nollställen har sina poler placerade i  $p_{1,2} = 0.9 \cdot e^{\pm j\frac{\pi}{4}}$ . Skissa amplitudfunktionen då.

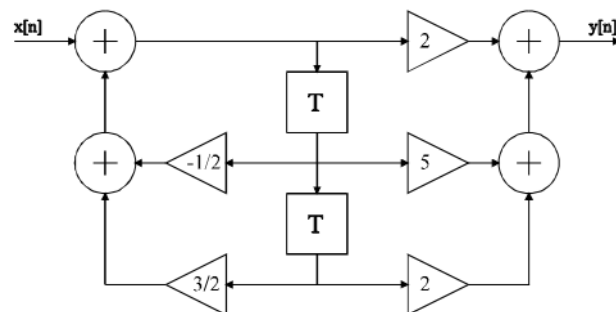
- $n_{1,2} = -1$
- $n_{1,2} = +1$
- $n_{1,2} = \pm 1$
- $n_{1,2} = \pm j$
- $n_{1,2} = e^{\pm j\frac{\pi}{4}}$

*Hint: Use MATLAB functions 'freqz()' or the more advanced 'fvtool()'*

# Kapitel 3

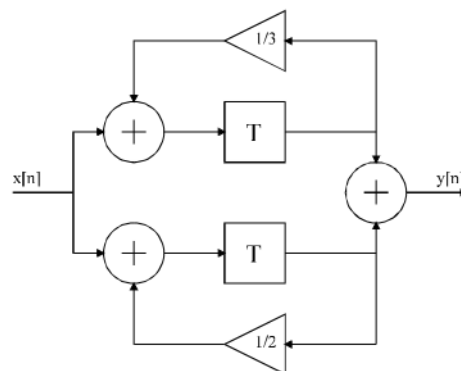
## Z-transform

1. En tidsdiskret krets beskrivs av nedanstående graf.



- (a) Bestäm systemfunktionen  $H[z]$
- (b) Avgör om kretsen är stabil

2. Bestäm systemfunktionen  $H[z]$  för nedanstående krets samt avgör om den är stabil.



3. Bestäm (mha definitionen) Z-transformen till.

- (a)  $x[n] = \delta[n]$
- (b)  $x[n] = u[n]$
- (c)  $x[n] = u[n - k]$
- (d)  $x[n] = a^n \cdot u[n]$

4. Uttryck  $Y[z]$  i  $X[z]$  då.

(a)  $y[n] = x[n - m]$

(b)  $y[n] = \alpha^n \cdot x[n]$

5. Bestäm (mha definitionen) Z-transformen till.

(a)  $x[n] = (\frac{1}{2})^n \cdot u[n]$

(b)  $x[n] = (\frac{1}{\sqrt{2}})^n \cdot \cos(\frac{\pi}{4}n) \cdot u[n]$

(c)  $x[n] = \sin(\frac{\pi}{2}n) \cdot u[n]$

6. Bestäm tidssekvensen  $x[n]$  då.

(a)  $X[z] = \frac{z}{z-\frac{1}{3}}, |z| > \frac{1}{3}$

(b)  $X[z] = \frac{z^2}{z^2-1}, |z| > 1$

testa i MATLAB! (problem 10)

7. Bestäm mha Z-transform utsignalen till en krets i vila med impulsvaret  $h[n]$  och insignalen  $x[n]$  då.

(a)  $h[n] = (\frac{1}{2})^n \cdot u[n]$  och  $x[n] = u[n]$

(b)  $h[n] = (\frac{1}{2})^n \cdot u[n]$  och  $x[n] = (\frac{1}{3})^n \cdot u[n]$

MATLAB PROBLEMS
-----------------

8. Plot a sinusoidal signal with: fundamental frequency ( $f$ )=10Hz, amplitude ( $A$ )=1, sampling frequency ( $f_s$ )=100Hz and duration ( $T$ )=2 seconds. (*Hint:  $y = A * \sin(2\pi ft)$  where  $t = \frac{T}{f_s}$* )

9. Find the poles and zeros of  $H(z)$ , and plot them.

$$H(z) = \frac{z^{-1} + 0.5z^{-2}}{1 + 0.6z^{-1} + 0.8z^{-2}} \quad (\text{Hint: Look up the 'zplane' matlab function})$$

10. Beräkna och plotta tidssekvensen  $x[n]$  då.

(a)  $X[z] = \frac{z}{z-\frac{1}{3}}, |z| > \frac{1}{3}$

(b)  $X[z] = \frac{z^2}{z^2-1}, |z| > 1$

*Tips: betrakta  $X(z)$  som ett filter: tidssekvensen  $x(n)$  är då filtrets impulssvar. Använd matlab-funktionen 'filter'.*

# Kapitel 4

## Fourierserier

1. Beräkna summan av den geometriska serien  $1+a+a^2+\dots+a^n$  om

(a)

$$a = \frac{1}{2}$$

(b)

$$a = -\frac{1}{3}$$

2. Vilka av följande funktioner är jämna och vilka är udda? Funktionen  $f(x)$  är

(a)  $\cos x$

(b)  $\sin x$

(c)  $x^2$

(d)  $x^3$

(e)  $\sin 3x$

(f)  $\cos 3x$

(g)  $1 - 2x$

3. En komplex Fourierserie skrivs som en reell Fourierserie:

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t) \text{ där } a_k = c_k + c_{-k} \text{ och } b_k = j(c_k - c_{-k})$$

En fyrkantsvåg har enligt formelsamlingen de komplexa Fourierkoefficienterna:

$$c_k = \begin{cases} -\frac{2j}{k\pi} & \text{om } k = 1, 3, 5, \dots \\ 0 & \text{annars} \end{cases}$$

(a) Vad blir de reella Fourierkoefficienterna?

(b) Varför är en av dem noll?

(c) Härled sambandet mellan komplexa och reella Fourierkoefficienter med hjälp av Eulers formel.

4. Funktionen  $f(t)$  har perioden  $2\pi$  och är udda. Då  $0 < t < \pi$  gäller att  $f(t) = t - \pi$ . Om man utvecklar  $f(t)$  som Fourierserie får man att

$$f(t) = -2 \sum_{n=1}^{\infty} \frac{\sin nt}{n}$$

Beräkna med hjälp av denna Fourierserie summan av serien

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

5. Vi har en fyrkantsvåg och en triangelvåg. De båda vågorna har samma periodicitet.
- Med vilken 'hastighet' går fyrkantsvågens Fourierkoefficienter mot noll?
  - Med vilken 'hastighet' går triangelvågens Fourierkoefficienter mot noll?
  - Vilken av vågtyperna klarar av smal bandbredd bättre, d v s vilken vågform distorderas minst om man undertrycker det högfrekventa området?  
Ledning: se a) och b).

MATLAB PROBLEMS
-----------------

6. En phasor (fasvektor) beskrivs av

$$x(n) = e^{j\omega n}$$

Låt  $0 \leq n \leq 1000$  och bestäm  $\omega$  så att phasorns vinkel gör ett helt varv över detta intervall.

- Plotta realdel och imaginärdel av  $x(n)$  i samma graf.

Funktionen

$$x(n) = -\frac{2j}{\pi}(e^{j\omega n} - e^{-j\omega n}) - \frac{2j}{3\pi}(e^{j3\omega n} - e^{-j3\omega n}) - \frac{2j}{5\pi}(e^{j5\omega n} - e^{-j5\omega n})$$

beskriver de tre första termerna i fourier-serien för en fyrkantvåg.

- Plotta funktionen.
- Plotta term 1, 1+2 och 1+2+3 i samma graf, och notera att funktionen blir mer lik en fyrkantvåg ju fler termer man har.
- Om du vill: lägg till fler termer efter samma mönster.

# Bilaga A

## Svar

### Komplexa tal

- 1

(a)  $1+i$

(b)  $10i$

- 2

(a)  $z = 2 + i$  och  $\bar{z} = 2 - i$  ger  $z\bar{z} = 5$

(b)  $z = 2 + i$  ger  $z^2 = 3 + 4i$

(c) Förlängning med konjugatet till nämnaren ger svaret

$$\frac{1}{5} - \frac{2i}{5}$$

- 4

(a)  $|6 + 6i| = \sqrt{12^2} = 12$

(b)  $|5 - 5i| = \sqrt{5^2 + 5^2} = 5\sqrt{2}$

- 5

(a)  $\theta = 0$

(b)  $\theta = \frac{\pi}{4}$

- 6

(a)  $z = e^{-i\frac{\pi}{2}}$

(b)  $z = \sqrt{2}e^{-i\frac{\pi}{4}}$

(c)  $z = 2e^{i\frac{\pi}{6}}$

- 7

(a)  $1 + 2i$

(b)  $\frac{i}{2}$



- 10

$$\cos^2\theta = \left(\frac{e^{j\theta} + e^{-j\theta}}{2}\right)^2 = \frac{e^{j2\theta} + 2 + e^{-j2\theta}}{4} = \frac{1}{2} + \frac{1}{2} \frac{e^{j2\theta} + e^{-j2\theta}}{2} = \frac{1 + \cos 2\theta}{2}$$

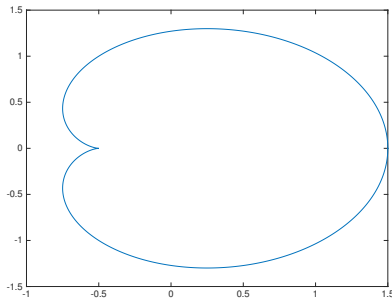
*Q.E.D.*

- 11

$$e^{j(\alpha-\beta)} + e^{j(\alpha+\beta)} = e^{j\alpha}e^{-j\beta} + e^{j\alpha}e^{j\beta} = e^{j\alpha}(e^{-j\beta} + e^{j\beta}) = e^{j\alpha}2\cos\beta$$

- 12

```
a = 0:pi/100:2*pi;
plot(exp(j*1*a)+exp(j*2*a)/2);
```



## Sampling

- 13

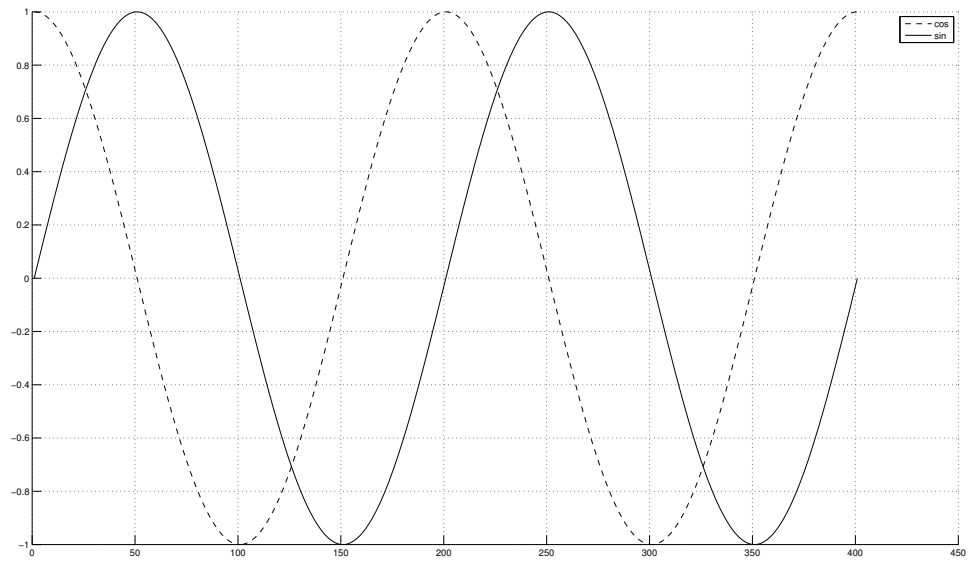
(a)  $v_0 = \frac{1}{4}$

(b)  $y(n) = x(nT_s) = \sin(2\pi f_0 n T_s) = \sin(2\pi v_0 n) = \sin\left(\frac{\pi}{2}n\right)$

- 14 (f) Man kan se att samplen exakt överensstämmer med varandra.
- 15 Alternativ (c) och (d)
- 16

```
w = -2*pi:pi/100:2*pi;

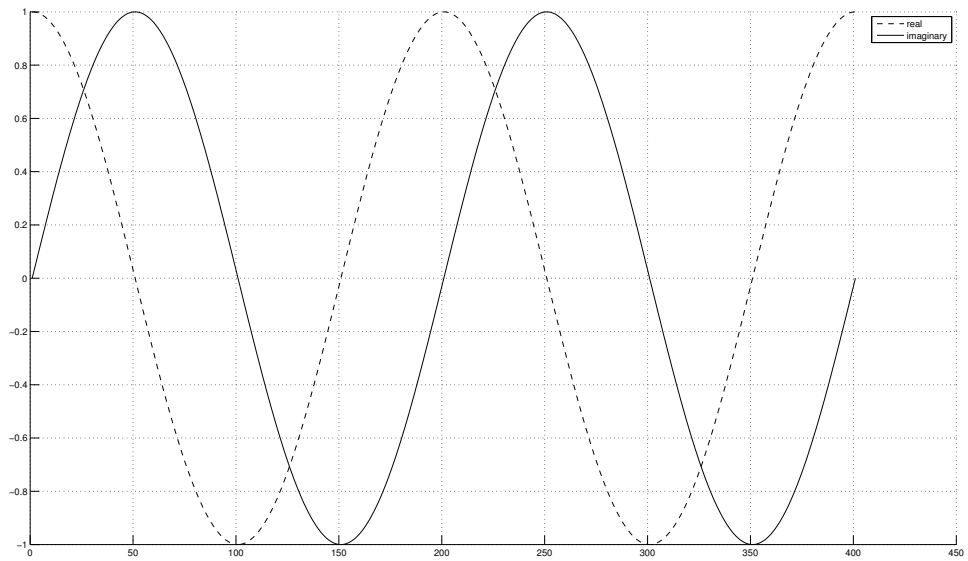
figure('Name','Cosine & Sine');
hold on;
y = cos(w);
plot(y,'k--');
y = sin(w);
plot(y,'k');
legend('cos','sin');
grid on;
hold off;
```



- 17

```
w = -2*pi:pi/100:2*pi;

figure('Name','Real & Imaginary');
hold on;
y = exp(1i*w);
plot(real(y),'k--');
plot(imag(y),'k');
legend('real','imaginary');
hold off;
grid on;
```



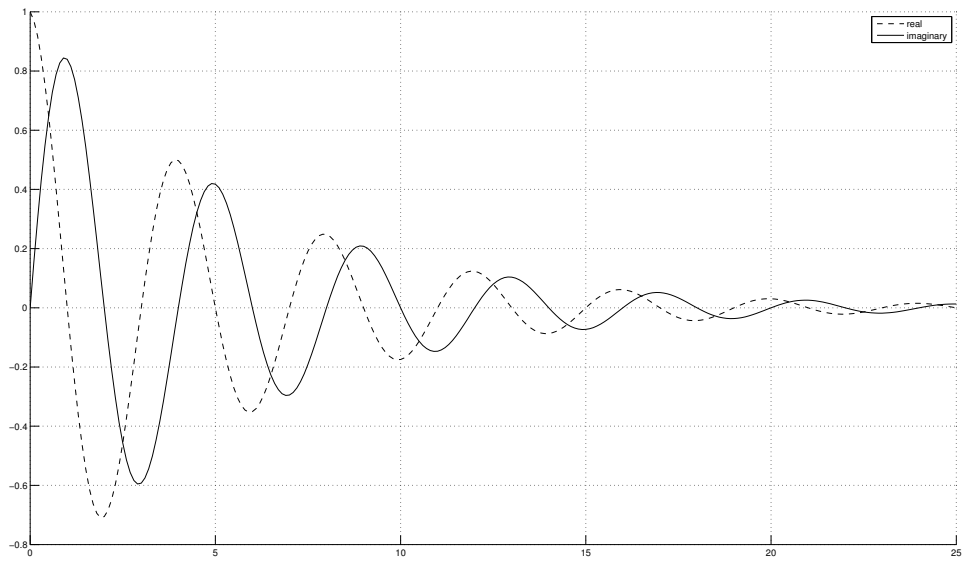
- 18

```

t = 0:0.1:25;

figure('Name','Decaying');
hold on;
alpha = pi/2;
beta = pi/18;
sig = exp((1i*alpha-beta)*t);
% calculate the signal magnitude
% pos_env = abs(sig);
% neg_env = -abs(sig);
% calculate the signal phase
% phase = angle(sig);
plot(t,real(sig),'k--');
plot(t,imag(sig),'k');
% plot(t,pos_env,'k');
% plot(t,neg_env,'k');
legend('real','imaginary');
% plot the signal spiral in 3D
% plot3(t,real(sig),imag(sig));
grid on;
hold off;

```



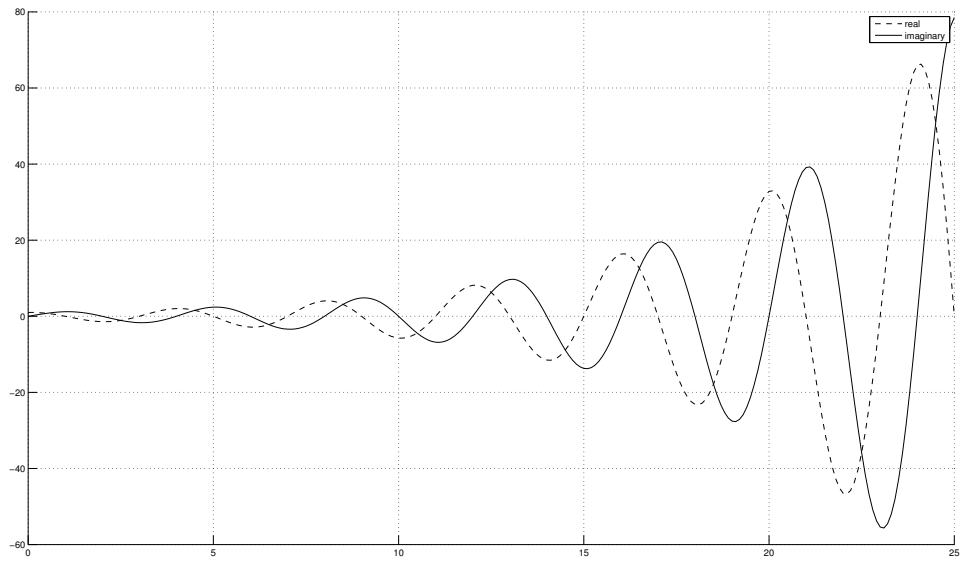
- 19

```

t = 0:0.1:25;

figure('Name','Growing');
hold on;
alpha = pi/2;
beta = pi/18;
sig = exp((1i*alpha+beta)*t);
% calculate the signal magnitude
% pos_env = abs(sig);
% neg_env = -abs(sig);
% calculate the signal phase
% phase = angle(sig);
plot(t,real(sig),'k--');
plot(t,imag(sig),'k');
% plot(t,pos_env,'k');
% plot(t,neg_env,'k');
legend('real','imaginary');
% plot the signal spiral in 3D
% plot3(t,real(sig),imag(sig));
grid on;
hold off;

