Digital Signal Processing

ESD5/IIV-elektro

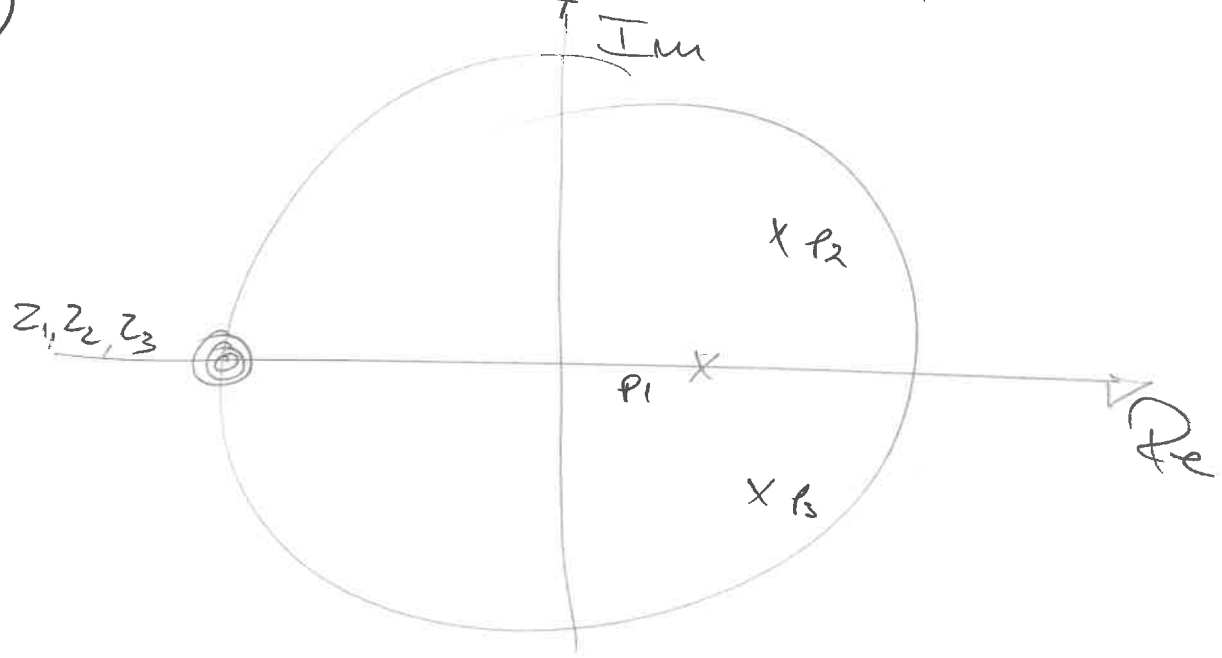
4. kursusgang – Løsnings-forslag

$$H(z) = 0.0317 \cdot \frac{1 + 3z^{-1} + 3z^{-2} + z^{-3}}{1 - 1.4590z^{-1} + 0.9104z^{-2} - 0.1978z^{-3}}$$

Find Zeros and Poles $\begin{cases} 3 \text{ zeros in } z = -1 \ (z_1, z_2, z_3) \\ 1 \text{ real pole in } z = 0.4142 \ (p_1) \\ \text{Complex conjugate pole pair} \\ z = 0.5224 \pm j\,0.4524 \ (p_2, p_3) \end{cases}$

$$H(z) = 0.0317 \cdot \frac{(z+1)^3}{(z-0.4142)(z-(0.5224+j0.4524))(z-(0.5224-j0.4524))}$$

a) Pole/Zero Diagram



PK

b) Vi søger nu et udtryk for amplitüderesponsen vha. vektorene $|\vec{V_i}|$, $i = \{1, \dots, 6\}$

$$\left| H(e^{j\omega}) \right| = 0.0317 \cdot \frac{|V_4| \cdot |V_5| \cdot |V_6|}{|V_1| \cdot |V_2| \cdot |V_3|} = 0.0317 \frac{|V_4|^3}{|V_1| \cdot |V_2| \cdot |V_3|}$$

Vi ønsker at lave et program, som kan üdtegne amplitüderesponsen i frekvens - intervallet 0..π. Derfor opstilles først üdtryk for modülüs af de fire vektorer.

