

Betrachtung eines Tschebyscheff-TP --- 6.Mai 2007 Ingenieurbüro Baumann, Dorsten

- `reset():ta:=time():DIGITS:=32:w:=2*PI*f:`

die Eingangsdaten

- `RippledB:=3:n:=100:fg:=10e3:ue2:=1:`

die Berechnungen

- `epsilon:=sqrt(10^(0.1*RippledB)-1):wg:=2*PI*fg:`
- `gam:=1/n*arcsinh(1/epsilon):`
- `if frac(n/2)=0 then`
 - `k:=n/2:`
 - `b:=[1/(cosh(gam)^2-cos((2*i-1)*PI/2/n)^2) $ i=1..k]:`
 - `a:=[2*b[i]*sinh(gam)*cos((2*i-1)*PI/2/n) $ i=1..k]:`
- `else`
 - `k:=(n+1)/2:`
 - `b:=[0]:a:=[1/sinh(gam)]:`
 - `Liste:=[1/(cosh(gam)^2-cos((i-1)*PI/n)^2) $`
 - `i=2..k]:b:=b.Liste:delete Liste:`
 - `Liste:=[2*b[i]*sinh(gam)*cos((i-1)*PI/n) $`
 - `i=2..k]:a:=a.Liste:delete Liste:`
- `end_if:`

die Koeffizienten für quadratische Glieder ($T(p) = A0 / [\text{product}(1 + a_i * p/wg + b_i * (p/wg)^2)]$) bei Normierung auf fg bei Amin

- `delete aQuad,bQuad:for i from 1 to k do`
 - `aQuad[i]:=float(op(a,i)):`
 - `bQuad[i]:=float(op(b,i)):`
- `end_for:`
- `aQuad;bQuad;`

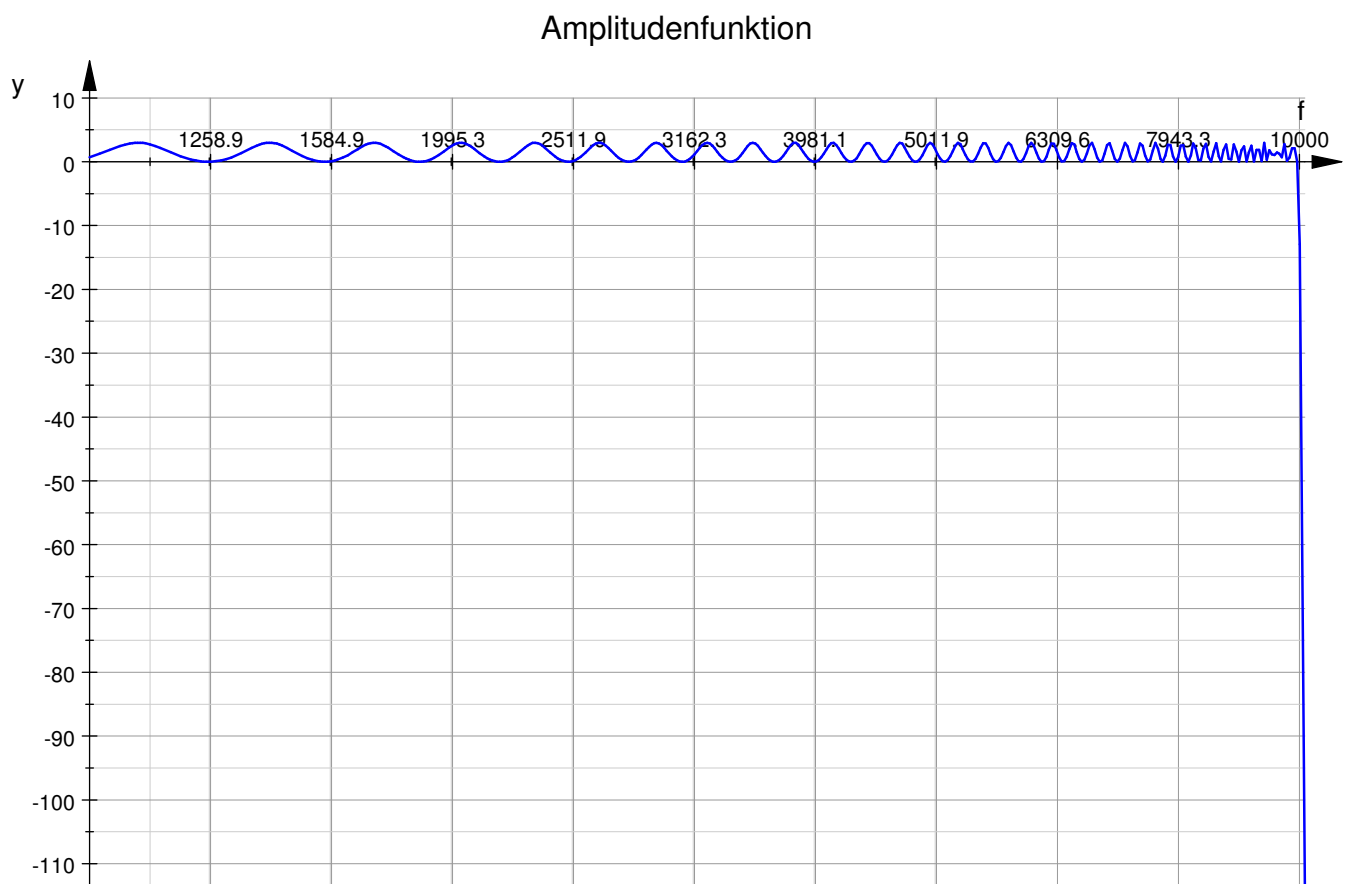
37 = 0.0087487448890242298158902198784893
1 = 54.385929023761373675261578490377
38 = 0.0079175714841979029358361754926847
2 = 7.6803263909842799105866489665018
39 = 0.0071335694598944080912231683287042
3 = 2.8244135026772738161762993866926
40 = 0.0063908485222774458278582536695637
4 = 1.4484508359257991794246852772063
41 = 0.0056841406990133720002322737108827
5 = 0.87727587376944114690134023650122
42 = 0.0050087016811942995113649694880518
6 = 0.58703521944691811706762310041865
43 = 0.004360227046115653497112444179457
7 = 0.41975895356964068930488113356459
44 = 0.0037347804929601859780498110389911
8 = 0.31467534187382647782981131843876
45 = 0.0031287317704877365982689609018095
9 = 0.24438502989486758595012563907229
46 = 0.0025387023992849047011313406718429
10 = 0.19506584678455965335531309716036
47 = 0.0019615176189340824480538372941411
11 = 0.15913355347314725761840935323881
48 = 0.0013941632433614439204482025468552
12 = 0.13214554234835645303212058619725
49 = 0.00083374630101863360870694831791529
13 = 0.11135956640234301203280597655079
50 = 0.00027745848181929385169003342138558
14 = 0.095008696263577788518935295751875
15 = 0.081912874628413110497319598522399
16 = 0.071259657318972658246621747791672
17 = 0.062474887168458557275756686721866
18 = 0.055143490081092847866820178102834
19 = 0.048959363455883960261921592270422
20 = 0.043692756414555801326693303990739
21 = 0.039168496923834309114895843909259
22 = 0.035251131406921218273261273120966
23 = 0.031834577962932664028736043110205
24 = 0.028834791756670469269779890506313
25 = 0.026184480486353281873894785898784
26 = 0.023829240206054528291242859006449
27 = 0.021724691301814359474273062702939
28 = 0.019834329254612473815541903407714
29 = 0.018127893244020147051326602537436
30 = 0.01658011464078419239071532416772
31 = 0.015169747428723004414933378548054
32 = 0.013878810107842216271010307926197
33 = 0.01269198781550849913038518668436
34 = 0.011596156950942250091905781901387
35 = 0.010580004269263372546492744894827
36 = 0.0096337194049837173237798167714102