```



## Filters

- 1

$$H(\omega) = 1 - 0.8e^{-j\omega}$$

- 2

(a)

$$H(z) = 1 - 0.25z^{-2} = \frac{z^2 - 0.25}{z^2} \rightarrow z^2 - 0.25 = 0 \text{ if and only if } z = \pm\sqrt{0.25} = \pm 0.5$$

(b)

$$H(z) = 1 + 0.25z^{-2} = \frac{z^2 + 0.25}{z^2} \rightarrow z^2 + 0.25 = 0 \text{ if and only if } z = \pm\sqrt{-0.25} = \pm 0.5i$$

- 3

$$\begin{aligned} H(z) &= \frac{(z - 0.5e^{j\pi/3})(z - 0.5e^{-j\pi/3})}{z^2} \\ &= \frac{z^2 - 0.5(e^{j\pi/3} + e^{-j\pi/3})z + 0.25}{z^2} \\ &= 1 - 0.5(e^{j\pi/3} + e^{-j\pi/3})z^{-1} + 0.25z^{-2} \\ &= 1 - \cos(\pi/3)z^{-1} + 0.25z^{-2} \\ &= 1 - 0.5z^{-1} + 0.25z^{-2} \end{aligned}$$

Thus,

$$y[n] = x[n] - 0.5x[n-1] + 0.25x[n-2]$$

- 4

$$\begin{aligned}
 H(z) &= (1 + z^{-1})^2 \\
 &= 1 + 2z^{-1} + z^{-2}
 \end{aligned}$$

Thus,

$$h[n] = \delta[n] + 2\delta[n - 1] + \delta[n - 2] \quad (y[n] = x[n] + 2x[n - 1] + x[n - 2])$$

- 5

(a)  $y(n) = [.5, .5, .5, .5, .5, .5]$

(b)  $y(n) = [.5, 1, 1, 1.5, 2, 2, 1]$

(c)  $y(n) = [.5, 1.5, 2.5, 1.5]$

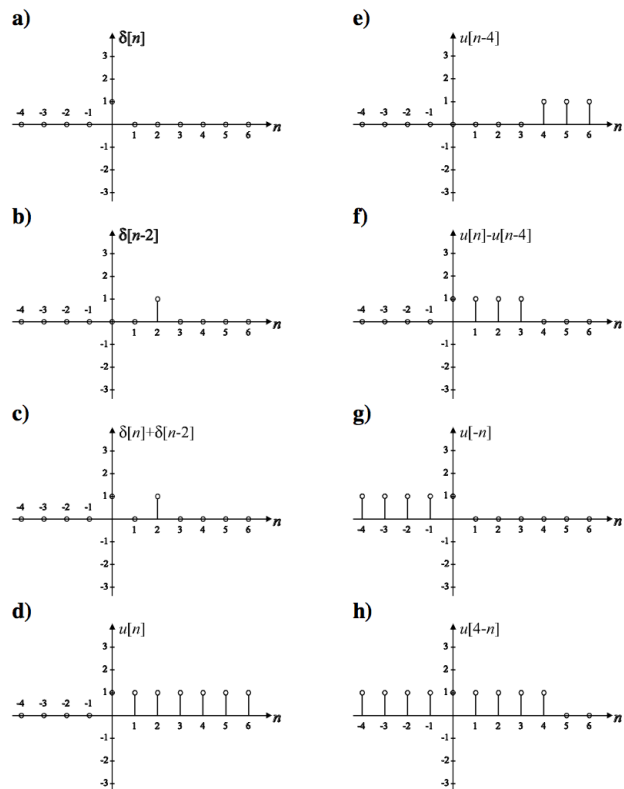
- 6

(a)  $y(n) = [3, 2, 1, 3, 2, 1]$

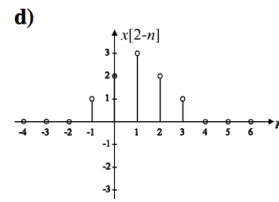
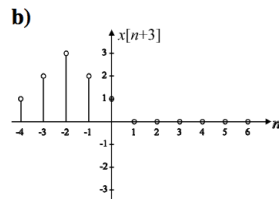
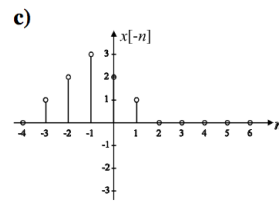
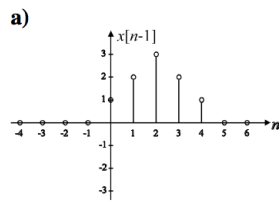
(b)  $y(n) = [9, 12, 10, 4, 1]$

(c)  $y(n) = [3, 8, 14, 8, 3]$

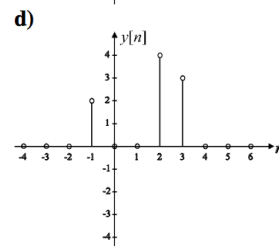
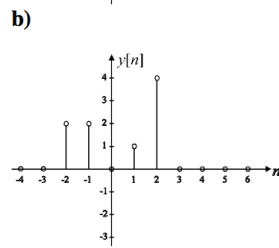
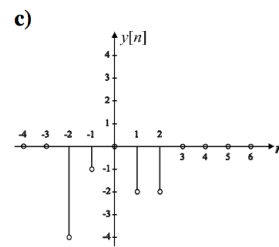
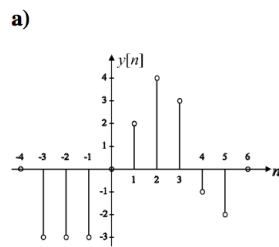
- 7



- 8



• 9

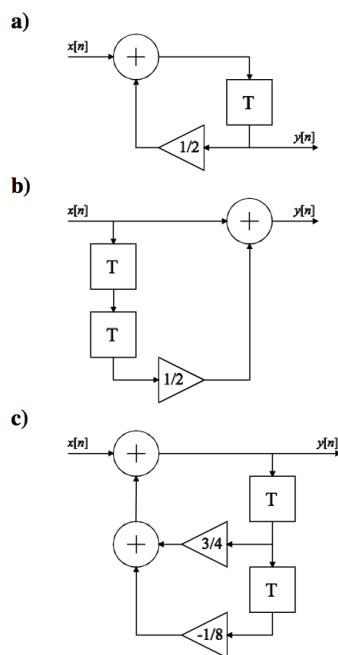


• 10

(a)  $y[n] - \frac{1}{2}y[n-1] = x[n]$

(b)  $y[n] = x[n] + \frac{1}{2}x[n-1]$

• 11



- 12

Z-transformera (enl. tabell i formelsamling,  $a = \frac{1}{4}$ ):  $h(n) = (\frac{1}{4})^n u(n) \rightarrow H(z) = \frac{z}{z-a}$ ,  
 $x(n) = u(n) \rightarrow X(z) = \frac{z}{z-1}$

Faltning i tid=multiplikation i z-domänen:  $Y(z) = X(z)H(z) = \frac{z^2}{(z-a)(z-1)}$

Partialbråksuppdelning (se formelsamling) - ansätt  $Y(z) = \frac{Az}{z-1} + \frac{Bz}{z-a}$  och sätt på samma bråkstreck:  $\frac{Az(z-a)+Bz(z-1)}{(z-1)(z-a)} = \frac{(A+B)z^2-(Aa+B)z}{(z-1)(z-a)}$

Identifiera koefficienterna framför  $z^2$  och  $z$  i täljaren i de två uttrycken för  $Y(z)$ :  $z^2$  ger  $A+B=1$ ,  $z$  ger  $Aa+B=1$ . Lösning av ekv.syst. ger  $A = \frac{-1}{a-1} = \frac{4}{3}$ ,  $B = \frac{a}{a-1} = -\frac{1}{3}$ .

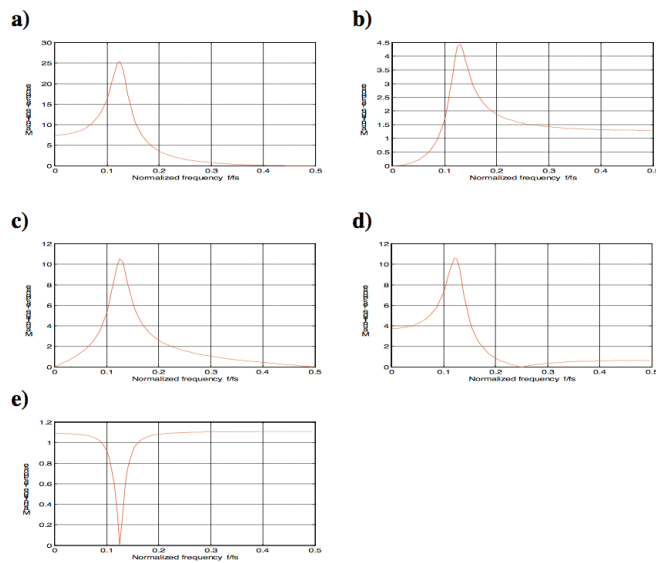
Nu har vi  $Y(z) = \frac{\frac{4}{3}}{(1-z^{-1})} - \frac{\frac{1}{3}}{(1-\frac{1}{4}z^{-1})}$

Invers Z-transform (enl. f.s.) ger slutligen

$$y(n) = (\frac{4}{3} - \frac{1}{3}(\frac{1}{4})^n) u(n)$$

- 13



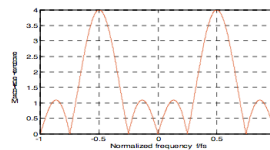


- 14 A - 3; B - 1; C - 2; D - 4

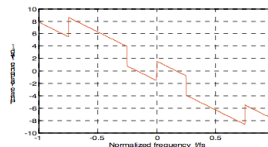
- 15

(a)  $p_{1,2,3} = 0; n_{1,2} = \pm j; n_3 = 1$

(b)  $|H[e^{j\omega}]| = 2 \cdot |\sin(\frac{3\omega}{2}) - \sin(\frac{\omega}{2})|$



(c)  $\arg\{H[e^{j\omega}]\} = \frac{\pi}{2} - \frac{3\omega}{2} \text{ (+ ev. } \pi)$



(d)  $\omega = \pm \frac{\pi}{2}; \omega = 0$

(e) Högpasfilter

- 16

```
% Impulse response has only two values
h = [0.5, 0.5];
% Input to the system
x1 = [1, 0, 1, 0, 1, 0];
% for LTI systems, output is
% convolution of impulse response and input
% Note: length(y) = length(x) + length(h) - 1
y1 = conv(h, x1);
disp('output of x1 is: ');disp(y1);
```

```
x2 = [1, 1, 1, 2, 2, 2];
y2 = conv(h, x2);
disp('output of x2 is: ');disp(y2);
```

```
x3 = [1, 2, 3];
y3 = conv(h, x3);
disp('output of x3 is: ');disp(y3);
```

```
output of x1 is:
```

```
0.5000    0.5000    0.5000    0.5000    0.5000    0.5000         0
```

```
output of x2 is:
```

```
0.5000    1.0000    1.0000    1.5000    2.0000    2.0000    1.0000
```

```
output of x3 is:
```

```
0.5000    1.5000    2.5000    1.5000
```

- 17

Nollställen  $n_1, n_2$  och poler  $p_1, p_2$  ger överföringsfunktionen

$$H(z) = \frac{(z - n_1)(z - n_2)}{(z - p_1)(z - p_2)} = \frac{z^2 - z(n_1 + n_2) + n_1n_2}{z^2 - z(p_1 + p_2) + p_1p_2}$$

Täljar- resp nämnarpolynomen representeras av vektorerna  $B = [1 \quad -(n_1 + n_2) \quad n_1n_2]$  (täljare) och  $A = [1 \quad -(p_1 + p_2) \quad p_1p_2]$  (nämnare).

Alternativt kan man (om är bekväm) använda matlab-funktionen `zp2tf()` för att göra omvandlingen från poler/nollställen till B/A-vektorer.

```
% poles
p = [0.9*exp(-j*pi/4); 0.9*exp(j*pi/4)];
% zeros
z = [-1; -1];
% first convert the poles and zeros into
% transfer equation using 'zp2tf' matlab function
[b, a] = zp2tf(z, p, 1);
% use 'fvtool' function to plot the magnitude response
% Note: 'fvtool' shows the magnitude response in log domain (dB)
fvtool(b, a);
```

```
% similarly for other questions..
```

```
z = [1; 1];
```

```
[b, a] = zp2tf(z, p, 1);
fvtool(b, a);
```

```
z = [1; -1];
[b, a] = zp2tf(z, p, 1);
fvtool(b, a);
```

```
z = [j; -j];
[b, a] = zp2tf(z, p, 1);
fvtool(b, a);
```

```
z = [exp(-j*pi/4); exp(j*pi/4)];
[b, a] = zp2tf(z, p, 1);
fvtool(b, a);
```

## Z-transform

- 1

(a)  $H[z] = \frac{2z^2+5z+2}{z^2+\frac{1}{2}z-\frac{3}{2}}$

(b)  $p_{1,2} = \begin{cases} 1 \\ -\frac{3}{2} \end{cases} \Rightarrow$  kreteren inte stabil

- 2  $H[z] = \frac{2z^{-1} - \frac{5}{6}z^{-2}}{1 - \frac{5}{6}z^{-1} + \frac{1}{6}z^{-2}}$ ;  $p_{1,2} = \begin{cases} \frac{1}{2} \\ \frac{1}{3} \end{cases} \Rightarrow$  kreteren stabil

- 3

(a)  $X[z] = 1, \forall z$

(b)  $X[z] = \frac{z}{z-1}, |z| > 1$

(c)  $X[z] = \frac{z^{-k+1}}{z-1}, |z| > 1$

(d)  $X[z] = \frac{z}{z-\alpha}, |z| > \alpha$

- 4

(a)  $Y[z] = z^{-m} \cdot X[z]$

(b)  $Y[z] = X\left[\frac{z}{\alpha}\right]$

- 5

(a)  $X[z] = \frac{z}{z-\frac{1}{2}}, |z| > \frac{1}{2}$

(b)  $X[z] = \frac{z^2-\frac{1}{2}z}{z^2-z+\frac{1}{2}}, |z| > \frac{1}{\sqrt{2}}$

(c)  $X[z] = \frac{z}{z^2+1}, |z| > 1$

- 6

(a)  $x[n] = \left(\frac{1}{3}\right)^n \cdot u[n]$

$$(b) x[n] = \frac{1}{2}(1 + (-1)^n) \cdot u[n] = \cos^2\left(\frac{\pi}{2}n\right) \cdot u[n]$$

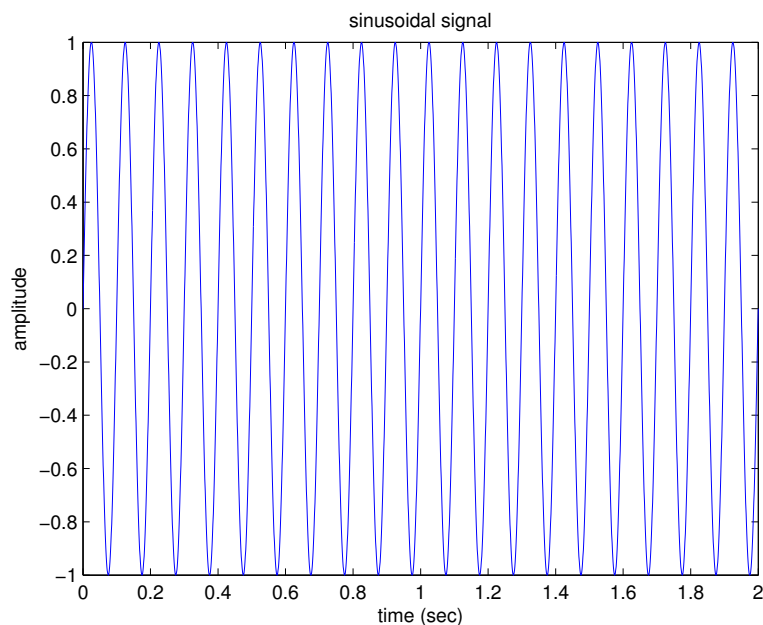
• 7

$$(a) y[n] = \left(2 - \left(\frac{1}{2}\right)^n\right) \cdot u[n]$$

$$(b) y[n] = \left(3 \cdot \left(\frac{1}{2}\right)^n - 2 \cdot \left(\frac{1}{3}\right)^n\right) \cdot u[n]$$

• 8

```
% amplitude
A = 1;
% fundamental frequency
f = 10; %10Hz
% sampling frequency
fs = 1000; % 1000 Hz
% duration, equally spaced by sampling frequency
t = 0: 1/fs: 2; % 2 seconds
% sinusoidal equation
y = sin(2*pi*f*t);
% plot the sinusoidal signal
figure();
plot(t,y)
title('sinusoidal signal')
xlabel('time (sec)')
ylabel('amplitude')
```



• 9

```
% Analysis of Z-Transforms
% Definition of numerators and denominator coefficients
num=[0, 1, 0.5];
```

```

den=[1, 0.6, 0.8];
% Conversion from rational to Factored form
[z,p]=tf2zp(num,den);
disp('Zeros are at');disp(z);
disp('Poles are at');disp(p);
% Pole Zero Map using numerator and denominator coefficients
zplane(num,den)

```

```

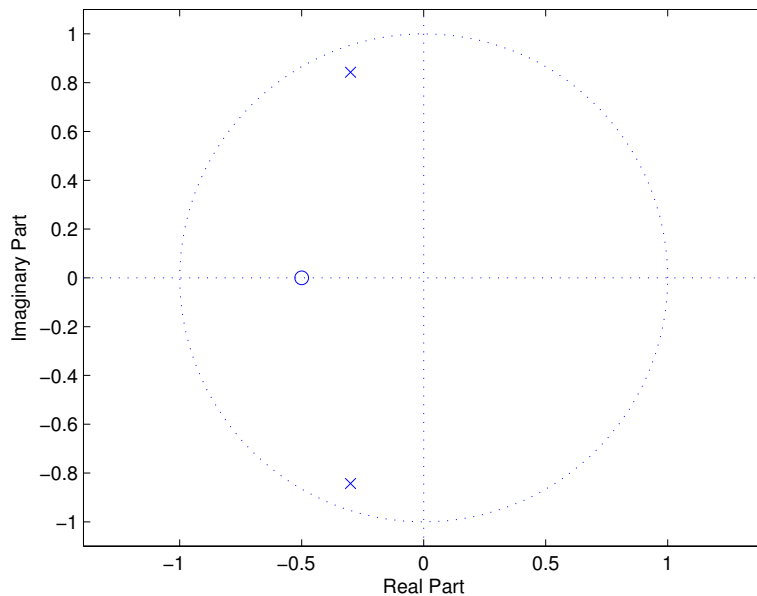
Zeros are at
    -0.5000

```

```

Poles are at
    -0.3000 + 0.8426i
    -0.3000 - 0.8426i

```



- 10

För att beräkna impulssvaret numeriskt: Använd 'filter(B,A,X)' där  $B$  och  $A$  är vektorer innehållande täljarens och nämnarens koefficienter och  $X$  är insignalen till filtret, vilken - om vi söker impulssvaret - givetvis ska vara en impuls.