$$|V_4| = \sqrt{(\cos\omega + 1)^2 + (\sin\omega)^2}$$

$$|V_1| = \sqrt{(\cos\omega - 0.5224)^2 + (\sin\omega - 0.4524)^2}$$

$$|V_2| = \sqrt{(\cos\omega - 0.5224)^2 + (\sin\omega + 0.4524)^2}$$

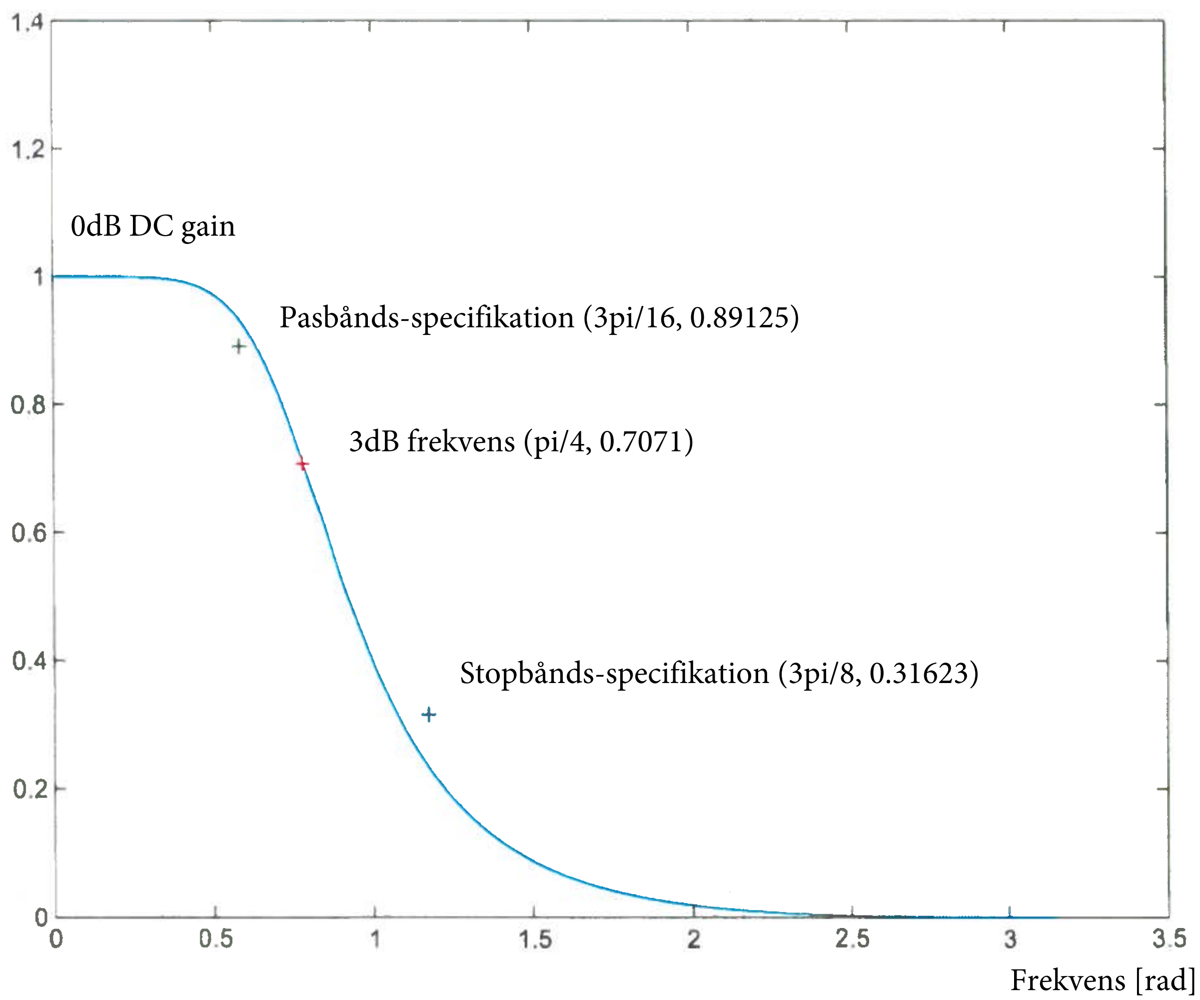
$$|V_3| = \sqrt{(\cos\omega - 0.4142)^2 + (\sin\omega)^2}$$

```
FOR i = 0..π STEP π/1000
    |V4| = ...;
    |V1| = ...;
    |V2| = ...;
    |V3| = ...;
    h(i) = 0.0317 · |V4|^3 / (|V1|·|V2|·|V3|);
ENDFOR;
```

```
% Dette MATLAB-program beregner amplituderesponsen
% af H(z) fremkommet ved bilineær transformation.

clear;

% Frekvens-sweep
for i=0:999,
  omega(i+1) = pi*i/999;
end;

% For hver værdi af omega beregnes amplituden.
for i=1:1000,
  % Først beregnes længden af vektorerne
  v4 = sqrt((cos(omega(i)) + 1)^2 + (sin(omega(i)))^2);
  v1 = sqrt((cos(omega(i)) - 0.5224)^2 + (sin(omega(i)) - 0.4524)^2);
  v2 = sqrt((cos(omega(i)) - 0.5224)^2 + (sin(omega(i)) + 0.4524)^2);
  v3 = sqrt((cos(omega(i)) - 0.4142)^2 + (sin(omega(i)))^2);
  % Herefter bestemmes amplituden
  h(i) = 0.0317 * ((v4)^3)/(v1 * v2 * v3);
end;

% og til slut plottes amplituderesponsen sammen med specifikationerne
plot(omega,h,omega(187),0.8913,'+',omega(250),1/sqrt(2),'+',omega(375),0.3162,'+');
```

