

# Cauer Filter float/double test für Vincent H.

## Initialisierung eines Standard-IIR Filters lt. Cauer.hpp

```
b0 = 0.45438943722126651;  
b1 = -0.87475628578418418;  
b2 = -0.033942738993371697;  
b3 = 0.87475628630511904;  
b4 = -0.4204466977069598;  
  
a1 = -3.782529328753677;  
a2 = 5.3516989219674223;  
a3 = -3.3558079291423386;  
a4 = 0.7866383359343796;
```

Für Matlab brauchen wir den Vector der Koeffizienten

```
b = [b0 b1 b2 b3 b4];  
a = [1 a1 a2 a3 a4];
```

Also  $H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots}$ . Passt das überein mit deinen Vorstellungen?

## Stabilitätstest, 64 Bit float

Nullstellen:

```
z = roots(b)
```

```
z = 4x1  
      -1  
      0.99999  
      0.99758  
      0.92755
```

Polstellen:

```
p = roots(a)
```

```
p = 4x1  
      1  
      0.99952  
      0.97962  
      0.80339
```

Die erste Polstelle ist schon gefährlich nah an 1. Stabilitätskriterium ist  $|p_i| < 1$ ,  $i = 1, \dots, 4$ .

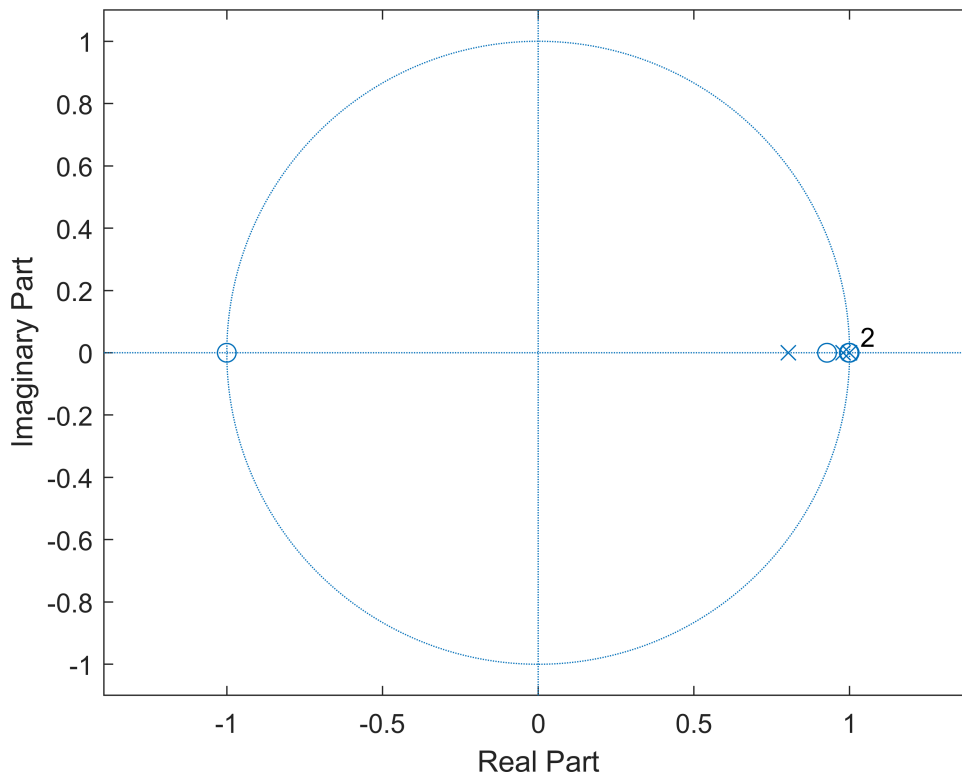
Matlab sagt zwar, dass das Filter Stabil ist, aber ich denke es ist knapp:

```
isstable(b,a)
```

```
ans = logical  
      1
```

Plot der komplexen Ebene. x sind Polstellen, Kreise Nullstellen:

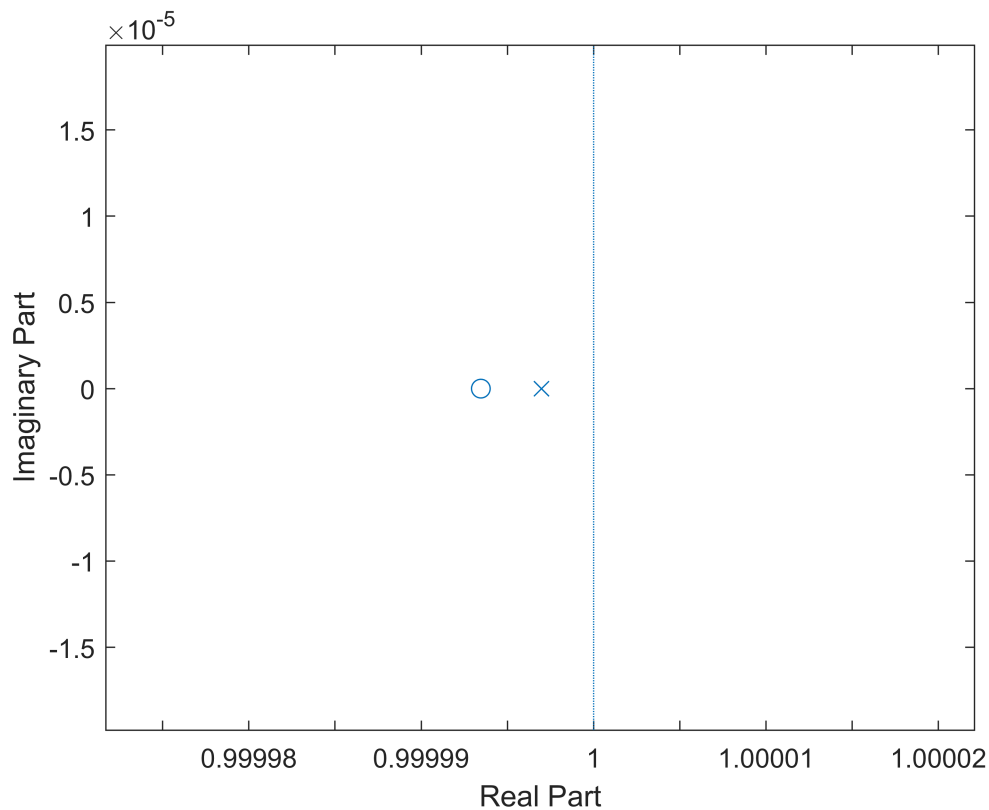
```
zplane(b,a)
```



```
zplane(b,a)
```

Man muss schon verdammt nah dran gehen um es zu erkennen. Ich denke, dass hier schon Probleme entstehen können.

```
xlim([0.9999717 1.0000221])  
ylim([-0.0000198 0.0000199])
```



Der Abstand des Poles zu 1 beträgt

```
(1-max(abs(p)))
```

```
ans =  
3.0193e-06
```

Der Abstand zwischen Pol und Zero beträgt.

```
p(1)-z(2)
```

```
ans =  
3.5229e-06
```

Ich bin mir nicht sicher, ob das schon was ausmacht, und ob Rundungsfehler ggf. sogar den Pol mit der Nullstelle kompensieren.

## 32 Bit Float Analyse

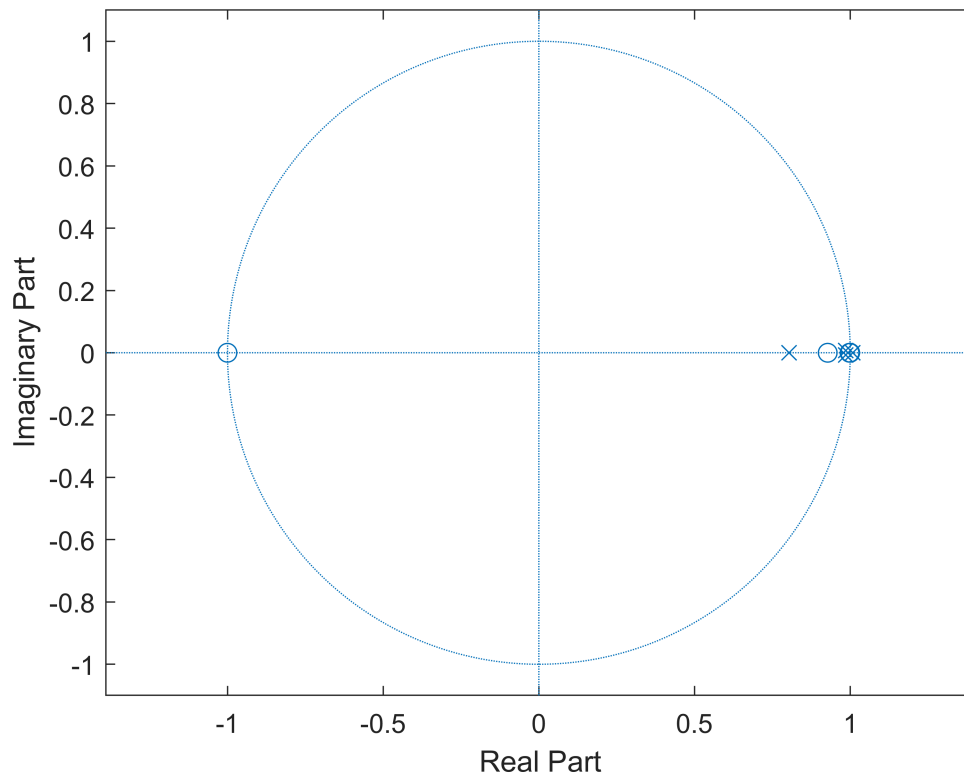
Mal gucken, was bei Float passiert:

```
aSingle = single(a);  
bSingle = single(b);  
  
isstable(bSingle,aSingle)
```

```
ans = Logical  
0
```