(a)  $X[z] = \frac{z}{z - \frac{1}{3}}$  ger  $B=[1 \ 0]; A=[1 \ -1/3]$

(b)  $X[z] = \frac{z^2}{z^2 - 1}$  ger  $B=[1 \ 0 \ 0]; A=[1 \ 0 \ -1]$

```

% Create input signal: an impulse vector of length 10
X = [1 zeros(1,9)]
% These are the filter coefficients from a) & b), uncomment one
B = [1 0]; A = [1 -1/3];

```

```

% B = [1 0 0]; A = [1 0 -1];
% Filter X to obtain the impulse response
Y = filter(B,A,X);
% Plot the result using 'balls-on-sticks'
stem(Y);

```

## Fourierserier

- 1
  - (a) 2
  - (b)  $\frac{3}{4}$
- 2 Jämna: (a), (c) och (f). Udda: (b), (d) och (e). Alternativ (g) är varken en jämn eller en udda funktion.

- 3
  - (a)  $a_k = 0$  och

$$b_k = \begin{cases} \frac{4}{n\pi} & (\text{udda } n) \\ 0 & (\text{jämna } n) \end{cases}$$

- (b)  $a_k = 0$  eftersom fyrkantsvågen är en udda funktion, givet att inte fasen ändras
- (c)

$$\begin{aligned}
 f(t) &= \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \\
 &= \sum_{k=-\infty}^{\infty} c_k (\cos k\omega_0 t + j \sin k\omega_0 t) \\
 &= c_0 + \sum_{k=1}^{\infty} ((c_k + c_{-k}) \cos k\omega_0 t + j(c_k - c_{-k}) \sin k\omega_0 t) \\
 &= \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos k\omega_0 t + b_k \sin k\omega_0 t)
 \end{aligned}$$

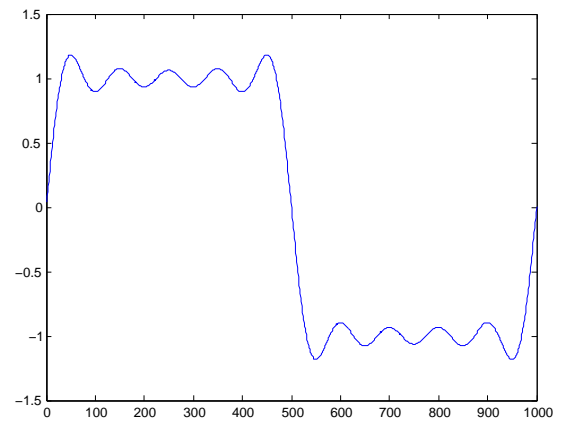
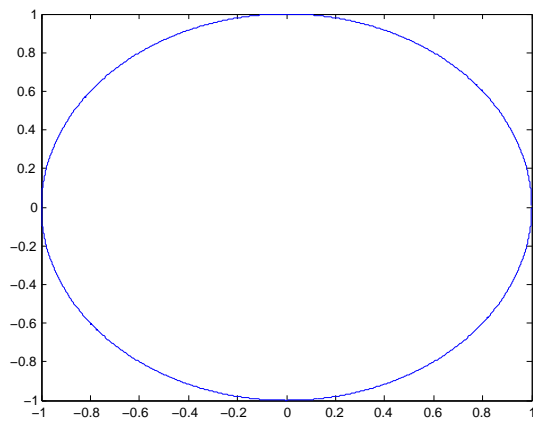
Där  $a_k = c_k + c_{-k}$  och  $b_k = j(c_k - c_{-k})$

- 4  $\frac{\pi}{4}$
- 5
  - (a)  $\frac{1}{n}$  (se Smith, Fourier Series, kap.13, s. 255-260)
  - (b)  $\frac{1}{n^2}$
  - (c) Triangelvågens Fourierkoefficienter går mot noll med högre hastighet än fyrkantsvågens Fourierkoefficienter. Därför klarar triangelvågen bättre av en smal bandbredd.
- 6

```

nmax = 1000;
N = 0:1000;
omega0=2*pi/nmax;
plot(exp(i*omega0*N))

```



```
figure; %Ny figur
Y1 = -2i/pi*(exp(1i*omega0*N) - exp(-1i*omega0*N));
Y3 = -2i/(3*pi)*(exp(3i*omega0*N) - exp(-3i*omega0*N));
Y5 = -2i/(5*pi)*(exp(5i*omega0*N) - exp(-5i*omega0*N));
plot(N,Y1+Y3+Y5)
```

# Bilaga B

## Matlabövningar

### Hjälpssystemet

Börja med att bekanta dig med matlabs inbyggda hjälpsystem, här kan man få information om inbyggda funktioner, nyckelord och koncept. Prova att skriva

```
help funktion
```

för några olika funktioner, t.ex. `diag`, `exp`, `magic` och `i`.

Du kommer att behöva kolla upp flera funktioner själv i uppgifterna nedan, men ofta får du tips om relevanta funktionsnamn.

Ibland vet man ju dock inte ens vad funktionen man söker heter, eller ens om den finns, då kan man använda

```
lookfor nyckelord.
```

Då letar matlab i alla hjälptexter efter nyckelordet.

### Matriser och arrayer

Läs på i matlabhäftet om matriser och arrayer. Prova funktionerna vart eftersom du läser om dem. Gör sedan nedanstående övningar.

- skapa en s.k. magisk kvadrat med `a=magic(7)` och verifiera att summan i alla rader och kolumner är konstant (*tips: sum*).
- plocka ut valfri rad respektive kolumn ur `a` (*tips: använd "kolon-notation"*)
- skriv ett uttryck som returnerar en ny, mindre matris genom att ta varannan kolumn och varannan rad från `a`
- vänd matrisen runt den horisontella respektive den vertikala axeln! (*tips: kan ses som en variant på ovanstående. Alternativt: lookfor flip...*)
- vänd matrisen längs huvuddiagonalen (*tips: kallas även transponat; lookfor transpose.*)
- skapa en ny matris `b` av samma storlek som `a`, som där varje element är kvadraten av motsvarande element i `a`.
- gör så att matrisens minsta elementet blir 0 och det största blir 1, genom att addera en konstant och applicera en skalfaktor (*tips: min,max*)
- skapa en ny matris `c` av samma storlek som `a`, där varje element är 1 om motsvarande element i `a` är större än 25, annars 0. (*tips: relop*)



- skriv ett uttryck som plockar ut alla tal som är delbara med 3 ur en vektor eller matris (*tips: find, mod*)
- skriv om nedanstående kod så att loopar undviks:

```

a = [1 .5 .3 .4];
w = [1 2 3 4];
y = 0;
for i = 1:length(a)
    y = y + a(i) * w(i);
end

```

- generera en vektor  $x$  av längden 1000 med simulerade resultat från en sexsidig tärning. Verifiera sedan att tärningen är välbalanserad! (*tips: rand, round, find*)
- Generera en mängd punkter med  $10 * \text{rand}(2, 10)$ . Sortera sedan dessa efter deras avstånd till origo.

## Plottar, signaler och bilder

Läs om grafikmöjligheterna i matlab under avsnitt 5 i *Crash course in MATLAB*. Här kommer vi att koncentrera oss på `plot()`.

### Enkla plottar

OBS!  $n$  betecknar ett heltal i nedanstående uppgifter.

- plotta  $y(n) = |n|$  när  $-10 \leq n \leq 10$
- plotta "gauss-klockan"

$$y(n) = e^{-\left(\frac{n}{5}\right)^2}$$

med *gröna cirklar* när  $-10 \leq n \leq 10$

- plotta  $x_1(n) = \cos \omega_1 n$  när  $\omega_1 = \frac{\pi}{100}$  och  $0 \leq n \leq 1000$
- plotta  $x_1(n)$  ovan tillsammans med

$$x_2(n) = \sin \omega_2 n + \sin(\omega_1 + \omega_2)n$$

när  $\omega_2 = \frac{\pi}{10}$ ,  $\omega_1 = \frac{\pi}{100}$  och  $0 \leq n \leq 1000$ .

## Ljud

För att läsa in en ljudfil i matlab använder man `[X,fs] = audioread('ljud.wav')`. Detta ger en vektor med sampelvärden i  $X$  och samplingsfrekvensen i  $fs$ . För att spela upp ljudet skriver man `sound(X,fs)`.

- Prova att generera några olika vektorer och spela upp som ljud. T.ex. en sekund vitt brus fås genom `sound(rand(16000,1),16000)`. Testa att skapa toner av olika frekvens och längd med sinusfunktionen.
- Läs in ett ljud och spela upp det i högtalaren.
- Gör så att ljudet spelas baklänges.
- Gör en tremoloeffekt genom att multiplicera ljudvektorn med en vektor av samma längd som innehåller en långsamt varierande sinussvängning.

## Faltning

Implementera så kalla "faltning" i en dimension.

- Skapa en datamängd i en dimension som du kan falta med din nya kernel  
`X = [zeros(1,3) ones(1,3) zeros(1,3)]`
- En enkel kärna kan skapas med uttrycket `kernel = [1 1 1] ./ 3`
- Skapa en ny vektor  $Y$ , där varje punkt  $Y(i)$  är en summa av produkten av  $X(i-2 : i)$  och `kernel`. *tips: Detta görs lämpligen med hjälp av en for-loop och funktionen sum*
- Plotta  $X$  och  $Y$ .

## Bilder

Bilder i representeras i matlab som matriser - varje pixel motsvarar ett matriselement. För att läsa in en bild i matlab använder man funktionen `imread()`. Den läser de flesta format och returnerar en matris av pixelvärden mellan 0 och 255. Det finns dock två problem man kan stöta på: vissa format returnerar matriser av heltalstyp och dessa måste konverteras till `double` för att fungera som vanliga matlab-matriser. `I=double(imread('minbild.gif'))` gör detta.

Färgbilder returneras som en  $M \times N \times 3$ -matris. Dessa kan konverteras till en vanlig 2D-gråskalematis genom att ta medelvärdet av de tre färgplanen. `mean(imread('minbild.gif'),3)` tar hand om detta.

Sammantaget innebär detta att följande rad troligtvis är bra om man vill kunna läsa in många olika typer av bildfiler:

```
I = mean(double(imread('minbild.gif')),3);
```

För att titta på en bild använder man kommandona `image` resp. `imagesc` - den förra förväntar sig pixelvärden mellan 0 och 255 medan den senare skalar om bildens gråskalevärden automatiskt, vilket ofta är praktiskt.

Du behöver även skriva `axis image` (säkerställer att höjd/bredd-förhållandet är bra) samt `colormap(gray)` (gör att matrisen tolkas som en gråskalebild).

Du ska nu pröva dessa funktioner och även behandla bilden på olika enkla sätt med hjälp av matlabs vanliga matrishantering. Du får använda valfri bildfil som inte är för stor (ca 500x500 pixlar kan vara lagom). Det finns en lämplig testbild åtkomlig från kurshemsidan. Nedanstående

punkter kan ses som inspiration - du måste inte göra alla och du får gärna prova andra operationer på bildmatrisen.

- Läs in din bild i matlab som en gråskale-matris och visa den på skärmen.
- Invertera bilden, så att mörkt blir ljust och vice versa!
- Beskär bilden, så du bara tittar på ett område i mitten *tips: använd matlabs kolon-notation för matrisindexering*
- Vänd bilden upp-och-ner (*tips: går att göra på samma sätt som föregående uppgift!*)
- Skala ner bilden genom att plocka bort varannan rad och varannan kolumn! Hur blir resultatet? Vad är problemet med denna metod för bildskalning?
- Gör bilden brusig genom att lägga till en slumpmässiga matris (med lämplig skalfaktor)

## Funktioner

Läs om funktioner i avsnitt 3 i *Crash course in MATLAB*.

Funktioner är ett utmärkt sätt att skapa återanvändbara algoritmer, och bringa reda i sin matlab-kod. Gör det till en vana att använda funktioner med vettiga in- och utparametrar så kommer dina matlab-program bli så mycket lättare att förstå.

Nu ska du göra om några av uppgifterna ovan till funktioner. Glöm inte att skriva hjälp-text för funktionerna (de kommentar-rader som följer efter funktionshuvudet). Testa sedan att skriva `help din funktion!`

- I tredje uppgiften i avsnitt 2 fick du reducera en matris genom att ta varannan rad och varannan kolumn. Skriv en generell funktion för detta! Inparametrar bör vara en matris, samt två heltal som anger vertikal och horisontell reduktionsfaktor (dvs 2 ger varannan rad, 3 var tredje etc.). Funktionen ska returnera den reducerade matrisen.
- Skriv en funktion som beräknar fourier-serie approximationen av en fyrkantvåg, med godtyckligt många termer! Generalisera utifrån ekvationen i uppgift 3.2 eller titta i formelsamlingen på kurshemsidan. Inparametrar ska vara sekvenslängd  $N$ , grundfrekvens  $\omega_0$ , samt  $K$  - den högsta frekvens som får ingå uttryckt i multiplar av grundfrekvensen. Returnvärdet ska vara den approximerade fyrkanvågen. Detta innebär att med inparametrarna  $N = 1000, \omega_0 = 2\pi/1000$  samt  $K = 5$  bör du erhålla samma resultat som i summan i uppgift 3.2.

Testa din funktion genom att plotta resultatet för olika värden på  $K$ !

- I andra uppgiften implementerade du faltning. Skriv en generell funktion som tar en endimensionell vektor `X` av arbiträr längd och en vektor `kernel` av arbiträr längd som inparametrar och returnerar resultatet av att falta `X` med `kernel`.

Experimentera gärna med olika kärnor och med data som exempelvis ljudfilen från avsnitt ??.

Som exempel, gör en kärna som består av en sinuskurva:

```
kernel = sin(0:0.01:2*pi);  
kernel = kernel./sum(kernel);
```

*tips: kom ihåg att normalisera kerneln så att den summeras till 1*

# Bilaga C

## Pythonövningar

### Installation

Börja med att se till så att du har python samt pip installerat på ditt system. Exemplena kommer att vara skrivna med Python 3 syntax och i de flesta fall fungera med Python 2, men vissa avvikelser kan förekomma.

Därefter kan du installera de bibliotek som kommer att användas under denna övning med följande kommando i terminalen:

```
pip install numpy matplotlib scipy simpleaudio pillow
```

Där numpy är ett bibliotek för bland annat matrisoperationer, matplotlib ett bibliotek för att visa plottar, scipy innehåller många olika användbara funktioner, simpleaudio som används för att återspela ljud samt pillow som behövs för läsa in bilder.

Om andra bibliotek kommer att behövas kommer instruktionerna att meddela detta och hur man i så fall installerar dem.

För att använda ovanstående bibliotek importerar du dem i dina python script. Många bibliotek brukar importera biblioteken med ett kortare name med hjälp av tillägget `as` till importuttrycket. Numpy och Matplotlib importeras oftast med följande rader (överst i python-skriptet):

```
import numpy as np
import matplotlib.pyplot as plt
```

Detta innebär att de blir tillgängliga under `np` samt `plt`. De flesta exempel man hittar utgår från att man importerat dem på detta sätt.

### Dokumentation

För dokumentation rekommenderas att använda sig av numpy och scipys onlinedokumentation <https://docs.scipy.org/doc/numpy-1.13.0/search.html> respektive <https://docs.scipy.org/doc/scipy/reference/search.html>. Här kan man söka efter olika funktionsnamn, såsom `np.array`, `np.dot` och `scipy.misc.imread` eller nyckelord för att hitta en lämplig funktion för den operation man är ute efter.

Ofta finns det ett par exempel på hur man kan använda funktionen i slutet av sidan.

