

# Labo Signaalverwerking

Dries Kennes (R0486630)

May 25, 2018

## Opdracht S2: Chebyshev filter

### Specificatie

- Type: Chebyshev
- $|H(0)| = 3dB$
- $A_p = 3dB$
- $A_s = 40dB$
- $f_c = 2kHz$
- $f_s = 6,2kHz$

### Synthese & Analyse (MATLAB)

```
1 % Gegevens
2 K = sqrt(2); % 3dB
3 Ap = 3; % 3dB
4 As = 40; % 40dB
5 fc = 2000; % 2kHz
6 fs = 6200; % 6.2kHz
7
8 % 's' nodig om analoge filter te maken
9 % Bepaal order
10 [n, wp] = cheblord(2*pi*fc, 2*pi*fs, Ap, As, 's') % OUTPUT: n = 3 wp = 1.2566e+04
11 % Bepaal H (als Teller en Noemer)
12 [T, N] = cheby1(n, Ap, wp, 's')
13 % OUTPUT:
14 % T = 1.0e+11 * [0 0 0 4.9728]
15 % N = 1.0e+11 * [0 0 0.0015 4.9728]
16 % K Niet vergeten!
17 filter = K * tf(T, N)
18 %OUTPUT: filter=
19 % 9.946e11
20 % -----
21 % s^3 + 7505 s^2 + 1.466e08 s + 4.973e11
22
23 % Zien of het klopt
24 figure(1); clf; hold on; bode(filter);
25 figure(2); clf; hold on; step(filter);
26 figure(3); clf; hold on; axis equal; pzmap(filter);
27
28 trappen = zpk(filter)
29 %OUTPUT: trappen =
30 % 9.9456e+11
31 % -----
32 % (s+3753) (s^2 + 3753s + 1.325e08)
33
34 [r, p, k] = zpkmdata(trappen)
35
36 % zeta = 1 bij 1e trap
37 [wn, zeta] = damp(filter)
38 % OUTPUT:
39 % wn = 1.0e+04 * [0.3753 1.1512 1.1512]
40 % zeta = [1.0000 0.1630 0.1630]
41 ileorde = 1;
42 i2eorde = 2;
43
```

```

44 H1N = [ 0 1];
45 H1D = [1/wn(i2eorde) 1];
46
47 H1 = tf(H1N, H1D)
48 % OUTPUT: H1=
49 % 1
50 % -----
51 % 0.0002665 s + 1
52 figure(1); bode(H1);
53 figure(2); step(H1);
54 figure(3); pzmap(H1);
55
56 Q2 = 1/(2*zeta(i2eorde))
57 wn2 = wn(i2eorde)
58 H2N = [0 0 1];
59 H2D = [1/wn2^2 1/(Q2 *wn2) 1];
60
61 % K meenemen
62 H2 = K * tf(H2N, H2D)
63 % OUTPUT: H2=
64 % 1
65 % -----
66 % 7.546e-09 s^2 + 2.832e-05 s + 1
67 figure(1); bode(H2);
68 figure(2); step(H2);
69 figure(3); pzmap(H2);
70
71 % Ontwerp VGL
72 C1 = 1 % OUTPUT: C1 = 1
73 R = 1/(wn2*C1) % OUTPUT: R = 8.6869e-05
74 R1 = R/K % OUTPUT: R1 = 6.1426e-05
75 R2 = Q2e/(wn2*C1) % OUTPUT: R2 = 2.6648e-04
76
77 %Realistische waarden
78 ISF = 10^8;
79 C1 = C1/ISF % OUTPUT: C1 = 1.0000e-08
80 R1 = R1*ISF % OUTPUT: R1 = 6.1426e+03
81 R2 = R2*ISF % OUTPUT: R2 = 2.6648e+04
82 R = R*ISF % OUTPUT: R = 8.6869e+03
83
84 H2Nc = [0 0 (R/R1)*(1/(R^2*C1^2))];
85 H2Dc = [1 1/(R2*C1) 1/(R^2*C1^2)];
86 H2c = tf(H2Nc, H2Dc)
87
88 figure(1); bode(H2c); %H2c moet hier op H2 liggen
89 figure(2); step(H2c);
90 figure(3); pzmap(H2c);
91
92 % Dit zou gelijk moeten zijn aan filter
93 %total = H1*H2;
94 %figure(1); bode(total);
95 %figure(2); step(total);
96 %figure(3); pzmap(total);