c)

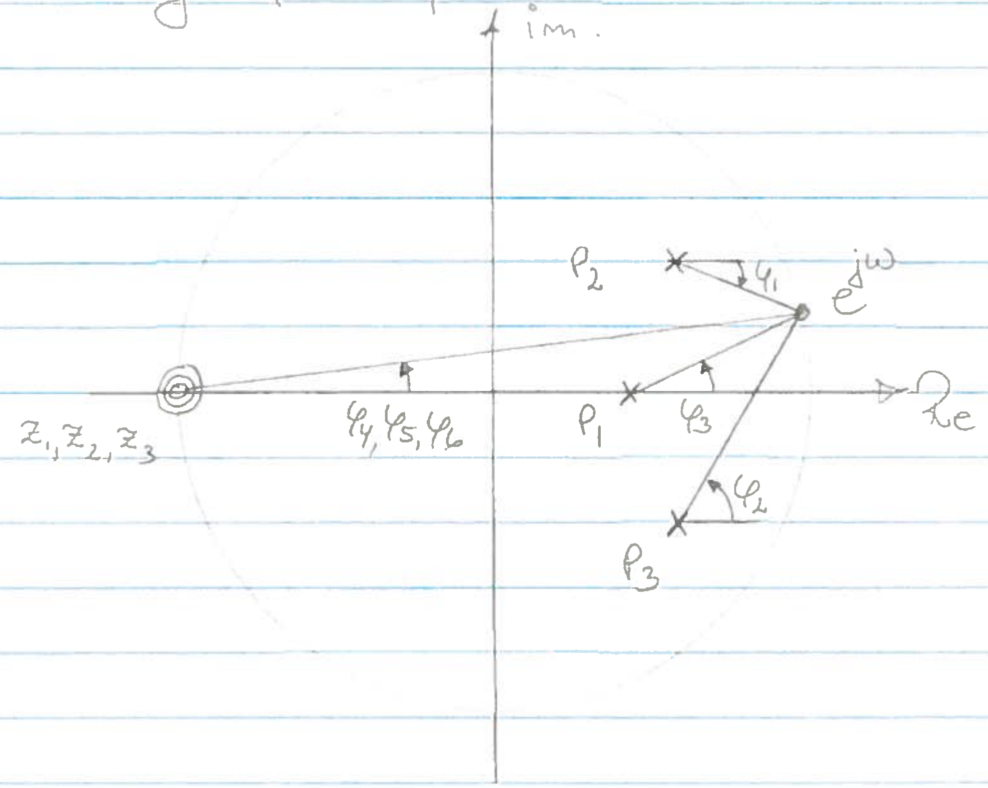
Monoton aftagende amplitude-response -- Butterworth-karakteristik for LP-filter



Ja, filteret overholder de opstillede design-specifikationer.