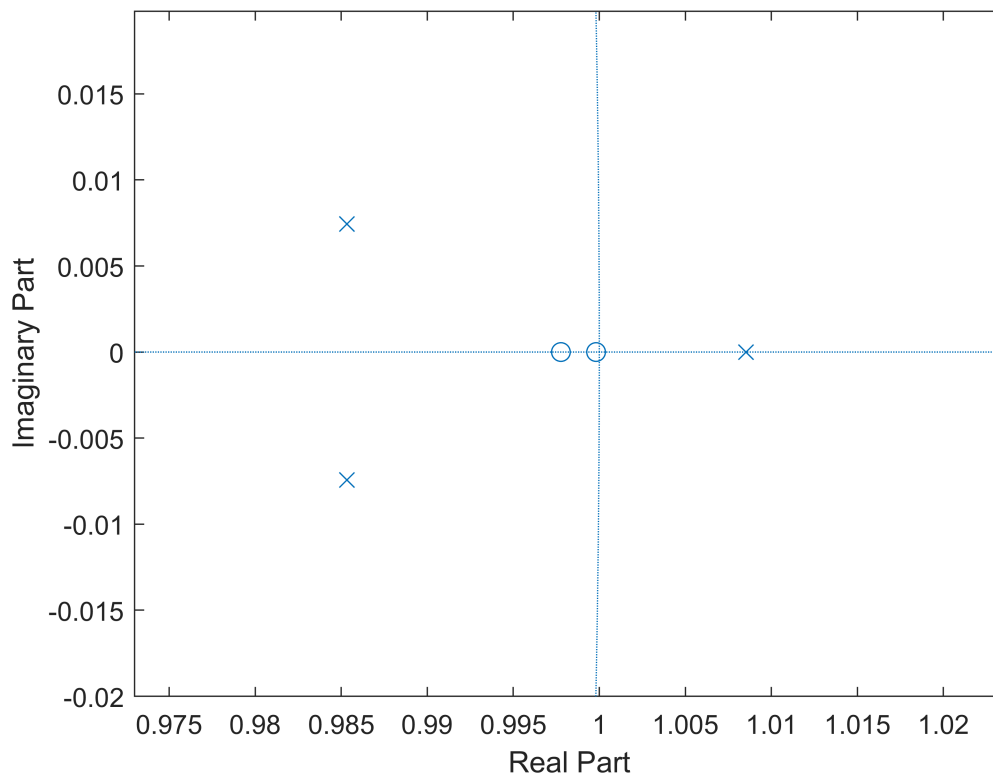
Ups, Überraschung...

```
zplane(bSingle,aSingle)
```



```
zplane(bSingle,aSingle)
```

```
xlim([0.9730 1.0235])  
ylim([-0.0200 0.0198])
```



Der Pol ist außerhalb des Einheitskreises und damit instabil. Du hast also Recht, es liegt an der Floating Point Genauigkeit.

Offensichtlich reicht es nicht, die in double berechneten Pole nach Float zu konvertieren, sondern die Koeffizienten. Da geht bei der Berechnung der Nullstellen was kaputt.

## Umbau auf biquads + Erhöhung der Filterordnung

```
% original second order sections
[sos] = tf2sos(b,a)
```

```
sos = 2x6
    0.45439    0.032921   -0.42147    1   -1.783    0.78702
         1   -1.9976    0.99757    1   -1.9995    0.99952
```

```
% fitten der impulsantwort und Erhöhen der Pole
[bb,aa] = stmcb(impz(sos),5,4);
sos2 = tf2sos(bb,aa);
sos2 = single(sos2)
```

```
sos2 = 3x6 single matrix
    0.4544    0.45438         0    1   -0.80339         0
         1   -1.9251    0.92531    1   -1.9791    0.97915
         1   -0.99999   -3.0427e-07    1        -1         0
```

```
% analyse
```

```
filtord(sos)
```

```
ans =  
     4
```

```
filtord(sos2)
```

```
ans =  
     5
```

```
isstable(sos2) % wheeee
```

```
ans = logical  
     1
```

```
% plot  
t = 5e4;  
  
figure,  
subplot(211)  
hold all, box on, grid on,  
title('Step response')  
h = gca;  
  
[resp1,t1] = stepz(sos,t);  
plot(h,t1,resp1,'linewidth',2);  
  
[resp2,t2] = stepz(single(sos),t);  
plot(h,t2,resp2,'color',[.5 .5 .5]);  
  
[resp3,t3] = stepz(single(b),a,t);  
plot(h,t3,resp3,'color',[.7 .7 .7]);  
  
[resp4,t4] = stepz(sos2,t);  
plot(h,t4,resp4,'--','linewidth',2,'color','k');  
  
ylim(h,[-1 300])  
xlim([-10,t])  
legend('double sos 2 sections','single sos 2 sections','single b,a', 'single sos 3 sections')  
  
subplot(212), hold all, box on, grid on  
title('Error of single vs double model')  
plot(t1,resp1-resp4,'linewidth',1.5)  
legend('double sos 2 sections - single sos 3 sections')  
xlabel('Samples')
```