```

Pole Zero plot

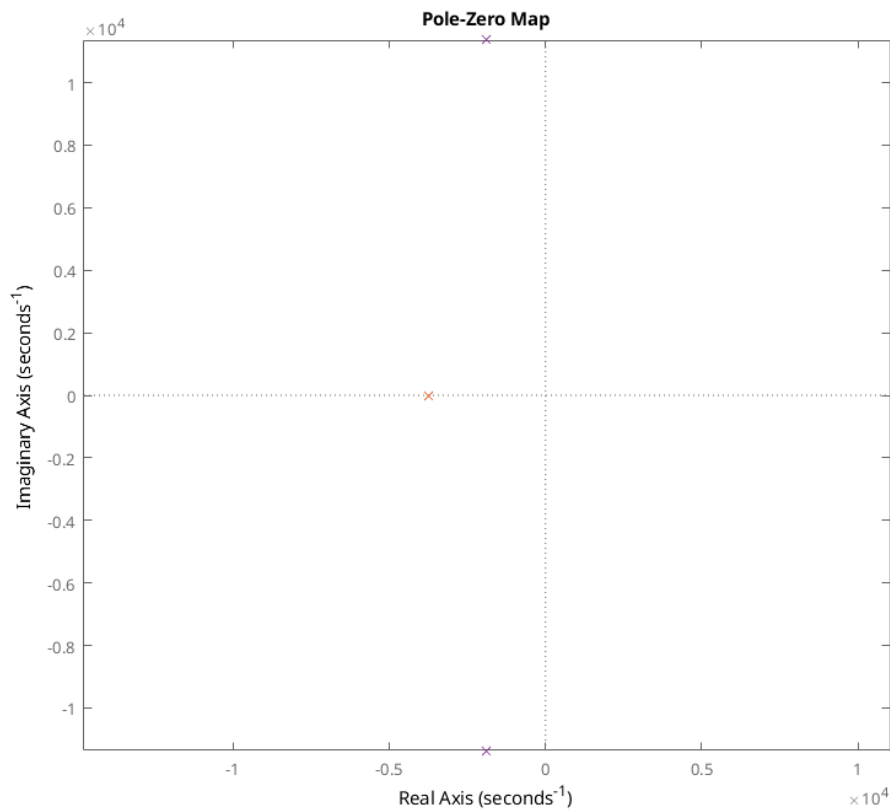


Figure 1: Pole zero plot

Bode plot

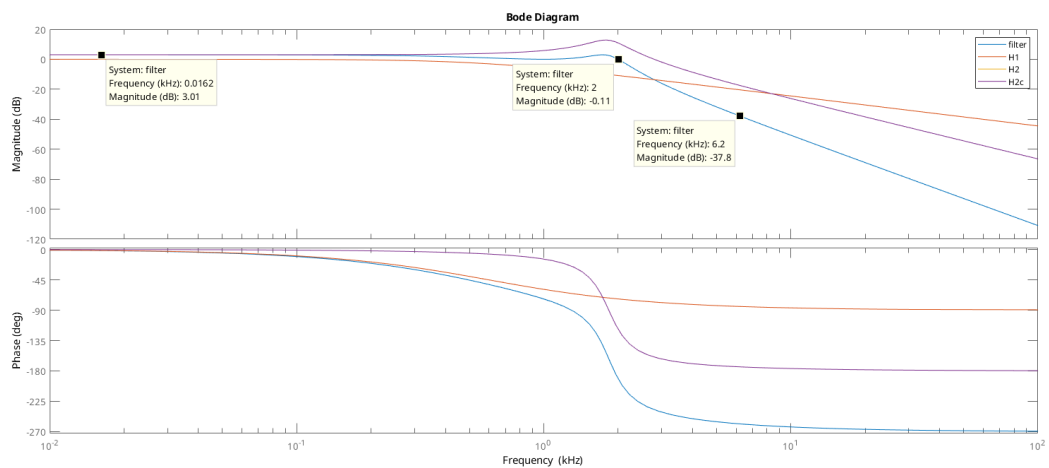


Figure 2: Bode Plot

## Stapresponsie

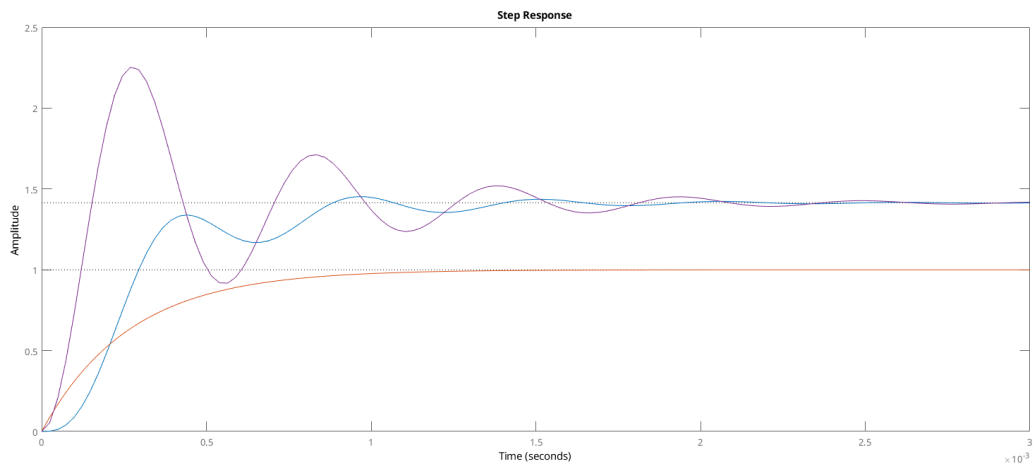


Figure 3: Stapresponsie

## Synthese 2e actieve filtertrap Biquad

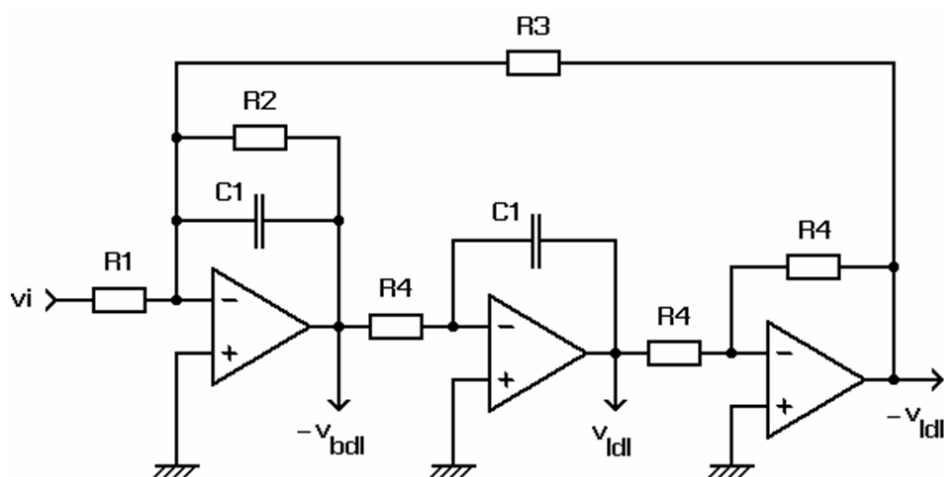


Figure 4: Het Biquad schema

$$H(s) = \frac{\frac{R3}{R1} \frac{1}{R3 \cdot R4 \cdot C1^2}}{s^2 + s\left(\frac{1}{R2 \cdot C1}\right) + \frac{1}{R3 \cdot R4 \cdot C1^2}}$$

## Ontwerpvergelijkingen

Staan in cursus Signaalverwerking, pagina CMT17-CMT18.

$$C1 = 1$$

$$R1 = \frac{1}{K\omega_n}$$

$$R2 = \frac{Q}{\omega_n}$$

$$R3 = R4 = R = \frac{1}{\omega_n}$$

MATLAB code staat bij in het eerste deel, samen met de uitkomsten van de bewerkingen.