PK

d) Beregn faserespons:

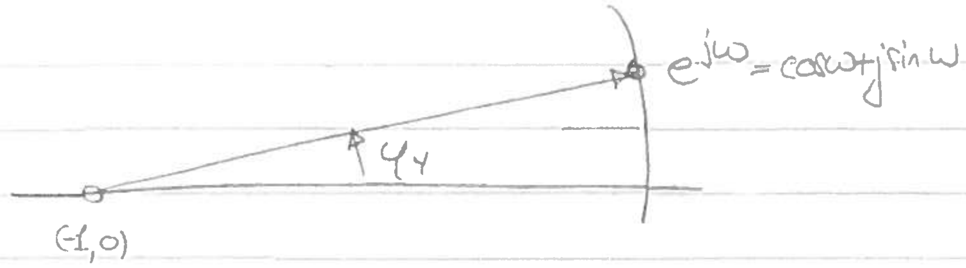


Vi søger nu et udtryk for fase responsen vha. vinklerne φ_i, $i = \{1..6\}$.

$$\underline{/H(e^{j\omega})} = \operatorname{Arg}(0.0317) + \underbrace{\sum_{i=4}^{6} \varphi_i}_{\text{Argumentet fra nulpunkterne}} - \underbrace{\sum_{j=1}^{3} \varphi_j}_{\text{Argumentet fra polerne}} = 3\cdot\varphi_4 - \sum_{j=1}^{3}\varphi_j$$

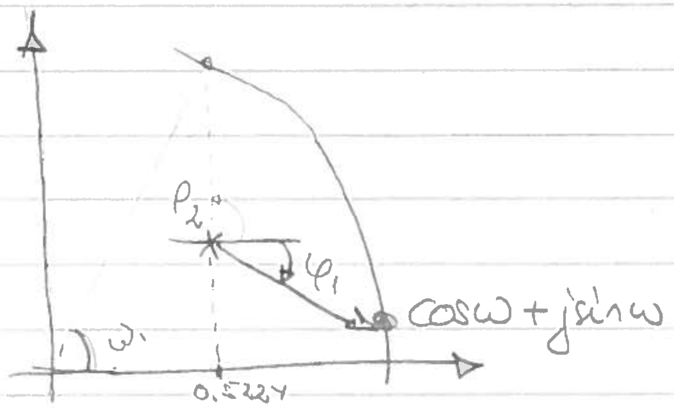
Vi ønsker at lave et program, som kan udtegne faseresponsen i frekvensintervallet $0..\pi$. Derfor opstilles først udtryk for de fire vinkler

$\underline{\underline{\varphi_4}}$



$$\tan\varphi_4 = \frac{\sin\omega}{\cos\omega + 1} \Rightarrow \varphi_4 = \arctan\left(\frac{\sin\omega}{\cos\omega+1}\right)$$

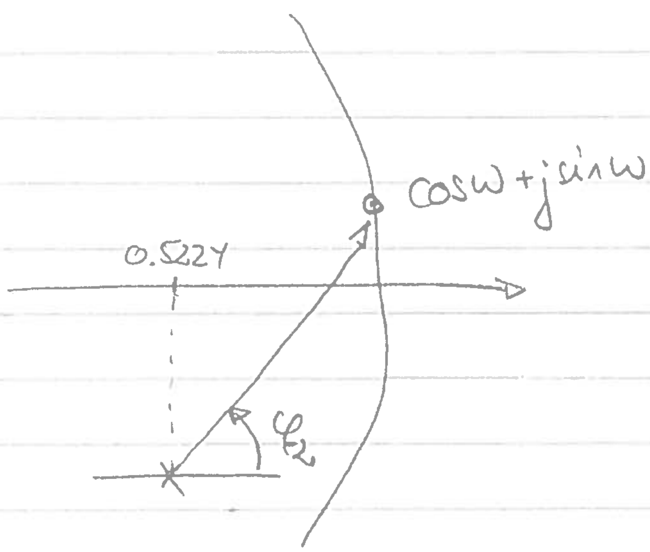
$\underline{\underline{\varphi_1}}$



$$\tan\varphi_1 = \frac{\sin\omega - 0.4524}{\cos\omega - 0.5224} \Rightarrow \varphi_1 = \arctan\left(\frac{\sin\omega - 0.4524}{\cos\omega - 0.5224}\right)$$

OBS: bemærk at φ_1 „springer" $-\pi$ når ω passerer $\arccos(0.5224)$!