Du kommer att behöva kolla upp flera funktioner själv i uppgifterna nedan, men ofta får du tips om relevanta funktionsnamn.

## Matriser och arrayer

Läs på i pythonhäftet om matriser och arrayer. Prova funktionerna vart eftersom du läser om dem. Gör sedan nedanstående övningar.

- skapa en 7x7 matris med slumpmässigt utvalda siffror mellan 0-100 (*tips: np.random.randint*).
- plocka ut valfri rad respektive kolumn ur  $a$  (*tips: använd "kolon-notation"*)
- skriv ett uttryck som returnerar en ny mindre matris genom att ta varannan kolumn och varannan rad från  $a$  (*tips: använd "kolon-notation"*)
- vänd matrisen runt den horisontella respektive den vertikala axeln! (*tips: np.flip*)
- vänd matrisen längs huvuddiagonalen (*tips: np.transpose*)
- skapa en ny matris  $b$  av samma storlek som  $a$ , där varje element är kvadraten av motsvarande element i  $a$ .
- gör så att matrisens minsta elementet blir 0 och det största blir 1, genom att addera en konstant och applicera en skalfaktor (*tips: np.min, np.max*)
- skapa en ny matris  $c$  av samma storlek som  $a$ , där varje element är 1 om motsvarande element i  $a$  är större än 25, annars 0. (*tips: np.where*)
- skriv ett uttryck som plockar ut alla tal som är delbara med 3 ur en vektor eller matris (*tips: np.extract*)
- skriv om nedanstående kod så att loopar undviks (*tips: np.dot*):

```
a = np.array([1, .5, .3, .4])
w = np.array([1, 2, 3, 4])
y = 0
for i in range(len(a)):
    y += a[i] * w[i]
```

- generera en vektor  $x$  av längden 1000 med simulerade resultat från en sexsidig tärning. Verifiera sedan att tärningen är välbalanserad! (*tips: np.random.randint, np.extract*)
- Generera en mängd punkter med  $10 * np.random.random((2, 10))$ . Sortera sedan dessa efter deras avstånd till origo. (*tips: np.sort*)

## Plottar, signaler och bilder

För att plotta saker kan man använda `matplotlib`. Exempelvis på följande sätt:

```
import matplotlib.pyplot as plt
X = np.random.random((2, 10))
plt.plot(X[0])
plt.plot(X[1], color='red')
plt.show()
```

`show` används för att visa själva plotten, och för att inte programmet ska avslutas direkt. Istället väntar programmet på att användaren stänger ner fönstret med plotten.

Man kan åstadkomma flera plottar i ett fönster på följande sätt:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(1, 2) # corresponds to 1 row, and 2 columns
X = np.random.random((3, 10))
ax[0].plot(X[0], marker='o')
ax[0].plot(X[1], color='red')
ax[1].plot(X[2])
plt.show()
```

## Enkla plottar

OBS!  $n$  betecknar ett heltal i nedanstående uppgifter.

- plotta  $y(n) = |n|$  när  $-10 \leq n \leq 10$
- plotta “gauss-klockan”

$$y(n) = e^{-\left(\frac{n}{5}\right)^2}$$

med gröna cirklar när  $-10 \leq n \leq 10$

- plotta  $x_1(n) = \cos \omega_1 n$  när  $\omega_1 = \frac{\pi}{100}$  och  $0 \leq n \leq 1000$
- plotta  $x_1(n)$  ovan tillsammans med

$$x_2(n) = \sin \omega_2 n + \sin(\omega_1 + \omega_2)n$$

när  $\omega_2 = \frac{\pi}{10}$ ,  $\omega_1 = \frac{\pi}{100}$  och  $0 \leq n \leq 1000$ .

## Ljud

För att läsa in en ljudfil i python kan man använda

```
fs, X = scipy.io.wavfile.read('ljud.wav').
```

OBS! `scipy.io.wavfile.read` kan inte läsa 24-bits ljud, använd i så fall bibliotek som t.ex. `PySoundFile`.

Detta ger en vektor med sampelvärden i  $X$  och samplingsfrekvensen i  $fs$ . För att spela upp ljudet kan man använda exempelvis `sounddevice`.

```
import scipy.io.wavfile
import simpleaudio as sa
fs, X = scipy.io.wavfile.read('ljud.wav')
sa.play_buffer(X, X.ndim, 2, fs)
input('press enter when ready')
```

- Prova att generera några olika vektorer och spela upp som ljud. T.ex. en sekund vitt brus fås genom `sa.play_buffer(np.random.random((16000,1)), 2, 2, 16000)`. Testa att skapa toner av olika frekvens och längd med sinusfunktionen.

- Läs in ett ljud och spela upp det i högtalaren.
- Gör så att ljudet spelas baklänges.
- Gör en tremoloeffekt genom att multiplicera ljudvektorn med en vektor av samma längd som innehåller en långsamt varierande sinussvängning.

## Faltning

Implementera så kalla "faltning" i en dimension.

- Skapa en datamängd i en dimension som du kan falta med din nya kernel  
`X = np.concatenate([np.zeros(3), np.ones(3), np.zeros(3)])`
- En enkel kärna kan skapas med uttrycket `kernel = np.array([1, 1, 1]) / 3`
- Skapa en ny vektor Y, där varje punkt Y(i) är en summa av produkten av X[i-2 : i] och kernel. *tips: Detta görs lämpligen med hjälp av en for-loop och funktionen np.sum*
- Plotta X och Y.

## Bilder

Bilder kan representeras som matriser i python - varje pixel motsvarar ett matriselement. För att läsa in en bild i python kan man använda funktionen `I = scipy.misc.imread('minblid.gif')`. Den läser de flesta format och returnerar en matris av pixelvärden mellan 0 och 255. Färgbilder returneras som en  $M \times N \times 3$ -matris. Dessa kan konverteras till en vanlig 2D-gråskalematis genom att ta medelvärdet av de tre färgplanen.

`np.mean(scipy.misc.imread('minblid.gif'), axis=2)` tar hand om detta.

För att titta på en bild kan man använda kommandona `plt.imshow(I)` - den förväntar sig pixelvärden mellan 0 och 255. Om man vill visa en gråskalebild kan man lägga till keyword argumentet `cmap, plt.imshow(I, cmap='gray')`

Du ska nu pröva dessa funktioner och även behandla bilden på olika enkla sätt med hjälp av numpys vanliga matrishantering. Du får använda valfri bildfil som inte är för stor (ca 500x500 pixlar kan vara lagom). Det finns en lämplig testbild åtkomlig från kurshemsidan. Nedanstående punkter kan ses som inspiration - du måste inte göra alla och du får gärna prova andra operationer på bildmatrisen.

- Läs in din bild i python som en gråskale-matris och visa den på skärmen.
- Invertera bilden, så att mörkt blir ljust och vice versa!
- Beskär bilden, så att du bara tittar på ett område i mitten *tips: använd numpys kolonnotation för matrisindexering*
- Vänd bilden upp-och-ner (*tips: går att göra på samma sätt som föregående uppgift!*)
- Skala ner bilden genom att plocka bort varannan rad och varannan kolumn! Hur blir resultatet? Vad är problemet med denna metod för bildskalning?
- Gör bilden brusig genom att lägga till en slumpmässiga matris (med lämplig skalfaktor)

## Funktioner

Funktioner är ett utmärkt sätt att skapa återanvändbara algoritmer, och bringa reda i sin python-kod. Gör det till en vana att använda funktioner med vettiga in- och utparametrar så kommer dina python-program bli så mycket lättare att förstå.

Nu ska du göra om några av uppgifterna ovan till funktioner. Glöm inte att skriva hjälp-text för funktionerna (de kommentar-rader som följer efter funktionshuvudet).

- I tredje uppgiften i avsnitt 2 fick du reducera en matris genom att ta varannan rad och varannan kolumn. Skriv en generell funktion för detta! Inparametrar bör vara en matris, samt två heltal som anger vertikal och horisontell reduktionsfaktor (dvs 2 ger varannan rad, 3 var tredje etc.). Funktionen ska returnera den reducerade matrisen.
- Skriv en funktion som beräknar fourier-serie approximationen av en fyrkantvåg, med godtyckligt många termer! Generalisera utifrån ekvationen i uppgift 3.2 eller titta i formelsamlingen på kurshemsidan. Inparametrar ska vara sekvenslängd  $N$ , grundfrekvens  $\omega_0$ , samt  $K$  - den högsta frekvens som får ingå uttryckt i multiplar av grundfrekvensen. Returvärdet ska vara den approximerade fyrkantvågen. Detta innebär att med inparametrarna  $N = 1000, \omega_0 = 2\pi/1000$  samt  $K = 5$  bör du erhålla samma resultat som i summan i uppgift 3.2.

Testa din funktion genom att plotta resultatet för olika värden på  $K$ !

- I andra uppgiften implementerade du faltning. Skriv en generell funktion som tar en endimensionell vektor  $X$  av arbiträr längd och en vektor `kernel` av arbiträr längd som inparametrar och returnerar resultatet av att falta  $X$  med `kernel`.

Experimentera gärna med olika kärnor och med data som exempelvis ljudfilen från avsnitt ??.

Som exempel, gör en kärna som består av en sinuskurva:

```
import numpy as np
kernel = np.sin(np.arange(0, 2 * math.pi, 0.01))
kernel = kernel / np.sum(kernel)
```

*tips: kom ihåg att normalisera kerneln så att den summeras till 1*



# Bilaga D

## Formelblad

### Komplexa tal

$$x + jy = re^{j\phi} = r \cos \phi + jr \sin \phi$$

$$\phi = \arctan \frac{y}{x} \quad r = \sqrt{x^2 + y^2}$$

Eulers formler

$$\sin \phi = \frac{e^{j\phi} - e^{-j\phi}}{2j} \quad \cos \phi = \frac{e^{j\phi} + e^{-j\phi}}{2}$$

$$e^{j\pi} + 1 = 0$$

Komplexa räknerelger

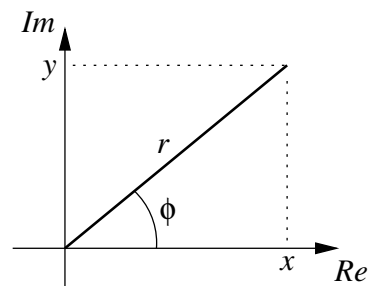
$$z_1 = x_1 + jy_1 = r_1 e^{j\phi_1} \quad z_2 = x_2 + jy_2 = r_2 e^{j\phi_2}$$

$$z_1 + z_2 = x_1 + x_2 + j(y_1 + y_2)$$

$$z_1 - z_2 = x_1 - x_2 + j(y_1 - y_2)$$

$$z_1 z_2 = (x_1 x_2 - y_1 y_2) + j(x_1 y_2 + x_2 y_1) = r_1 r_2 e^{j(\phi_1 + \phi_2)}$$

$$\frac{z_1}{z_2} = \frac{(x_1 x_2 + y_1 y_2) + j(x_2 y_1 - x_1 y_2)}{x_2^2 + y_2^2} = \frac{r_1}{r_2} e^{j(\phi_1 - \phi_2)}$$



### Trigonometriska identiteter

$$\sin^2 \alpha + \cos^2 \alpha = 1$$

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha}$$

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta \quad \cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta$$

$$\sin^2 \alpha = \frac{1 - \cos 2\alpha}{2} \quad \cos^2 \alpha = \frac{1 + \cos 2\alpha}{2}$$

## Deriveringsregler

$$\begin{aligned}[g(f(x))] &= g'(f(x)) f'(x) \\ [g(x)h(x)] &= g'(x)h(x) + h'(x)g(x) \\ \left[\frac{g(x)}{h(x)}\right] &= \frac{g'(x)h(x) - h'(x)g(x)}{h^2(x)}\end{aligned}$$

## Decibel

Två signaler med amplituderna  $A_1$  och  $A_2$  har dB-förhållandet  $20 \log(A_2/A_1)$

Två signaler med effekterna  $P_1$  och  $P_2$  har dB-förhållandet  $10 \log(P_2/P_1)$

### Samplingsteoremet

En signal som samplas vid frekvensen  $f_s$  kan rekonstrueras exakt om signalen endast innehåller information under halva samplingsfrekvensen, dvs om signalen är bandbegränsad till  $f_s/2$

## Faltning

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

*Faltning i 2D*

$$g(x, y) = f(x, y) * h(x, y) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} h(s, t)f(x-s, y-t)$$

### Faltningsteoremet

Faltning av två signaler i tids-/spatialdomänen motsvarar multiplikation i frekvensdomänen. Faltning av två signaler i frekvensdomänen motsvarar multiplikation i tids-/spatialdomänen.

## Filter

Ett tidsdiskret linjärt filter som beskrivs av differensekvationen

$$\begin{aligned}y(n) &= b_0x(n) + b_1x(n-1) + b_2x(n-2) + \dots + b_Nx(n-N) \\ &\quad - a_1y(n-1) - a_2y(n-2) \dots - a_My(n-M)\end{aligned}$$

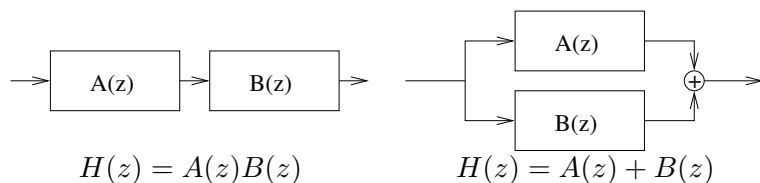
har överföringsfunktionen

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Nz^{-N}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_Mz^{-M}}$$

och frekvenssvaret

$$H(\omega) = \frac{b_0 + b_1e^{-j\omega} + b_2e^{-j2\omega} + \dots + b_Ne^{-jN\omega}}{1 + a_1e^{-j\omega} + a_2e^{-j2\omega} + \dots + a_Me^{-jM\omega}}$$

## Kaskad- och parallellkoppling



## Geometrisk serie

$$\sum_{k=0}^n a^k = 1 + a + a^2 + \dots + a^n = \frac{a^{n+1} - 1}{a - 1}$$

## Fourierserier

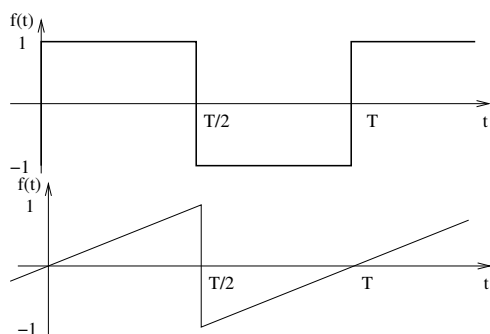
En tidskontinuerlig periodisk signal  $f(t) = f(t+T)$  där  $T = \frac{2\pi}{\omega_0}$  kan skrivas som en viktad summa av komplexa fasvektorer

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$$

där vikterna  $c_k$  kan beräknas ur den ursprungliga signalen med integralen

$$c_k = \frac{1}{T} \int_0^T f(t) e^{-jk\omega_0 t} dt$$

## Några vanliga fourierserier



$$c_k = \begin{cases} -\frac{2j}{k\pi} & \text{om } k = 1, 3, 5, \dots \\ 0 & \text{annars} \end{cases}$$

$$c_k = \begin{cases} -\frac{j}{k\pi} & \text{om } k = 1, 3, 5, \dots \\ \frac{j}{k\pi} & \text{annars} \end{cases}$$

## Z-transform

$$X(z) = \sum_{n=0}^{\infty} x(n) z^{-n}$$

## Några användbara Z-transformpar

$x(n)$	$X(z)$	Anm.
$\delta(n)$	1	$\delta(n) = \begin{cases} 1 & , n = 0 \\ 0 & , n \neq 0 \end{cases}$
$\delta(n - k)$	$z^{-k}$	
$u(n)$	$\frac{1}{1-z^{-1}}$	$u(n) = \begin{cases} 1 & , n \geq 0 \\ 0 & , n < 0 \end{cases}$
$a^n u(n)$	$\frac{1}{1-az^{-1}}$	
$x(n - k)u(n - k)$	$z^{-k} X(z)$	

## Diskret fouriertransform (DFT)

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jnk2\pi/N}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{jnk2\pi/N}$$

## Partialbråksuppdelning

1. Faktorisera nämnaren genom att hitta de komplexa nollställerna, t.ex. genom kvadratkompletering.
2. Skapa ett bråk för varje gradtal i täljaren och anta en av följande ansatser för var och en av bråken:

$$\frac{z}{(z-p_1)(z-p_2)} = \frac{A}{z-p_1} + \frac{B}{z-p_2}$$

$$\frac{1}{(z-p_1)(z-p_2)} = \frac{A}{z-p_1} + \frac{B}{z-p_2}$$

$$\frac{z}{(z-p)^2} = \frac{A}{(z-p)^2} + \frac{B}{z-p}$$

Observera att varje ansats kan multipliceras med  $z$  för att höja gradtalet, vilket kan vara önskvärt om man har högre grad i täljaren eller om man vill finna transformparen  $a^n \leftrightarrow \frac{z}{z-a}$ .

3. Skriv på samma bråkstreck och formulera ett ekvationssystem med en ekvation per gradtal hos  $z$ .

Exempel: Anta första ansatsen och höj gradtalet genom att multiplicera med  $z$  på bägge sidor. Ekvationssystemet blir:

$$z^2 = z^2(A+B) - z(Ap_2 + Bp_1)$$

$$z^2 : 1 = A + B$$

$$z^1 : 0 = Ap_2 + Bp_1$$

$$A = \frac{p_1}{p_1 - p_2} \quad B = \frac{p_2}{p_2 - p_1}$$

Bilaga E

Extentor med lösningar



KTH Datavetenskap  
och kommunikation

## 2F1120 Spektrala transformeringer för Media Tentamen 15 december 2005

Tentamen består av fem uppgifter där varje uppgift maximalt ger 4 p. Normalt gäller följande betygsgränser: **3:10 p, 4: 14 p, 5: 18 p**

Tillåtna hjälpmedel: räknare, formelblad

*Lycka till!*

### 1

En analog fyrkantvåg med frekvensen  $f_0$  samplas vid  $f_s = 8000$  Hz, efter att ha filtrerats med ett anti-vikningsfilter vid  $f_s/2$ . Den samplade signalen placeras i vektorn  $x(n)$ .

- a Skissa spektrum upp till halva samplingsfrekvensen av  $x(n)$  då  $f_0 = 800$  respektive 1600 Hz. Frekvensaxeln ska vara graderad från 0 till  $\pi$ , och inbördes relationen mellan deltonernas amplituder ska framgå. (2p)
- b En ny signal  $y(n)$  skapas genom att ta vartannat sampel av  $x(n)$ , dvs  $y(n) = x(2n)$ . Skissa spektrum för  $y(n)$  då  $f_0 = 800$  Hz. (2p)

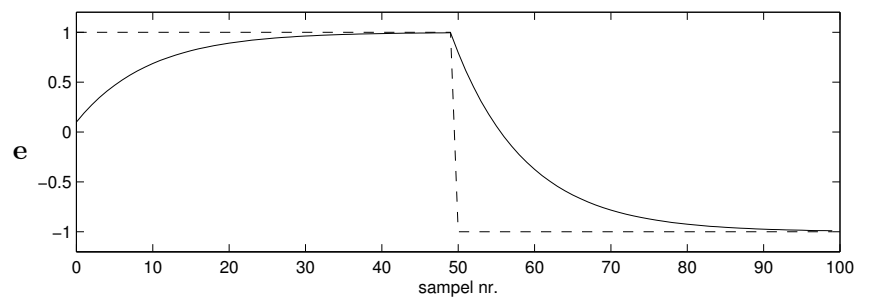
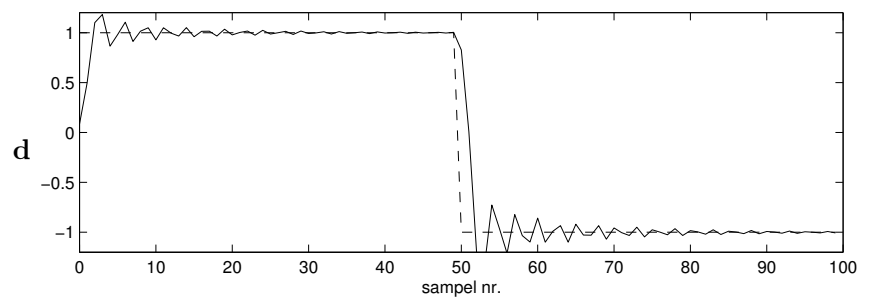
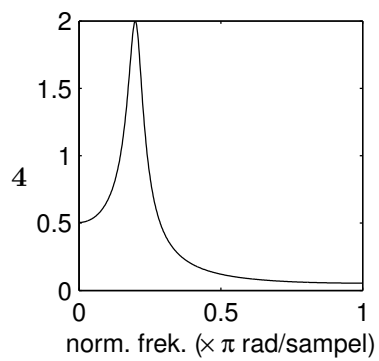
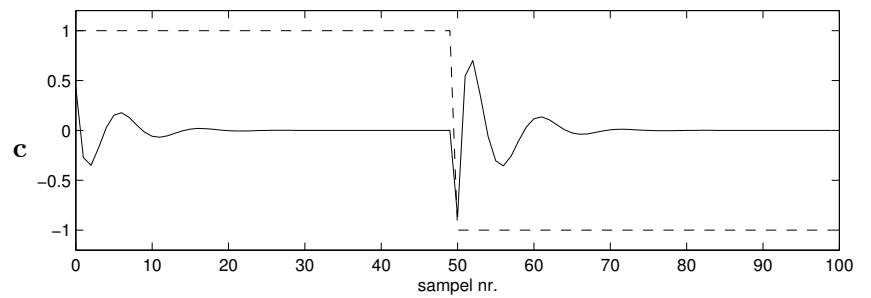
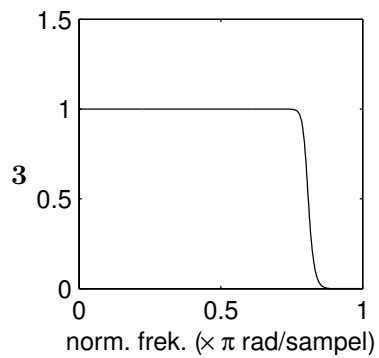
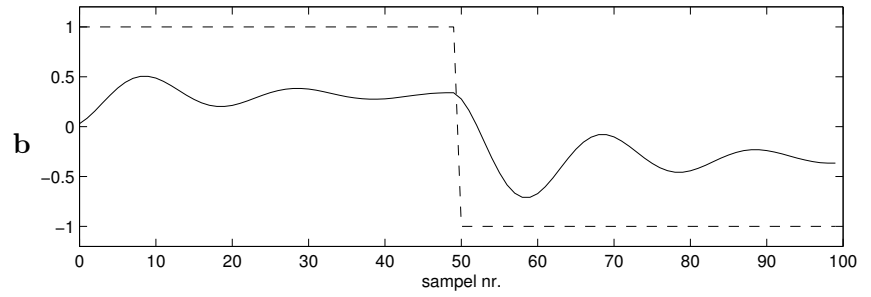
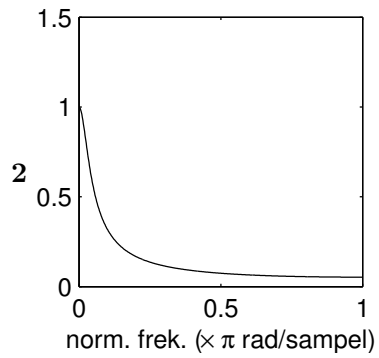
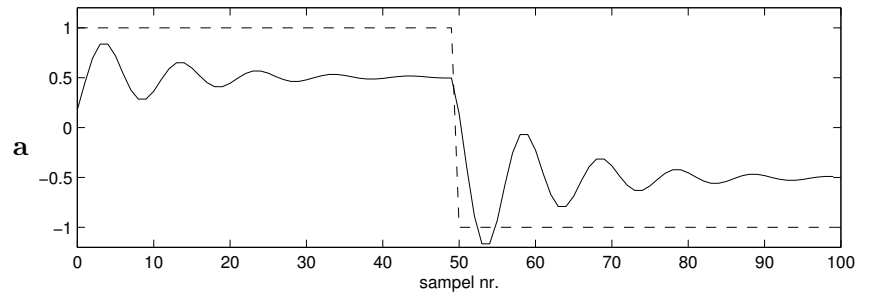
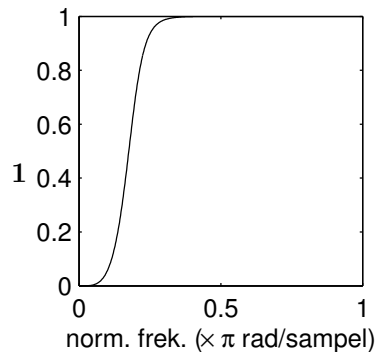
### 2

I figur 1 ser du frekvenssvar (till vänster) och insignal/utsignal-plottar (till höger) för ett antal linjära tidsinvarianta system. I insignal/utsignal-plottarna har en fyrkantvåg (1,-1) utgjort insignalen (streckad linje), och utsignalen är heldragen. Para ihop rätt frekvenssvar med rätt in/utsignalplot! Observera att det blir ett in/utsignalplot över. Motivera kort varje par. (1p/korrekt par)

### 3

Teknologen Kalle har fått ett extraknäck som assistent på ett laboratorium. Hans första uppgift blir att försöka "snygga upp" ett antal brusiga mätdatafiler. Kalles första idé är att filtrera alla filer med ett rullande-medelvärdesfilter av längden 5.

- a Hjälpt kalle att skriva *filterekvationen* ( $y(n) = \dots$ ) och *överföringsfunktionen*  $H(z)$  för filtret. (1p)
- b Kalle tycker att filtret tar bort lite för mycket högfrekvent information och vill prova något annat, men upptäcker till sin fasa att han skrivit över originalfilerna! Efter en sekunds panik, så drar han sig till minnes att det ska gå att konstruera ett filter som gör motsatsen, och får tillbaka originalsekvensen, ur den filtrerade. Tyvärr minns han inte hur...  
Hjälpt Kalle att få behålla jobbet, och skriv *överföringsfunktionen*  $G(z)$  och *filterekvationen* för ett filter som är motsatsen (inversen) till filtret i uppgift a (2p)



Figur 1.

- c Visa att inversfiltret  $G$  fungerar genom att filtrera impulsvaret från filter  $H$  – utsignalen ska bli en impuls ( $1p$ ).

#### 4

Låt  $h$  vara en Gauss-kärna av storleken  $N \times N$  som beskrivs av funktionen  $h(x, y) = \alpha e^{-(x^2+y^2)/\tau^2}$ , där  $\alpha$  är en konstant satt så att summan av kärnans element ett:  $\sum_{x=-N/2}^{N/2} \sum_{y=-N/2}^{N/2} h(x, y) = 1$ . Låt vidare  $a$  och  $b$  vara två bilder, där  $b = a * h$  (dvs  $b$  fås genom att filtrera  $a$  med kärnan  $h$ ).

- a Beskriv hur  $b$  förhåller sig till  $a$  med avseende på frekvensinnehåll, dvs vilken typ av filter är  $h$ ? ( $1p$ )
- b En ny bild  $c$  skapas som differensen  $c = a - b$ . Denna operation är ekvivalent med att filtrera  $a$  med en annan filterkärna  $g$ , dvs  $c = a * g$ . Bestäm  $g(x, y)$ . ( $1p$ )
- c Vad är  $g$  för typ av filter? Förklara genom ett frekvensdomänsresonemang med utgångspunkt från  $h$ . ( $2p$ )

#### 5

När frekvensinnehållet i en signal ska analyseras över tid (t.ex. som ett spektrogram), är ett vanligt förfarande att göra FFT på  $N$  sampel åt gången, och hela tiden flytta startpunkten ett bestämt steg om  $M$  sampel, där typiskt  $M < N$ . Om steget  $M$  är väldigt litet, t.ex. bara ett sampel, så blir denna metod onödigt kostsam – det visar sig att det finns ett bättre sätt:

Låt  $X_q(k)$  beteckna en  $N$ -punkters DFT för sekvensen  $x(q) \dots x(q + N - 1)$ . Då kan  $X_{q+1}(k)$  (DFT för sekvensen  $x(q + 1) \dots x(q + N)$ ) räknas ut ur det föregående värdet med följande rekursiva uttryck:

$$X_{q+1}(k) = e^{j2\pi k/N} (X_q(k) - x(q) + x(N + q))$$

Denna rekursiva metod kallas ibland för "Sliding DFT" (SDFT), eftersom man låter DFT'ns inputfönster "glida" över signalen, ett sampel i taget.

- a Gör en uppskattning för vilka värden på  $M$  som det är effektivare (räknat i antal multiplikationer) att använda SDFT än FFT. ( $2p$ )
- b Visa, med utgångspunkt i formeln för DFT'n, att ovanstående rekursiva uttryck stämmer! ( $2p$ )



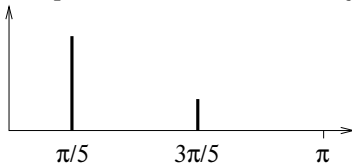
## Lösningar

1

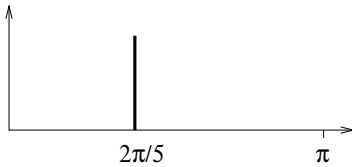
a

Fourierseriekoefficienterna för en fyrkantvåg är  $c_k = \frac{-2j}{k\pi}$  för udda  $k$ , och 0 för jämna  $k$ , vi får deltoner vid  $f_0, 3f_0, 5f_0$  osv.

$f_0 = 800$  Hz ( $\omega = 2\pi f_0/f_s = \pi/5$ ) ger deltoner vid 800 Hz, 2400 Hz (4000 Hz och uppåt tas bort av antivikningsfiltret). Spektrum kommer att ha två komponenter, vid  $\pi/5$  och  $3\pi/5$ , där höjden på den andra är en tredjedel av den första.



$f_0 = 1600$  Hz ( $\omega = 2\pi/5$ ) ger deltoner vid 1600 Hz, 4800 Hz... endast 1600 Hz passerar antivikningsfiltret. Spektrum kommer att ha en komponent, vid  $2\pi/5$ .

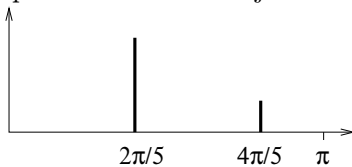


b

Att ta vartannat sampel innebär att sampla om signalen till halva samplingsfrekvensen. Detta innebär att frekvenskomponenternas normerade vinkelfrekvens ( $\omega$  uttryckt i radianer per sampel) kommer att dubblas. Det innebär också att alla komponenter som hamnar över  $\pi$  vid denna frekvensdubbling kommer att vikas ned.

Första komponenten blir  $2\pi/5$ , och klarar sig undan vikning. Andra komponenten blir  $6\pi/5$ , vilket är större än  $\pi$  och kommer därför vikas ned till  $2\pi i - 6\pi/5 = 4\pi/5$ .

Spektrums två komponenter kommer alltså att ligga vid  $\frac{2\pi}{5}$  och  $\frac{4\pi}{5}$  där höjden på den andra komponenten är en tredjedel av den första.



## 2

- 1 - c Frekvenssvaret är högpas, ingen DC slipper igenom, utsignalen ska svänga in mot noll, och det gör den bara i  $c$ .