# SPICE

## Netlist

```
1 * Bode (LTSpice export)
2 .inc opampIdeaal.cir
3 R1 N001 vin 6.1426K
4 V1 vin 0 AC 1
5 XU1 N001 0 N002 opampIdeal
6 C1 N002 N001 1e-8
7 R2 N002 N001 26.648K
8 R3 vout N001 8.6869K
9 XU2 N003 0 N004 opampIdeal
10 XU3 N005 0 vout opampIdeal
11 R4 N003 N002 8.6869K
12 R5 N005 N004 8.6869K
13 R6 vout N005 8.6869K
14 C2 N004 N003 1e-8
15 .ac dec 100 100 1MEG
16 .probe
17 .end
```

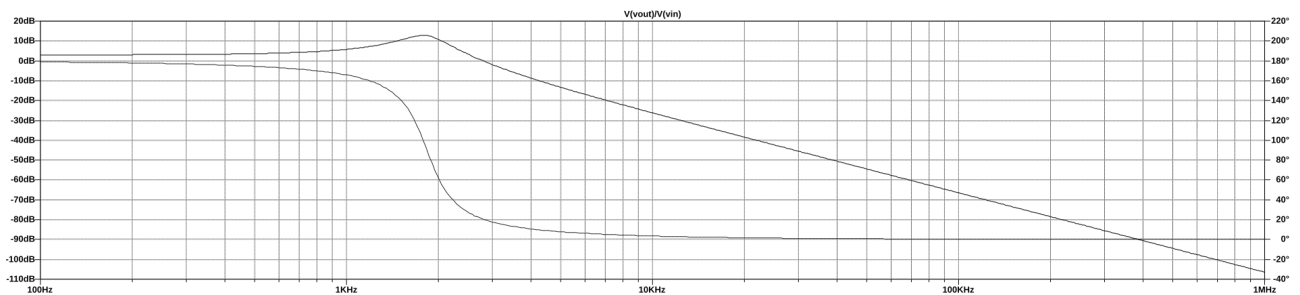


Figure 5: Bode Plot

==TODO== Monte Carlo analyse

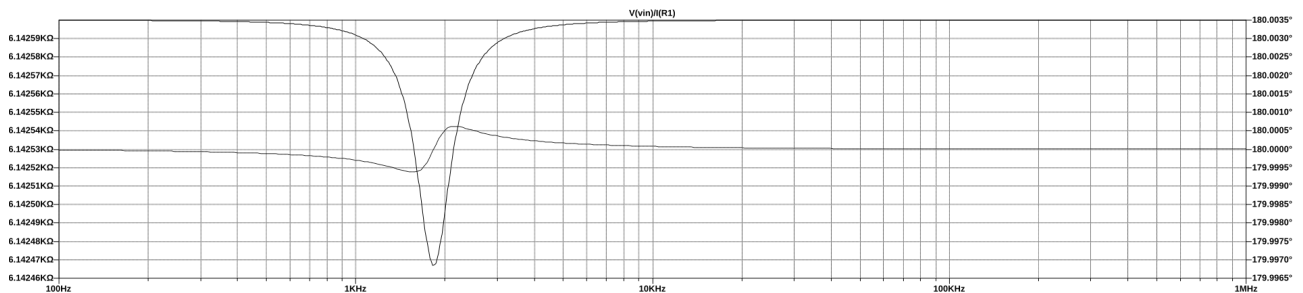


Figure 6: Ingangsimpedantie

Er is bijna geen variatie in de ingangsimpedantie.

==TODO== Uitgangsimpedantie