$\underline{\underline{\varphi_2}}$



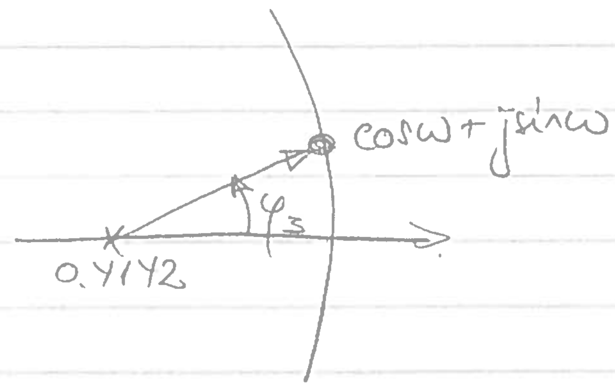
$$\tan\varphi_2 = \frac{\sin\omega + 0.4524}{\cos\omega - 0.5224}$$

$$\Downarrow$$

$$\varphi_2 = \arctan\left(\frac{\sin\omega + 0.4524}{\cos\omega - 0.5224}\right).$$

OBS. φ_2 springer $-\pi$ för $\omega = \arccos(0.5224)$

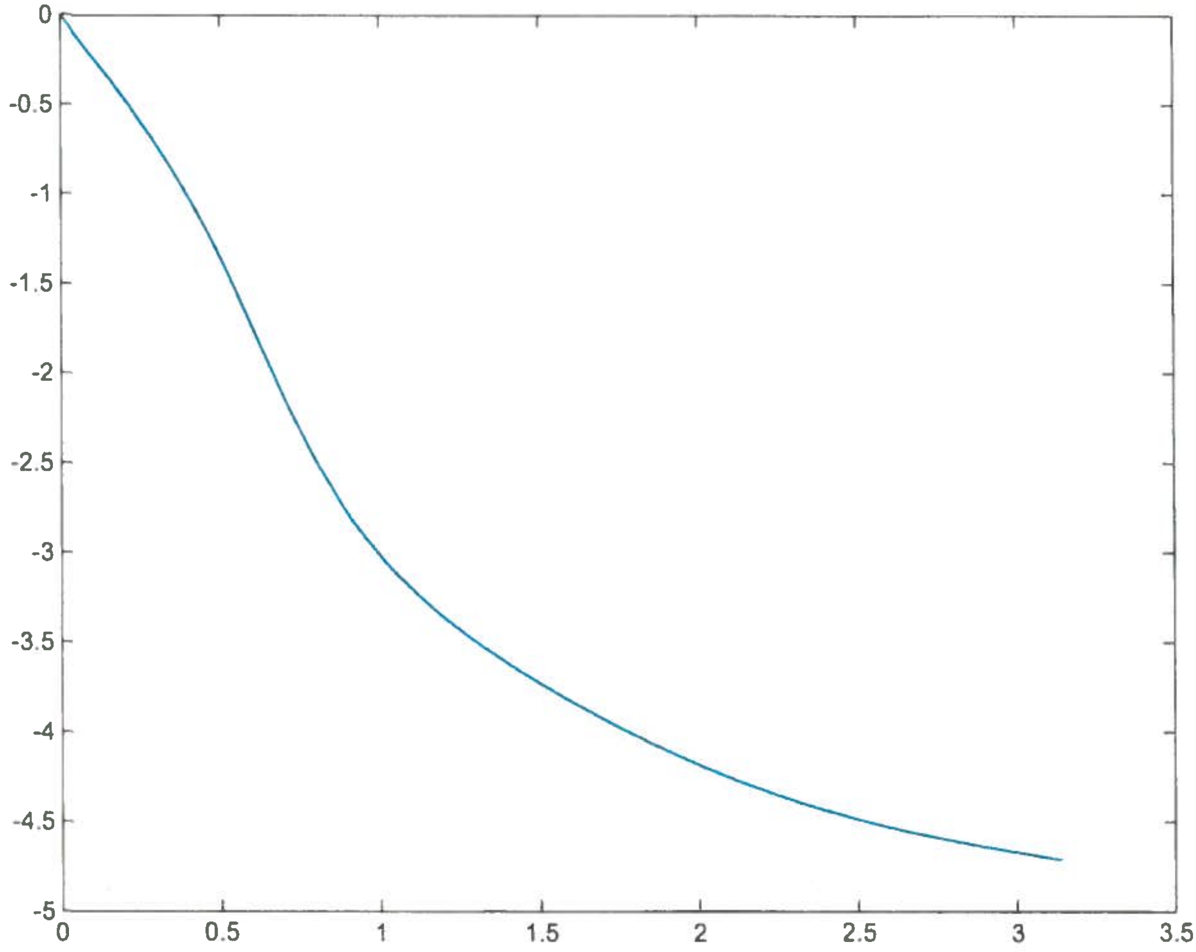
φ_3



$$\tan\varphi_3 = \frac{\sin\omega}{\cos\omega - 0.4142} \Rightarrow \varphi_3 = \arctan\left(\frac{\sin\omega}{\cos\omega - 0.4142}\right)$$