- 2 - e Frekvenssvaret är ett lågpasfilter med låg brytfrekvens, i princip bara DC slipper igenom  $\rightarrow$  långsamma rörelser i utsignalen.
- 3 - d Brant lågpas-filter, med hög brytfrekvens "ringningar" uppåt nykvistfrekvensen. Förstärkning=1 vid  $\omega = 0$  innebär att utsignalen ska svänga in mot insignalen.
- 4 - a Frekvenstopp vid ca  $0.2\pi$  innebär en svängningsperiod på 10 sampel, vilket stämmer med  $a$ , men inte  $b$ . DC-förstärkning = 0.5, utsignalen ska svänga in mot  $0.5 \times$  utsignalen, tyder också på  $a$ .

## 3

### a

Moving-average-filter av längden 5:

$$y(n) = \frac{x(n) + x(n-1) + x(n-2) + x(n-3) + x(n-4)}{5}$$

ger

$$H(z) = \frac{1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}}{5}$$

### b

Vi söker ett filter  $G(z)$  som uppfyller  $H(z)G(z) = 1$ .

$$G(z) = \frac{1}{H(z)} = \frac{5}{1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}}$$

vilket ger filterekvationen

$$y(n) = 5x(n) - y(n-1) - y(n-2) - y(n-3) - y(n-4)$$

### c

Sätt  $x(n)$  till impulssvaret  $h(n)$ :

$n$	$h(n)$
0	.2
1	.2
2	.2
3	.2
4	.2
5	0
$\vdots$	$\vdots$

Sätt upp alla termer i filterekvationen i en tabell och räknar ut  $y(n)$  sampel för sampel.

$n$	$5x(n)$	$-y(n-1)$	$-y(n-2)$	$-y(n-3)$	$-y(n-4)$	$= y(n)$
0	1	0	0	0	0	1
1	1	-1	0	0	0	0
2	1	0	-1	0	0	0
3	1	0	0	-1	0	0
4	1	0	0	0	-1	0
5	0	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$y(n)$  blir en impuls.

4

a

Gauss-kärnan är ett lågpasfilter, högfrekvent innehåll har tagits bort ur bilden, den har blivit "suddig".

b

$$c = a - a * h$$

Men  $a = a * \delta$  där  $\delta$  är en impuls:

$$\delta(x, y) = \begin{cases} 1 & \text{om } x = 0 \text{ och } y = 0 \\ 0 & \text{annars} \end{cases}$$

(Om man filtrerar en signal med en impuls får man tillbaka signalen själv.)

Då kan  $c$  skrivas som

$$c = a * \delta - a * h = a * (\delta - h)$$

vilket ger att  $g = \delta - h$ :

$$g(x, y) = \delta(x, y) - e^{-(x^2+y^2)/r^2}$$

c

Eftersom  $h$  är ett lågpasfilter, och  $\delta$  lämnar bilden oförändrad, (släpper igenom alla frekvenser lika mycket), så måste  $\delta - h$  innebära "hela bilden minus de låga frekvenserna", och alltså släppa igenom mera av de höga frekvenserna.  $g$  är alltså ett *högpasfilter*.

5

a

En  $N$ -punkters FFT'n kräver ca  $N \log N$  multiplikationer. En  $N$ -punkters SDF'T kräver en multiplikation per frekvensvärde ( $k = 0 \dots N-1$ ), dvs  $N$  multiplikationer för varje sampel som fönstret ska flyttas. Detta innebär att om fönstret flyttas  $\log N$  steg så kommer de två metoderna vara ungefär lika snabba.

Alltså: SDF'T är snabbare när  $M < \log N$ , annars är FFT att föredra.

6 (7)

**b**

DFT-ekvationen ger att

$$X_q(k) = \sum_{n=0}^{N-1} x(n+q)e^{-j2\pi nk/N}$$

och

$$X_{q+1}(k) = \sum_{n=0}^{N-1} x(n+q+1)e^{-j2\pi nk/N}$$

Om vi sedan sätter  $m = n + 1$  kan vi skriva

$$X_{q+1}(k) = \sum_{m=1}^N x(m+q)e^{-j2\pi(m-1)k/N}$$

och bryta ut exponentialtermen  $e^{j2\pi k/N}$  vilket ger

$$X_{q+1}(k) = e^{j2\pi k/N} \sum_{m=1}^N x(m+q)e^{-j2\pi mk/N}$$

Nu kan vi skriva om summorna genom att separera ut första resp. sista termen:

$$X_q(k) = x(q) + \sum_{n=1}^{N-1} x(n+q)e^{-j2\pi nk/N}$$

respektive

$$X_{q+1}(k) = e^{j2\pi k/N} \left( \sum_{m=1}^{N-1} x(m+q)e^{-j2\pi mk/N} + x(N+q) \right)$$

Men nu är det uppenbart att summorna i de två uttrycken är identiska, det är bara olika summationsvariabler. Vi skriver summan som  $X_q(k) - x(q)$  och stoppar in i uttrycket för  $X_{q+1}(k)$ , vilket ger

$$X_{q+1}(k) = e^{j2\pi k/N} (X_q(k) - x(q) + x(N+q))$$

Q.E.D.



KTH Datavetenskap  
och kommunikation

## 2F1120 Spektrala transformeringer för Media Tentamen 061218

Tentamen består av fem uppgifter där varje uppgift maximalt ger 4 p. Normalt gäller följande betygsgränser: **3:9 p, 4: 13 p, 5: 17 p**

Tillåtna hjälpmedel: räknare, formelblad

### 1

Den analoga signalen i figur 1 lågpasfilteras vid 8000 Hz och samplas sedan med samplingsfrekvensen  $f_s$ .

- Skissa spektrum upp till nykvistfrekvensen av den samplade signalen då  $f_s = 16000$  Hz. Frekvensaxeln ska vara graderad, och deltonernas inbördes relation ska vara korrekt. (2p)
- Som ovan, men  $f_s = 8000$  Hz. (2p)

### 2

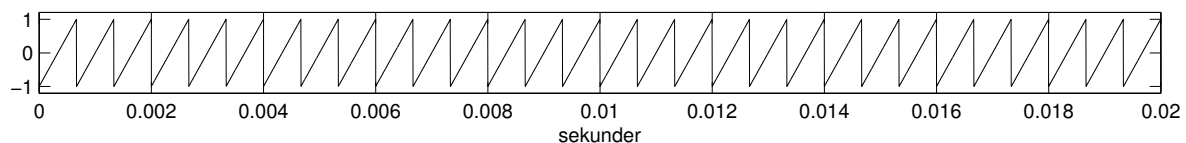
I figur 2 ser du impulssvaren från ett antal filter. Impulssvaren till vänster kommer alla från icke-återkopplade filter, sk. FIR-filter. Impulssvaren till höger kommer alla från återkopplade filter (eller IIR-filter), och fyra av dessa utgör *inversen* till något av FIR-filtren till vänster (dvs sätter man dem i serie blir sammanlagda impulssvaret en impuls). Para ihop varje FIR-impulssvar med motsvarande IIR-impulssvar! Observera att det blir ett IIR-impulssvar över. Motivera kort varje par. (1p/ korrekt par)

Kom ihåg: rötterna till andragradsekvationen

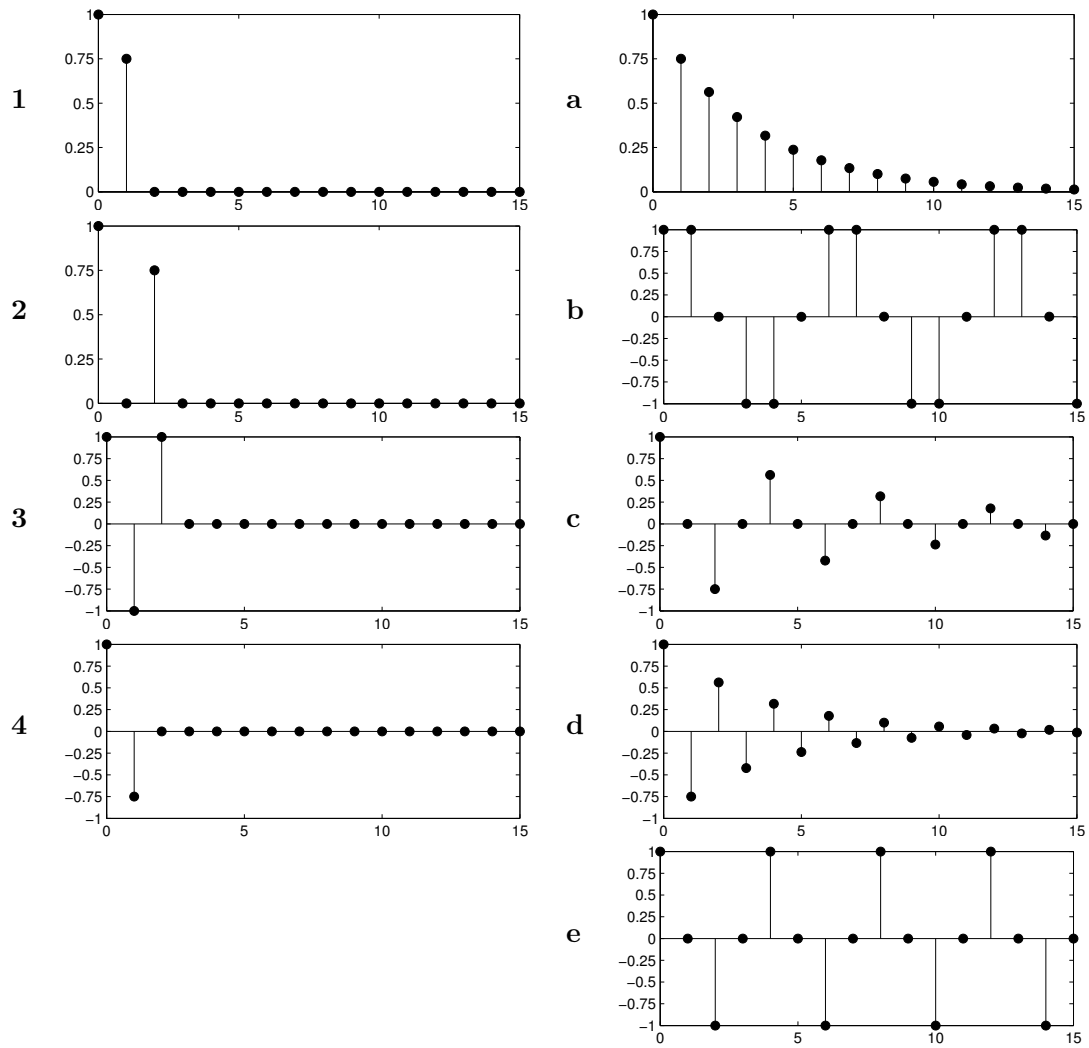
$$z^2 + pz + q = 0$$

ges av

$$z = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$$



Figur 1.



Figur 2.

### 3

Gunilla extraknacker som frilansande journalist. Arbetet, som hon sköter mestadels hemifrån, är till stor del baserat på telefonintervjuer. Av en bekant får hon låna en liten "låda" för att tappa av telefonljudet in i datorn för att underlätta transkriptionen. När Gunilla har gjort tre intervjuer lyssnar hon på sina filer - och får en smärre chock. Det låter hemskt, det går med nöd och näppe att höra vad som sägs... Gunilla, som har läst spektrala transformeringar, studerar spektrogrammet av en inspelning och konstaterar en kraftig svart linje som löper genom hela signalen vid frekvensen 50 Hz - gissningsvis överlagrat nätbrum härstammande från en dåligt isolerad nätadapter. Gunilla noterar vidare att telefonsignalens energiinnehåll är koncentrerat till området 400-3400 Hz, och hon inser att det borde gå ganska bra att filtrera bort brummet med ett enkelt filter.

Hjälp Gunilla att skapa enklaste tänkbara filter som eliminerar 50 Hz-störningen, men släpper igenom talsignalen! (Det ska kunna åstadkommas med ett icke återkopplat filter av längden 3, med ett strategiskt placerat nollställespar.) Inspelningarna är gjorda med samplingsfrekvensen 8000 Hz.

- a Ange filterekvationen och överföringsfunktionen för filtret. (2p)
- b Skissa frekvensgången för filtret från 0 till 4000 Hz, med båda axlarna graderade. Skissa även, med graderad frekvensaxel, spektrum för insignal och utsignal, där den önskvärda delen av telefonsignalen (talet) kan antas ha sitt energiinnehåll jämnt fördelat i intervallet 400-3400 Hz. (2p)

### 4

- a Filtrering av bilder med stora filterkärnor kan vara en kostsam operation, om man direkt applicerar formeln för faltning i 2D. Som tur är finns det några tricks att tillgå... Beskriv två tekniker för att snabba upp filtrering av bilder, och ange för varje teknik
  - hur den fungerar, och varför det går snabbare
  - eventuella begränsningar eller restriktioner på filterkärnan

(2p)

- b Varför är det önskvärt att en filterkärna är cirkulärt symmetrisk? Ge två exempel på sådana filterkärnor! (2p)

### 5

*Subtraktiv syntes* kallas en populär ljudsyntesmetod som åt minstone fram till digitalteknikens intåg var den helt dominerande i musiksynthesizers. Huvudidén är att en enkel källvågform behandlas av olika filter och förstärkare (som kan göras tidsvariabla), för att skapa ett ljud som är mer intressant och dynamiskt än ursprungsvågformen. Ordet subtraktiv syftar alltså på att man formar sitt önskade ljud genom att skära (filtrera) bort delar av råmaterialet (källvågformen)<sup>1</sup>.

- a På de flesta subtraktiva synthar finns flera källvågformer att välja mellan, såsom såg-tandsvåg, fyrkantvåg och brus. Däremot hittar man mindre ofta sinusvåg. Vad kan vara anledningen till det? (2p)

---

<sup>1</sup>Motsatsen heter additiv syntes, där ljudet byggs upp genom att kombinera ihop ett stort antal sinustoner.

**b** För att göra källvågformen mer komplex, kan olika tekniker användas. *Ringmodulation* är en sådan teknik som man hittar ibland på synthar, och det är bara ett finare ord för multiplikation. Man multiplicerar helt enkelt en vågform med en annan. Beskriv (med ord) hur spektrum av en fyrkantvåg med  $f_0 = 500$  Hz påverkas när den “ringmoduleras” med en sinusvåg på 100 Hz. *ledning: använd faltningsteoremet (2p)*

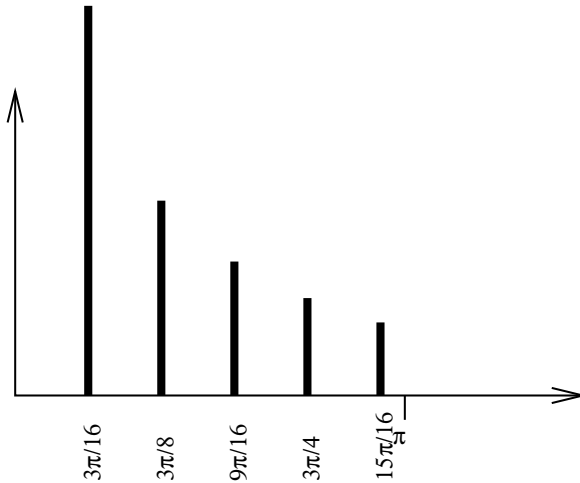


## Lösningar

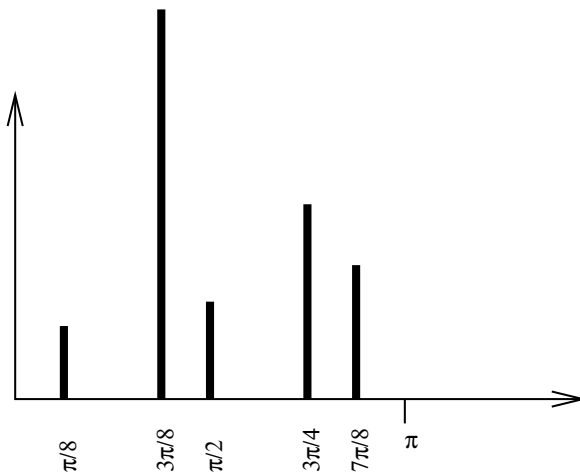
### 1

Sågtandsvågen i figur 1 gör tre perioder på 0.002 s, vilket ger  $f_0 = 3/0.002 = 1500$  Hz. spektrum av signalen ges av fourierserien (se formelblad) där beloppet av deltonernas amplitud avtar som  $1/k$ .

- a Lågpassfiltrering vid 8 kHz och sampling vid 16000 Hz resulterar inte i deltoner vid 1500, 3000, 4500, 6000 och 7500 Hz eller  $\frac{3\pi}{16}$ ,  $\frac{3\pi}{8}$ ,  $\frac{9\pi}{16}$ ,  $\frac{3\pi}{4}$  samt  $\frac{15\pi}{16}$  rad/sampel.



- b Lågpassfiltrering vid 8 kHz och sampling vid 8000 Hz ger vikning av deltonerna mellan 4000 och 8000 Hz, vilket ger deltonsfrekvenserna ger 1500, 3000, 3500, 2000 och 500 Hz eller  $\frac{3\pi}{8}$ ,  $\frac{3\pi}{4}$ ,  $\frac{7\pi}{8}$ ,  $\frac{\pi}{2}$  samt  $\frac{\pi}{8}$  . . .



### 2

Om vi betecknar FIR-filtrets överföringsfunktion med  $H(z)$  och IIR-filtret med  $G(z)$  så gäller ju att  $G(z) = \frac{1}{H(z)}$ . Impulssvaret för FIR-filtret ger direkt  $H(z)$  eftersom filterkoefficienterna för ett icke återkopplat filter är lika med impulssvaret, dvs  $b_n = h(n)$ . Sedan kan man beräkna nollställena  $H(z) = 0$ , vilka alltså blir till poler i  $G(z)$ .

*Alternativ lösning:* faltningen mellan mellan impulsvaret till höger och det till vänster ska bli en impuls om de "passar ihop". Bara att räkna på...

- 1 - d impulsvaret  $\{1, 0.75, 0, 0 \dots\}$  ger  $H(z) = 1 + 0.75z^{-1} = \frac{z+0.75}{z}$ , vilket ger nollstället  $z = -0.75$ . IIR-filtret får en pol vid  $-0.75$  vilket innebär en svängning med frekvensen  $\pi$  rad/sampel, dvs periodtiden 2 sampel, alltså *d*.
- 2 - c  $H(z) = 1 + 0.75z^{-2}$  ger nollställena  $z = \pm j\sqrt{\frac{3}{2}}$ . IIR-filtret får polvinkeln  $\pi/2$ , vilket innebär en period på 4 sampel. Detta stämmer för *c* och *e*. Polradien 0.75 innebär dock en dämpad svängning, alltså *c*.
- 3 - b  $H(z) = 1 - z^{-1} + z^{-2}$  ger nollställen i  $z = \frac{1}{2} \pm j\sqrt{\frac{1}{4} - 1} = \frac{1}{2} \pm j\frac{\sqrt{3}}{2}$ . För IIR-filtret innebär detta polradie 1 och polvinkel  $\arctan \sqrt{3} = \pi/3$  dvs en periodtid på 6 sampel, vilket bara kan vara *b*.
- 4 - a  $H(z) = 1 - 0.75z^{-1}$  ger nollstället  $z = 0.75$ . IIR-filtret får en pol på positiva reella axeln  $< 1$ . Det ger ingen svängning, endast avklingning, dvs *a*.

### 3

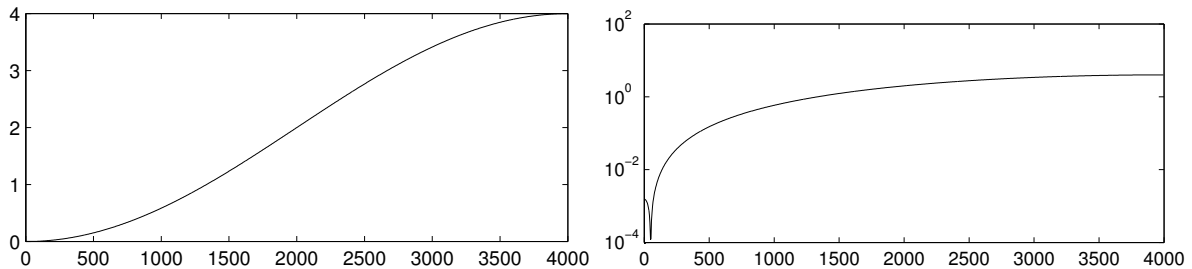
Störningen kan filtreras bort med ett FIR-filter med ett nollställespar i  $e^{\pm j\omega_0}$ .  $f = 50$  Hz vid  $f_s = 8000$  Hz ger  $\omega_0 = 2\pi\frac{50}{8000} = \frac{\pi}{80}$  rad/sampel.

$$H(z) = \frac{(z - e^{j\frac{\pi}{80}})(z - e^{-j\frac{\pi}{80}})}{z^2} = \frac{z^2 - z(e^{j\frac{\pi}{80}} + e^{-j\frac{\pi}{80}}) + 1}{z^2} = 1 - 2\cos\frac{\pi}{80}z^{-1} + z^{-2}$$

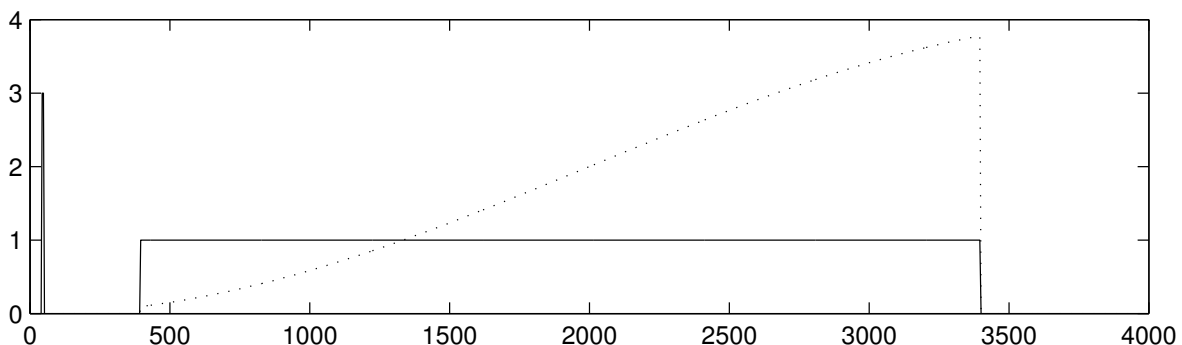
och

$$y(n) = x(n) - 2\cos\frac{\pi}{80}x(n-1) + x(n-2)$$

Frekvensgången fås genom att beräkna  $|H(e^{j\omega})|$  för några olika värden på  $\omega$ , t.ex.  $\omega = 0, \pi/80, \pi/4, \pi/2, 3\pi/4$  samt  $\pi$ .



Absolutbeloppet av  $H(\omega)$  plottat med y-axel-skalan linjär (till vänster) respektive logaritmisk (till höger).