PROGRAMMER i MATLAB!

```
% Dette MATLAB-program beregner amplituderesponsen
% af H(z) fremkommet ved bilineær transformation.

clear;

% Frekvenssweep i intervallet 0..pi, 1000 samples
for i=1:1000;
  w(i) = (pi/1000)*i;
end;

% Beregning af vinklen hidrørende fra nulpunkterne
for i=1:999,
  phi4(i) = atan(sin(w(i))/(cos(w(i))+1));
end;
phi4(1000)=pi/2;

% Beregning af vinklen fra pol i 0.5224+j0.4524
for i=1:1000,
  if w(i) < acos(0.5224), % Der tages højde for spring i Arctan
     phi1(i) = atan(sin(w(i)-0.4524)/(cos(w(i))-0.5224));
  else
     phi1(i) = atan(sin(w(i)-0.4524)/(cos(w(i))-0.5224)) + pi;
  end
end;

% Beregning af vinklen fra pol i 0.5224-j0.4524
for i=1:1000,
  if w(i) < acos(0.5224), % Der tages højde for spring i Arctan
     phi2(i) = atan(sin(w(i)+0.4524)/(cos(w(i))-0.5224));
  else
     phi2(i) = atan(sin(w(i)+0.4524)/(cos(w(i))-0.5224)) + pi;
  end
end;

% Beregning af vinklen fra pol i 0.4142
for i=1:1000,
  if w(i) < acos(0.4142), % Der tages højde for spring i Arctan
     phi3(i) = atan(sin(w(i))/(cos(w(i))-0.4142));
  else
     phi3(i) = atan(sin(w(i))/(cos(w(i))-0.4142)) + pi;
  end
end;

%Beregning af den samlede fasevinkel
for i=1:1000,
  vinkel(i)=3*phi4(i) - (phi1(i)+phi2(i)+phi3(i));
end;

plot(w,vinkel);
```


0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
0
0.5
1
1.5
2
2.5
3
3.5

2) Eftersom $H(z)$ har samtlige sine nulpunkter placeret i $z=-1$, så trækkes amplitude-karakteristikken med $-\infty$ for $\omega \to \pi$. Dette kunne indikere, at $H(z)$ er frembragt vha den bilineære transformation, hvilket også er tilfældet.

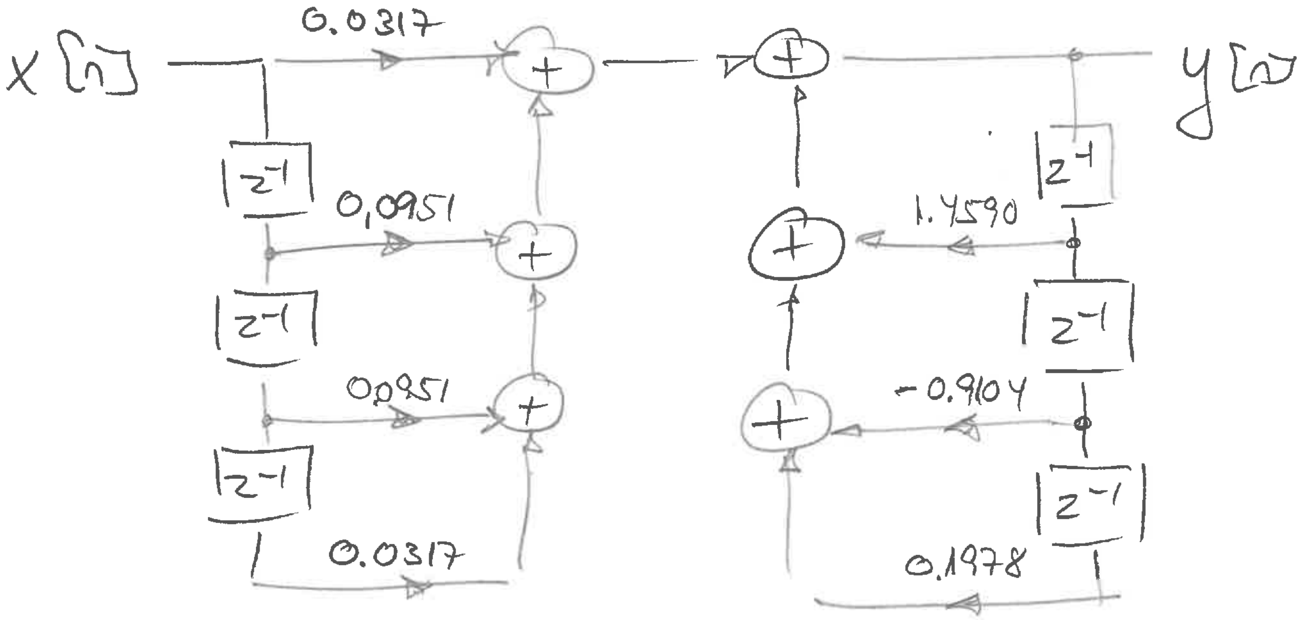
3)

$$H(z) = \frac{Y(z)}{X(z)} = 0{,}0317 \cdot \frac{1 + 3z^{-1} + 3z^{-2} + z^{-3}}{1 - 1.4590z^{-1} + 0{,}9104z^{-2} - 0{,}1978z^{-3}}$$

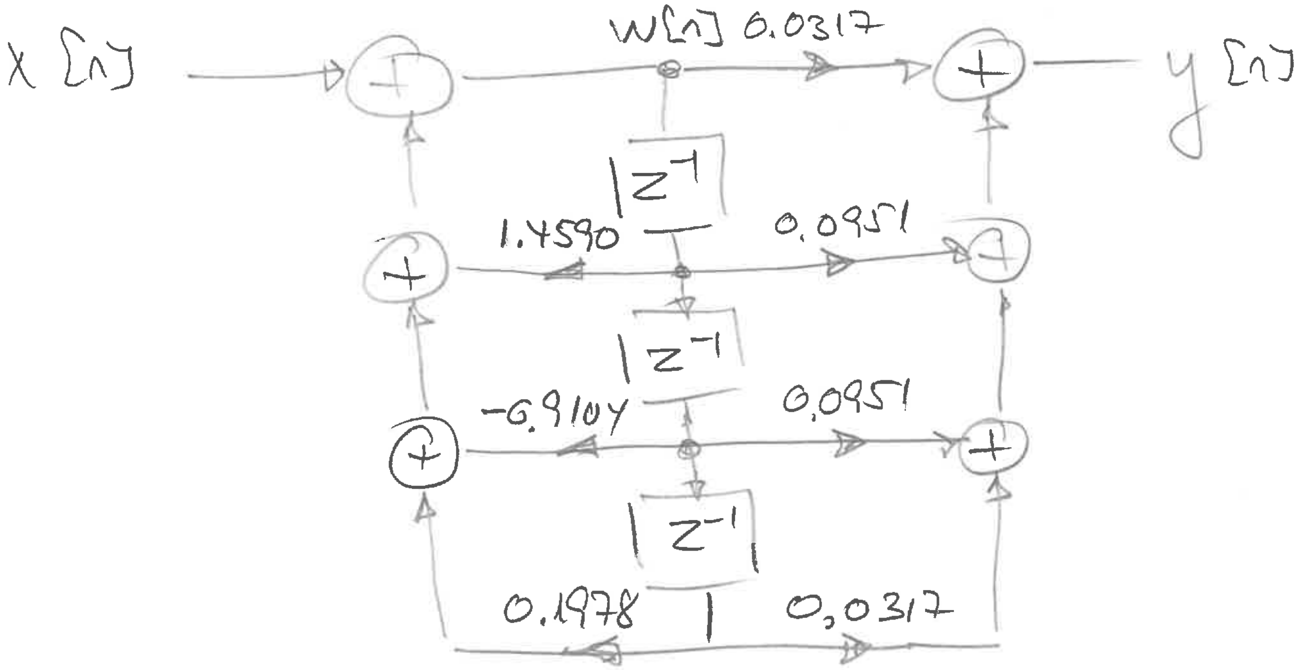
$\Downarrow$

$$y[n] = 1.4590\,y[n-1] - 0{,}9104\,y[n-2] + 0.1978\,y[n-3] + 0{,}0317\left(x[n] + 3x[n-1] + 3x[n-2] + x[n-3]\right)$$