Spektrum av signalen (heldaragen) samt och utsignal(streckad).

## 4

- a** *Metod 1: separerbar filterkärna.* Om filterkärnan är beskaffad så att den går att dela upp i två oberoende funktioner av  $x$  respektive  $y$  så kan summationen i  $x$ - och  $y$ -led göras separat, vilket ger en besparing på en faktor  $N$  för en  $N \times N$  stor filterkärna.

*Metod 2: faltning som multiplikation i frekvensdomänen.* Genom att beräkna FFT'n i två dimensioner av både bilden och filterkärnan, kan faltningen göras som en multiplikation av de trasformerade bilderna. Resultatet transformeras sedan tillbaka till en bild. Metoden är snabbare än direkt faltning tack vare FFT'n.

- b** En cirkulär filterkärna påverkar bilden på samma sätt i alla riktningar. Två exempel är Gausskärna och s.k. "pillerburk", som är konstant innanför en cirkel, och noll utanför.

## 5

- a** Metoden bygger på att det är en övertonrik källvågform, så det blir något att filtrera, och en sinusvåg har inga övertoner.

- b** Ringmodulation, eller amplitudmodulering, innebär en faltning av signalernas spektra. Om spektrum från en sågtandsvåg faltas med spektrum från en sinuston på 100 Hz, kommer sågtandsvägens deltoner att förskjutas uppåt i frekvensled med 100 Hz. Men vid denna faltning behöver man även ta hänsyn till negativa frekvenskomponenter. Sinustonen har en sådan spegelbild i -100 Hz, därför kommer även en kopia av spektrum att förskjutas nedåt med 100 Hz. Resultatet blir alltså två 1:a deltoner på 400 Hz och 600 Hz, 2:a deltoner på 900 och 1100 Hz, 3:e på 1400 och 1600 Hz, osv. Ljudet blir metallsikt, eftersom deltonerna inte längre ligger på jämna multiplar av grundtonen, effekten används ofta för att simulera klockljud mm, därav namnet ringmodulation.



KTH Datavetenskap  
och kommunikation

## 2F1120 Spektrala transformeringer för Media Tentamen 070604

Tentamen består av fem uppgifter där varje uppgift maximalt ger 4 p. Normalt gäller följande betygsgränser: **3:9 p, 4: 13 p, 5: 17 p**  
Tillåtna hjälpmedel: räknare, formelblad

*Lycka till!*

### 1

En ren sinuston med frekvensen 1500 Hz samplas med  $f_s = 8000$  Hz. Samplerna placeras i vektorn  $x(n)$ . Man skapar sedan en ny signal  $y(n) = x(n)^2$ .

- a Skissa spektrum upp till halva samplingsfrekvensen för  $y(n)$ . Motivera din skiss med ett tydligt resonemang. Frekvensaxeln ska vara graderad. (2p)
- b Som ovan, men  $y(n) = x(n)w(n)$ , där  $w(n)$  samplats på samma sätt som  $x(n)$  från en sinuston med samma amplitud som den första men med frekvensen 2000 Hz. (2p)

### 2

I figur 1 ser du ett antal pol/nollställesplottar och ett antal impulssvar. Para ihop rätt polplot med rätt impulssvar! Observera att det blir ett impulssvar över. Motivera kort varje par. (1p/korrekt par)

### 3

Betrakta filtret

$$H(z) = \frac{z^2 - z + 1}{z^2}$$

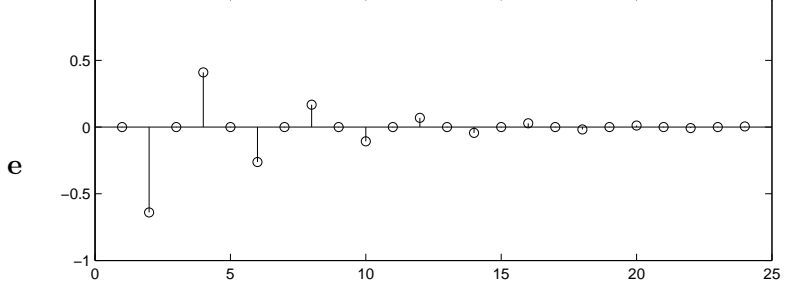
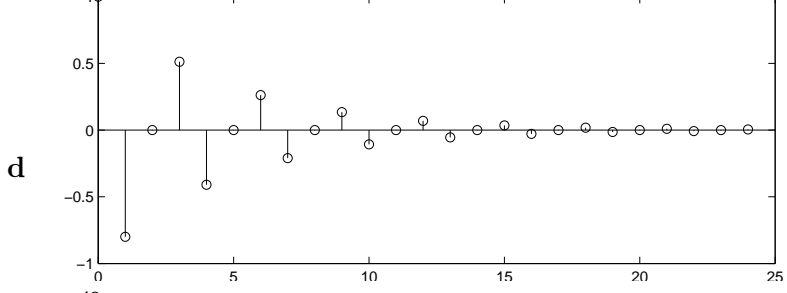
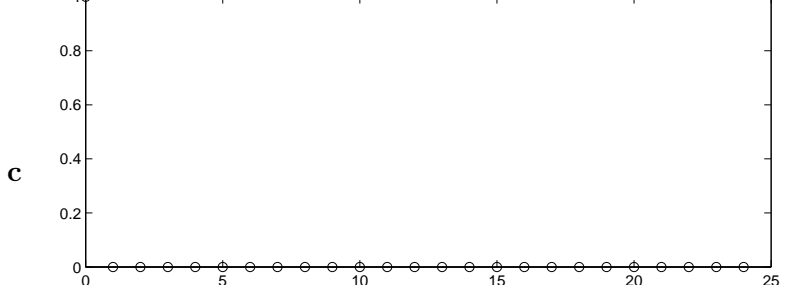
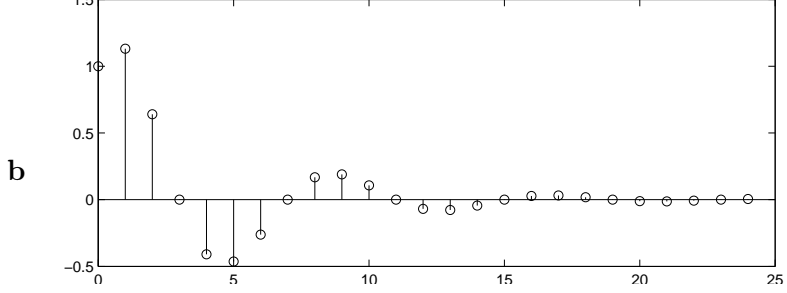
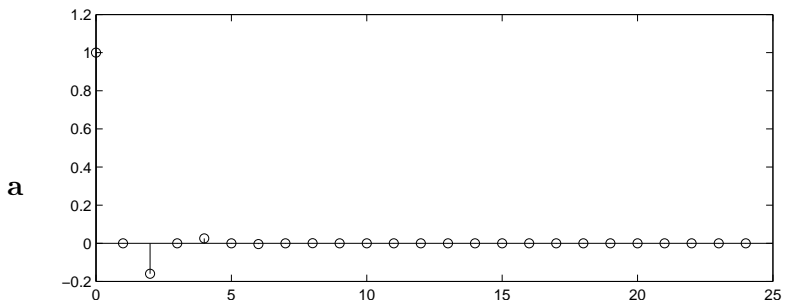
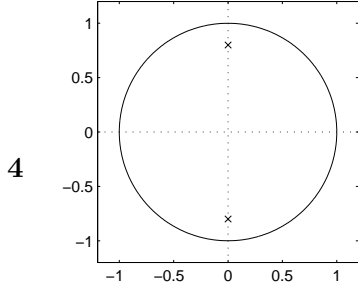
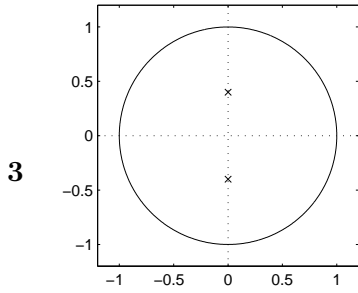
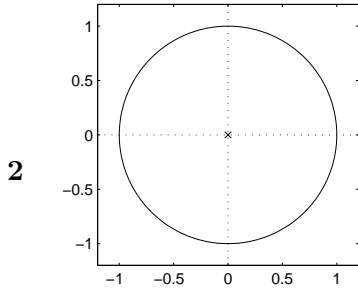
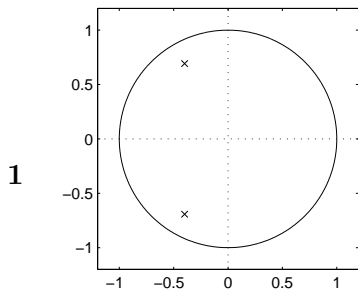
- a Skissa överföringsfunktionens belopp  $|H(\omega)|$ . Vid vilken frekvens infaller dess minimum? (2p)
- b Skriv filterekvationen  $y(n) = \dots$  för *inversen* till ovanstående filter (2p)

*Kom ihåg:* rötterna till andragradsekvationen

$$z^2 + pz + q = 0$$

ges av

$$z = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$$



**Figur 1.**

## 4

JPEG-kompression bygger på att bilden delas in i block, där varje block transformeras med DCT, vilket resulterar i en koefficientmatris.

- a Väldigt ofta är informationen koncentrerad till övre vänstra hörnet av koefficientmatrisen. Vad är tolkningen av detta, och hur utnyttjas det i JPEG-algoritmen? (2p)
- b Hur ska en bild vara beskaffad om informationen istället ska koncentreras till nedre högra hörnet av koefficientmatrisen? (2p)

## 5

Betrakta filtret  $y(n) = x(n) + x(n - 1) + x(n - 2) + x(n - 3) + x(n - 4)$

- a  $y(n)$  kan, om man tar hjälp av återkoppling, skrivas med endast tre termer. Hur? (2p)
- b På ett liknande sätt kan filtret

$$y(n) = x(n) + 2x(n - 1) + 3x(n - 2) + 4x(n - 3) + 5x(n - 4) + 4x(n - 5) + 3x(n - 6) + 2x(n - 7) + x(n - 8)$$

skrivs med endast 5 termer. Gör det!

*ledning: studera impulssvaret från första filtret, faltat med sig själv (2p)*

## Lösningar

1

$x(t) = \sin 2\pi f_0 t$  där  $f_0 = 1500$  Hz

Sampling med  $f_s = 8000$  Hz ger

$$x(n) = \sin 2\pi f_0 n / f_s = \sin \omega_0 n \text{ där } \omega_0 = 2\pi f_0 / f_s = \frac{3\pi}{8}$$

a

$$y(n) = x^2(n) = \sin^2 \omega_0 n = \frac{1 - \cos 2\omega_0 n}{2}$$

se formelsamling för  $\sin^2 \alpha$

Spektrum innehåller alltså frekvenserna 0 och  $2\omega_0$  i lika proportion, dvs två lika höga "spikar", vid 0 och  $\frac{3\pi}{4}$ .

b

$$y(n) = x(n)w(n) = \sin \omega_0 n \sin \omega_1 n$$

där  $\omega_1 = 2\pi f_1 / f_s = \frac{\pi}{2}$  Med hjälp av Eulers formler kan detta göras om till

$$y(n) = \frac{\cos(\omega_0 - \omega_1)n - \cos(\omega_0 + \omega_1)n}{2}$$

men  $\cos(x) = \cos(-x)$  vilket innebär att vi lika gärna kan skriva

$$y(n) = \frac{\cos(\omega_1 - \omega_0)n - \cos(\omega_0 + \omega_1)n}{2}$$

Signalen innehåller alltså skillanden och summan av de två frekvenserna,  $\omega_1 - \omega_0 = \frac{\pi}{8}$  och  $\omega_1 + \omega_0 = \frac{7\pi}{8}$  i lika stora proportioner.

Alla frekvenser ligger under  $\pi$ , så ingen vikning förekommer.

2

- 1 - d Polradien strax under 1 vilket ger svagt dämpad svängning. Polvinkeln är  $2\pi/3$  vilket ger svängningsperiod på 3 sampel
- 2 - c Endast en pol i origo = rak frekvensgång (påverkar alla frekvenser lika), impulssvaret blir en impuls.
- 3 - a Polvinkeln är  $\pi/2$  vilket ger svängningsperioden 4 sampel. Det måste vara a eller e. Polen ligger dock ganska nära origo vilket ger stor dämpning, alltså a.
- 4 - e Polvinkeln är  $\pi/2$  vilket ger svängningsperioden 4 sampel. Det måste vara a eller e. Polradien är dock nära 1 vilket ger liten dämpning, alltså e.

3

$$|H(\omega)| = |e^{j2\omega} - e^{j\omega} + 1|$$

4 (??)

**a**

Täljarpolynommet  $z^2 - z + 1$  har rötterna  $z = \frac{1 \pm j\sqrt{3}}{2} = e^{\pm j\frac{\pi}{3}}$ . Filtret har alltså nollställen på enhetscirkeln, och  $|H(\omega)|$  kommer att ha nollställen vid  $\omega = \pm\pi/3$ . Några punkter på kurvan blir:

- $\omega = 0$  ger  $|H(\omega)| = 1$
- $\omega = \pi/3$  ger  $|H(\omega)| = 0$
- $\omega = \pi/2$  ger  $|H(\omega)| = 1$
- $\omega = 2\pi/3$  ger  $|H(\omega)| = 2$
- $\omega = \pi$  ger  $|H(\omega)| = 3$

**b**

Det inversa filtret är  $G(z) = \frac{z^2}{z^2 - z + 1} = \frac{1}{1 - z^{-1} + z^{-2}}$  vilket ger  $Y(z)(1 - z^{-1} + z^{-2}) = X(z)$ . Detta motsvarar filterekvationen

$$y(n) = y(n - 1) - y(n - 2) + x(n)$$

**4**

**a**

Övre vänstra hörnet av koefficientmatrisen motsvarar låga spatiala frekvenser, dvs. information som ändrar sig lite från en pixel till nästa, t.ex. vid enfärgade eller långsamt varierande fält i bilden, vilket är vanligt förekommande. Stora delar av matrisen ofta består ofta av nollor, och kan därför komprimeras effektivt, vilket är en av grundidéerna bakom JPEG-algoritmen.

**b**

Om bilden ändras mycket från pixel till pixel så koncentreras informationen till nedre högra hörnet i koefficientmatrisen. Ett exempel kan vara en bild med varannan svart och varannan vit pixel, som ett schackbräde.

**5**

**a**

Om

$$y(n) = x(n) + x(n - 1) + x(n - 2) + x(n - 3) + x(n - 4)$$

så kan föregående värde  $y(n - 1)$  skrivas som

$$y(n - 1) = x(n - 1) + x(n - 2) + x(n - 3) + x(n - 4) + x(n - 5)$$

om man nu subtraherar andra uttrycket från det första får man

$$y(n) - y(n - 1) = x(n) - x(n - 5)$$

vilket ger

$$y(n) = y(n - 1) + x(n) - x(n - 5)$$



**b**

impulssvaret från filtret

$$y(n) = x(n) + 2x(n-1) + 3x(n-2) + 4x(n-3) + 5x(n-4) + 4x(n-5) + 3x(n-6) + 2x(n-7) + x(n-8)$$

är lika med impulssvaret från filtret

$$y(n) = x(n) + x(n-1) + x(n-2) + x(n-3) + x(n-4)$$

faltat med sig själv. Det innebär att vi kan erhålla det andra filtret genom att kaskadkoppla två av den första sorten. Filtret i uppgift a har i sin återkopplade form överföringsfunktionen

$$H_a(z) = \frac{1 - z^{-5}}{1 - z^{-1}}$$

Kaskadkoppling innebär multiplikation av överföringsfunktionerna, alltså blir vårt nya filter

$$H_b(z) = \left( \frac{1 - z^{-5}}{1 - z^{-1}} \right)^2 = \frac{1 - 2z^{-5} + z^{-10}}{1 - 2z^{-1} + z^{-2}}$$

vilket ger

$$Y(z)(1 - 2z^{-1} + z^{-2}) = X(z)(1 - 2z^{-5} + z^{-10})$$

vilket ger filterekvationen

$$y(n) = 2y(n-1) - y(n-2) + x(n) - 2x(n-5) + x(n-10)$$



KTH Datavetenskap  
och kommunikation

## DT1120 Spektrala transformeringer för Media Tentamen 071217

Tentamen består av fem uppgifter där varje uppgift maximalt ger 4 p. Normalt gäller följande betygsgränser: **3:9 p, 4: 13 p, 5: 17 p**  
Tillåtna hjälpmedel: räknare, formelblad

*Lycka till!*

### 1

Ett bilhjul med fem ekrar (se fig. 1) filmas med 25 bilder/sekund. På filmen kommer hjulet ibland att se ut som det roterar framlänges, ibland baklänges, och ibland som det står stilla, även när bilen åker framåt. Mellan två på varandra följande bilder kommer det att *se ut* som att hjulet roterat vinkeln  $\theta$ <sup>1</sup>. Ett positivt värde på  $\theta$  innebär att hjulet ser ut att rotera framlänges. Hjulets radie är 20 cm.

- a Vad blir  $\theta$  om bilen kör med hastigheten  $\pi$  m/s? (1p)
- b Vad blir  $\theta$  om bilen kör med hastigheten  $2\pi$  m/s? (1p)
- c Om man vill uppskatta bilens hastighet utifrån filmsekvensen, så inser man att det finns flera möjliga svar. Ange vad detta fenomen kallas (framförallt i ljud-domänen) samt ställ upp ett allmänt uttryck för bilens hastighet, givet rotationsvinkeln  $\theta$  *ledning: uttrycket*

---

<sup>1</sup>Man ska i detta antaga att hjulet är helt symmetriskt m.a.p. de fem ekrarna, och alltså bortse från ev. ventiler, logotyper, smuts och andra detaljer på hjulet som inte är i symmetri med ekerstrukturen.



Figur 1.

kommer innehålla en obestämd heltalsvariabel, t.ex.  $k$ , där  $k=0$  representerar en lösning,  $k=1$  en annan osv... (2p)

## 2

I figur 2 ser du ett antal filterekvationer och ett antal pol/nollställesplottar. Para ihop rätt filterekvation med rätt plot! Observera att det blir en plot över. Motivera kort varje par. (1p/korrekt par)

## 3

En konsult inom ljudesign har utvecklat en signal för användning i restaurangmiljö, som på tidskontinuerlig form kan skrivas

$$x(t) = \sin(2\pi 1500t) + 0,1\sin(2\pi 2000t)$$

När signalen spelas upp visar det sig att en störfrekvens på 1333 Hz smugit sig in och förstört resultatet. För att eliminera störfrekvensen samplar man signalen med samplingsfrekvensen 8000 Hz och låter den passera ett digitalt filter (se figur 3). Därefter skickas den filtrerade signalen  $y[n]$  till en D/A-omvandlare för rekonstruktion, eftersom man vill återfå signalen på tidskontinuerlig form.

Konsulten ger dig i uppdrag att bestämma lämpliga värden på  $a_1$ ,  $a_2$  och  $a_3$  så att störsignalen försvinner samt att den vackra 2 kHz-tonens amplitud förblir oförändrad även efter D/A-omvandlingen (de andra frekvenserna behöver inte bibehålla sina respektive amplituder). (4p)

## 4

En svartvit bild  $B$  beskrivs av följande matris (0=svart, 1=vit)

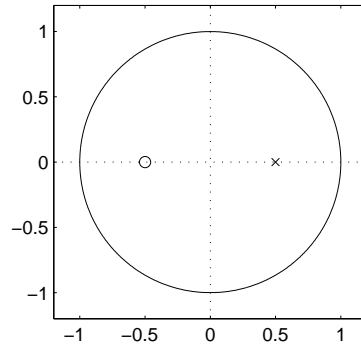
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
```