4) Direct Form I



Direct Form II



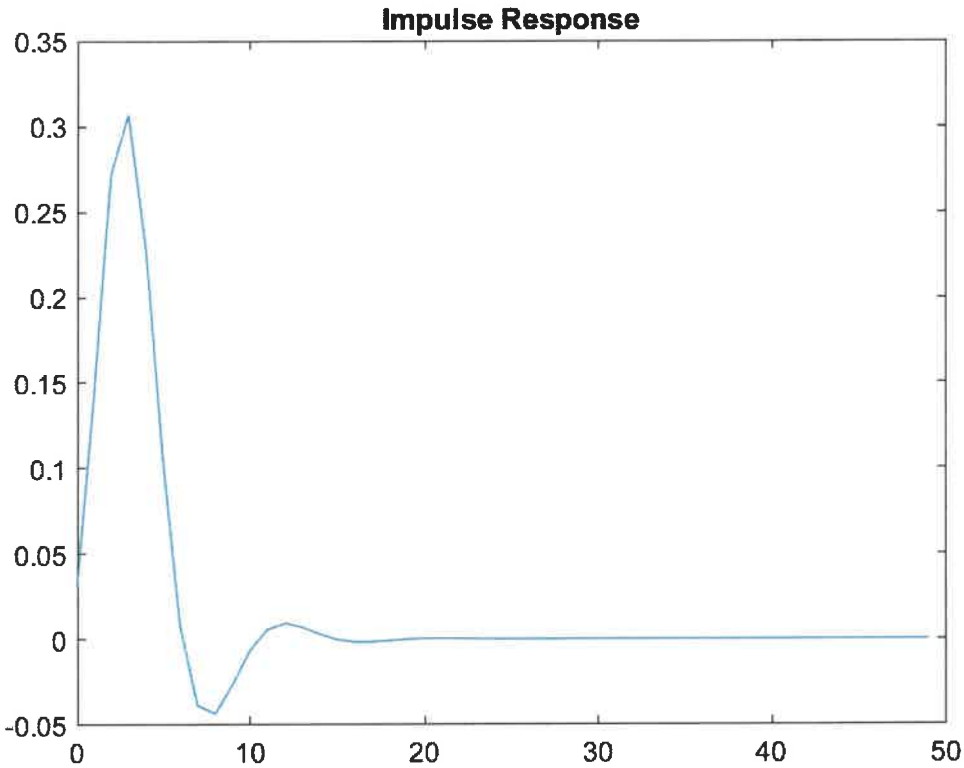
5) Her skal vi huske, at et delay-element (z^{-1}) i vores data flow graf bliver realiseret i form af et register i vores HW/SW-realisation. Så spørgsmålet er, hvordan vi vil initiere disse registre ...?

Den umiddelbare løsning vil være at nulstille samtlige disse registre inden algoritme-afviklingen påbegyndes.

Men i praksis viser det sig at være uden betydning, hvilken initial værdi hvert register indeholder.

Sagen er nemlig den, at efter relativt få sample-perioder vil systemet (filteret) „svinge ind" og dermed elimineres begyndelsesbetingelserne efter ganske kort tid — her underforstået, at samplefrekvensen er i kHz-området, eller højere.

```
% Forslag til Matlab-program, som kan beregne filteres impulsrespons vha.
% Direct Form II strukturen.

clear

% Først defineres og nulstilles de interne variable
w_1=0;
w_2=0;
w_3=0;

% Vi beregner N samples af impulsresponse
N=50;

for n=1:N
    % Beregn værdien af w[n]
    % Hvis n=0 er x[n]=1, ellers 0, altså en impuls
    if (n-1) == 0
        x=1;
    else
        x=0;
    end

    w(n) = x + 1.4590*w_1 - 0.9104*w_2 + 0.1978*w_3;

    % Og nu beregnes output
    y(n) = 0.0317*w(n) + 0.0951*w_1 + 0.0951*w_2 + 0.0317*w_3;

    % Og til slut opdates delay line
    w_3 = w_2;
    w_2 = w_1;
    w_1 = w(n);

    % tidsakse
    tid(n)=n-1;
end

plot(tid,y)
```


Impulse Response
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
10
20
30
40
50