Bilden  $B$  filtreras (faltas) med filterkärnan  $K$ .

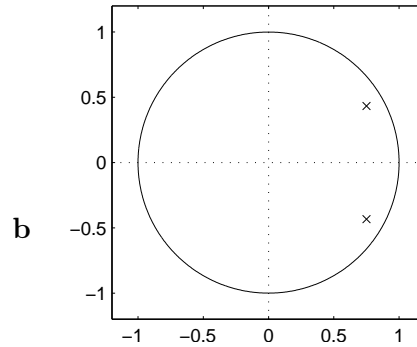
a Vad blir den resulterande bilden om  $K$  är

```
0 0 0
1 1 1
0 0 0
```

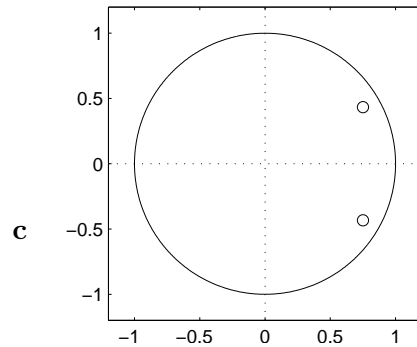
**1**  $y(n] = x(n] - x(n - 2] + \frac{3}{2}y(n - 1] - \frac{3}{4}y(n - 2]$  **a**



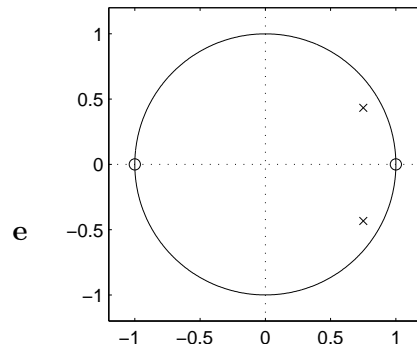
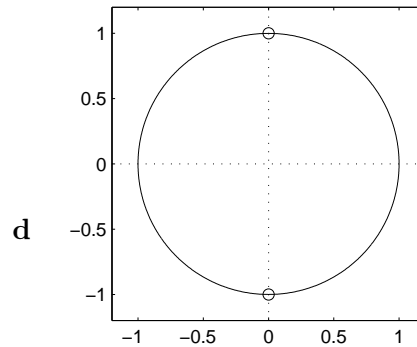
**2**  $y(n] = x(n] + \frac{3}{2}y(n - 1] - \frac{3}{4}y(n - 2]$  **b**



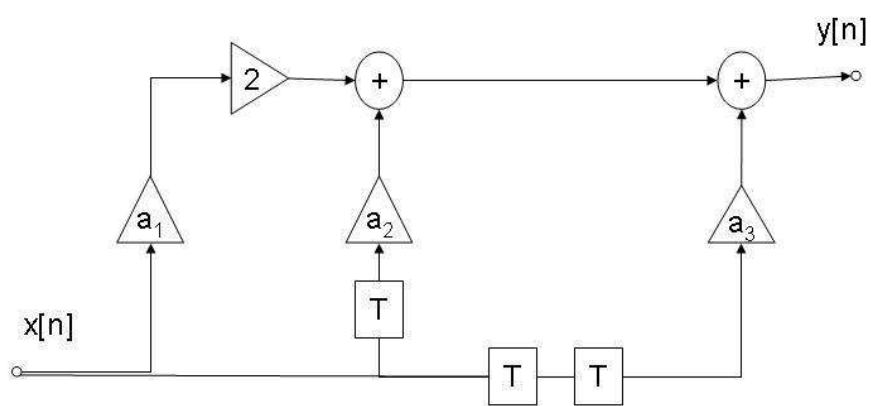
**3**  $y(n] = x(n] + x(n - 2]$  **c**



**4**  $y(n] = x(n] + 0.5x(n - 1] + 0.5y(n - 1]$  **d**



Figur 2.



Figur 3.

(1p)

b Vad blir den resulterande bilden om  $K$  är

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

(1p)

c Hur bör  $K$  se ut om bilden ska bli helt vit? (1p)

d Hur bör  $K$  se ut om bilden ska bli inverterad (svarta prickar på vit botten)? (1p)

## 5

När man beräknar den diskreta fouriertransformen (DFTn) av en sekvens  $x(n)$  där  $0 \leq n < N$ , så innebär det i praktiken att sekvensen skalärmultipliceras med var och en av DFTns basvektorer. Om signalen  $x(n)$  innehåller någon frekvens som matchar basvektorns frekvens, så kommer resultatet av skalärprodukten (i belopp) bli stort, i annat fall kommer det bli litet.

I det fall då insignalen är en cosinus-signal som *exakt* sammanfaller med någon av basvektorerna, så kommer skalärprodukten bli lika med  $N$  för den basvektor insignalen sammanfaller med, och 0 för alla andra basvektorer.

Basvektor  $k$  ges av

$$b_k(n) = e^{jnk2\pi/N} \quad (n = 0, 1, 2 \dots N - 1)$$

Låt nu  $x(n) = \cos(nm2\pi/N)$ , och visa att skalärprodukten

$$X_k = \sum_{n=0}^{N-1} x(n)e^{-jnk2\pi/N}$$

blir  $N$  då  $k = m$  samt 0 då  $k \neq m$

*Ledning: utnyttja Eulers formler (se formelblad) samt det faktum att  $e^{jk2\pi} = 1$  för alla heltal  $k$*

(4p)

## Lösningar

### 1

Vi kallar den verkliga vinkeln som hjulet roterar på  $\frac{1}{25}$  sekund för  $\phi$ . Bilen kommer då att tillryggälägga  $25\phi r$  meter på en sekund, dvs  $\phi = \frac{v}{25r}$  där  $v$  är bilens hastighet. Sätter vi in att  $r = 0.2$  får vi  $\phi = \frac{v}{5}$

- a** Vinkeln mellan två ekrar är  $\frac{2\pi}{5}$ . Så länge  $\phi$  är mindre än halva denna vinkel, kommer den upplevda rotationen  $\theta$  vara lika med  $\phi$ . Men alla jämna multiplar av  $\frac{2\pi}{5}$  som läggs till  $\phi$  kommer inte att märkas. Vidare kommer ett  $\phi$  strax under  $\frac{2\pi}{5}$  göra så att hjulet ser ut att rotera baklänges med, dvs  $\theta$  blir negativ  $\phi - \frac{2\pi}{5}$ .

$$v = \pi \Rightarrow \phi = \frac{\pi}{5}$$

vilket är precis halva vinkeln mellan två ekrar. Alltså kommer den rotation man uppfattar,  $\theta$ , vara lika med den verkliga rotationen, alltså  $\theta = \frac{\pi}{5}$ .

**b**  $v = 2\pi \Rightarrow \phi = \frac{2\pi}{5}$

Vilket är precis avståndet mellan två ekrar, dvs  $\theta = 0$ .

- c** Enl. resonemanget i **a** kan vi skriva

$$\phi = \theta + k \frac{2\pi}{5}$$

där  $k$  är ett godtyckligt heltal

Hastigheten  $v = 5\phi$  kan då skrivas som

$$v = 5\theta + k2\pi$$

### 2

- 1**  $y(n) = x(n) - x(n-2) + \frac{3}{2}y(n-1) - \frac{3}{4}y(n-2)$  ger överföringsfunktionen

$$H(z) = \frac{1 - z^{-2}}{1 - \frac{3}{2}z^{-1} + \frac{3}{4}z^{-2}}$$

Nollställena ges av  $z^2 - 1 = 0 \Rightarrow z = \pm 1$ , vilket bara stämmer in på plot **e**. Vi behöver alltså inte räkna ut polerna (upplysningsvis ligger de dock i  $z = \frac{\sqrt{3}}{2}(\frac{\sqrt{3}}{2} \pm \frac{j}{2})$ ).

- 2**  $y(n) = x(n) + \frac{3}{2}y(n-1) - \frac{3}{4}y(n-2)$  ger

$$H(z) = \frac{1}{1 - \frac{3}{2}z^{-1} + \frac{3}{4}z^{-2}} = \frac{z^2}{z^2 - \frac{3}{2}z + \frac{3}{4}}$$

Denna funktion har två poler, och nollställen endast i origo (vilka inte påverkar frekvensgången och inte ritas ut i plottarna i denna uppgift), alltså måste det vara plot **b**.

**3**  $y(n) = x(n) + x(n - 2) \Rightarrow H(z) = 1 + z^{-2} = \frac{z^2 + 1}{z^2}$

Nollställen:  $z = \pm j$ , alltså plot **d**.

**4**  $y(n) = x(n) + 0.5x(n - 1) + 0.5y(n - 1)$  ger

$$H(z) = \frac{1 + 0.5z^{-1}}{1 - 0.5z^{-1}} = \frac{z + 0.5}{z - 0.5}$$

Nollställe:  $z = -0.5$ , pol:  $z = 0.5$ , alltså plot **a**

**3**

Samplingsfrekvensen  $f_s$  kan ses som ett varv kring enhetscirkeln. Då motsvaras 1333 Hz av vinkeln  $\frac{\pi}{3}$  och 2000 Hz av vinkeln  $\frac{\pi}{2}$ .

För att eliminera störfrekvensen på 1333 Hz:  $|H(\omega)| = 0$  då  $z = e^{\pm \frac{j\pi}{3}}$ . Filtret kan alltså skrivas som

$$\begin{aligned} H(z) &= k(z - e^{\frac{j\pi}{3}})(z - e^{-\frac{j\pi}{3}}) = k(z^2 - ze^{-\frac{j\pi}{3}} - ze^{\frac{j\pi}{3}} + 1) = \\ &k(z^2 - 2z(\frac{e^{-\frac{j\pi}{3}} + e^{\frac{j\pi}{3}}}{2}) + 1) = k(z^2 - 2z \cos(\frac{\pi}{3}) + 1) = \\ &k(z^2 - z + 1)(\frac{z^{-2}}{z^{-2}}) = \frac{k(1 - z^{-1} + z^{-2})}{z^{-2}} \end{aligned}$$

För att 2 kHz-tonens amplitud skall förbli oförändrad:  $|H(\omega)| = 1$  då  $z = e^{\pm \frac{j\pi}{2}}$  ger att  $k = 1$ .  
Ur filterschemat:  $2a_1x[n] + a_2x[n - 1] + a_3x[n - 2] = y[n]$   
ger att  $H(z) = 2a_1 + a_2z^{-1} + a_3z^{-2}$ .

De båda filterekvationerna skall sammanfalla:

$$1 - z^{-1} + z^{-2} \longleftrightarrow 2a_1 + a_2z^{-1} + a_3z^{-2} \text{ ger svaret } a_1 = 0,5, a_2 = -1 \text{ och } a_3 = 1.$$

**4**

Faltingen av bilden kan beräknas enligt faltningssumman i formelsamlingen, eller så kan man resonera sig fram till resultatet, ungefär som nedan:

- a** Eftersom filterkärnan bara sprider sig horisontellt så kommer de tomma raderna att förbli tomma. Raderna som innehåller tända pixlar kommer däremot att delvis smetas ut. Punkterna mellan de tända pixlarna kommer att få bidrag från två håll och därför sättas till två, medan de tända pixlarna bara bidrar till sig själva och förblir ett. Bilden blir alltså horisontellt randig enligt mönstret

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 2 1 2 1 2 1 2 1 2 1 2 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 2 1 2 1 2 1 2 1 2 1 2 1
...
```

(Bilden kommer alltså att innehålla pixlar större än 1, detta hanteras i praktiken vid uppritningen antingen genom normalisering eller klippning.)



- b** Bilden kommer att bli diagonalt randig, enligt samma princip som ovan - de pixlar som ligger diagonalt mellan de tända pixlarna kommer få bidrag från två håll och sättas till två.
- c** Eftersom de tända pixlarna sitter på varannan position både i x- och y-led så kan vi enklast åstadkomma den önskade matrisen (helt fylld med ettor) med följande kärna:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Då kommer varje pixel i den nya bilden bara få bidrag från en tänd pixel i  $B$ . Om man önskar en symmetrisk filterkärna, så kan vi åstadkomma det genom att fördela om vikterna och erhålla

$$\begin{bmatrix} .25 & .5 & .25 \\ .5 & 1 & .5 \\ .25 & .5 & .25 \end{bmatrix}$$

vilket resulterar i samma bild (dock ej längs bildens kanter). Nu kommer pixlarna horisontellt och vertikalt mellan de tända pixlarna få två bidrag á 0.5, och pixlarna diagonalt mellan de tända få fyra bidrag á 0.25.

- d** Vi tar samma filterkärna som i **c** (båda varianterna går bra), men vi "släcker" centumpixeln:

$$\begin{bmatrix} .25 & .5 & .25 \\ .5 & 0 & .5 \\ .25 & .5 & .25 \end{bmatrix}$$

denna kärna ger alltså en bild som blir vit, utom där det tidigare varit en tänd pixel, där den blir svart.

## 5

Skriv om signalen enligt euler:

$$x(n) = \cos(nm2\pi/N) = \frac{e^{jnm2\pi/N} + e^{-jnm2\pi/N}}{2}$$

Då kan vi skriva skalärprodukten som

$$X_k = \sum_{n=0}^{N-1} \frac{e^{jnm2\pi/N} e^{-jnk2\pi/N} + e^{-jnm2\pi/N} e^{-jnk2\pi/N}}{2} = \sum_{n=0}^{N-1} \frac{e^{jn(m-k)2\pi/N} + e^{-jn(m+k)2\pi/N}}{2}$$

**Fallet  $m = k$ :**

$$X_k = \sum_{n=0}^{N-1} \frac{1 + e^{-jn2k2\pi/N}}{2} = \frac{N}{2} + \frac{1}{2} \sum_{n=0}^{N-1} e^{-jn2k2\pi/N}$$

Den andra delen i detta uttryck är en geometrisk serie (en summa av formen  $\sum a^n$  där i detta fallet  $a = e^{-j4k\pi/N}$ ) vilken enl. formelsamlingen kan beräknas till

$$\frac{e^{-j4k\pi} - 1}{e^{-j4k\pi/N} - 1}$$

$e^{-j4k\pi} - 1$  där  $k$  är ett heltal kommer alltid att bli noll, alltså blir  $X_k = \frac{N}{2}$  då  $m = k$

**Fallet  $m \neq k$ :**

Dela upp summan

$$\sum_{n=0}^{N-1} \frac{e^{jn(m-k)2\pi/N} + e^{-jn(m+k)2\pi/N}}{2}$$

i två summor

$$\frac{1}{2} \sum_{n=0}^{N-1} e^{jn(m-k)2\pi/N} + \frac{1}{2} \sum_{n=0}^{N-1} e^{-jn(m+k)2\pi/N}$$

och beräkna dessa som geometriska serier:

$$X_k = \frac{1}{2} \left( \frac{e^{j(m-k)2\pi} - 1}{e^{j(m-k)2\pi/N} - 1} + \frac{e^{j(m+k)2\pi} - 1}{e^{j(m-k)2\pi/N} - 1} \right)$$

Men täljarna  $e^{j(m-k)2\pi} - 1$  och  $e^{j(m+k)2\pi} - 1$  där  $k$  och  $m$  är heltal kommer alltid att bli noll, och nämnarna kommer att vara skilda från noll när  $m \neq k$ , alltså blir hela uttrycket

$X_k = 0$  då  $m \neq k$

*Q.E.D*