37 = 1.2037548778107300598043580122628
1 = 3079.7639647025228277103592314162
38 = 1.1714658504725376234554273436693
2 = 435.35081924408751969751648056602
39 = 1.1426827822546278818197694557107
3 = 160.41553801448381255256117069243
40 = 1.1171243599500013892994992881794
4 = 82.510977897898767318101695012367
41 = 1.0945490425728239088317937217945
5 = 50.172750921521328365202552133881
42 = 1.0747499396137219028320562679665
6 = 33.740928343489245231954172024226
43 = 1.0575505890661601055606816028563
7 = 24.271451398528303764719836330446
44 = 1.0428014810041340367628651863585
8 = 18.323502735521817559910692512694
45 = 1.0303772048002496555418227006397
9 = 14.345768359044721912652765456195
46 = 1.0201741236316484887318124156362
10 = 11.555658146206839945247084917488
47 = 1.0121085003625443464400576243096
11 = 9.5237665123898603131688219090265
48 = 1.0061150154540097945368302763962
12 = 7.9985637410591067659965060601979
49 = 1.0021456311839846253159283467376
13 = 6.8247930356347495264544526636671
50 = 1.0001687679040449507586554585411
14 = 5.9024195103531390474144040868869
15 = 5.1646369122864727596341284463149
16 = 4.5654534840432948712384215922862
17 = 4.0723697154440901887855574335651
18 = 3.6618939545965850763202149971852
19 = 3.3167051968516041103138408534489
20 = 3.0238062426869563850829752363565
21 = 2.7732909983723342086435916211527
22 = 2.5575031587366820497882425511159
23 = 2.3704504508827303460100628549302
24 = 2.2073894299042294775335555640138
25 = 2.0645263547966710608469126261744
26 = 1.9387984911645997472409399206744
27 = 1.8277120503128991856628992480436
28 = 1.7292206085646669757306670790818
29 = 1.6416328570387297530129718190682
30 = 1.5635418724529382105792476891734
31 = 1.4937703639856771077986654602858
32 = 1.4313279090817944558325601871856
33 = 1.3753772775288280905973727393879
34 = 1.3252077104690067982263010676728
35 = 1.2802135694194526258764239986238
36 = 1.2398771666715302306870153354915

- `delete i:prod:=(f)->product(1+a[i]*I*w/wg+b[i]*(I*w/wg)^2, i=1..k):`
- `f:=0:b0:=float(expand(prod(f))):a0:=b0:`
- `U2U0:=(f)->a0/(1+1/ue2)/prod(f):`
- `U2U0dB:=(f)->20*log(10,abs(U2U0(f))):`
- `Winkel:=(f)->180/PI*arg(U2U0(f)):`
- `tg1:=(f)->-diff(Winkel(f),f)/360:`
- `tg:=(f)->1/2/PI/fg*sum(a[i]*(1+b[i]*(w/wg)^2)/(1+(a[i]^2-2*b[i])*(w/wg)^2+b[i]^2*(w/wg)^4),i=1..k):`

Betrag der Übertragungsfunktion doppelt Logarithmisch

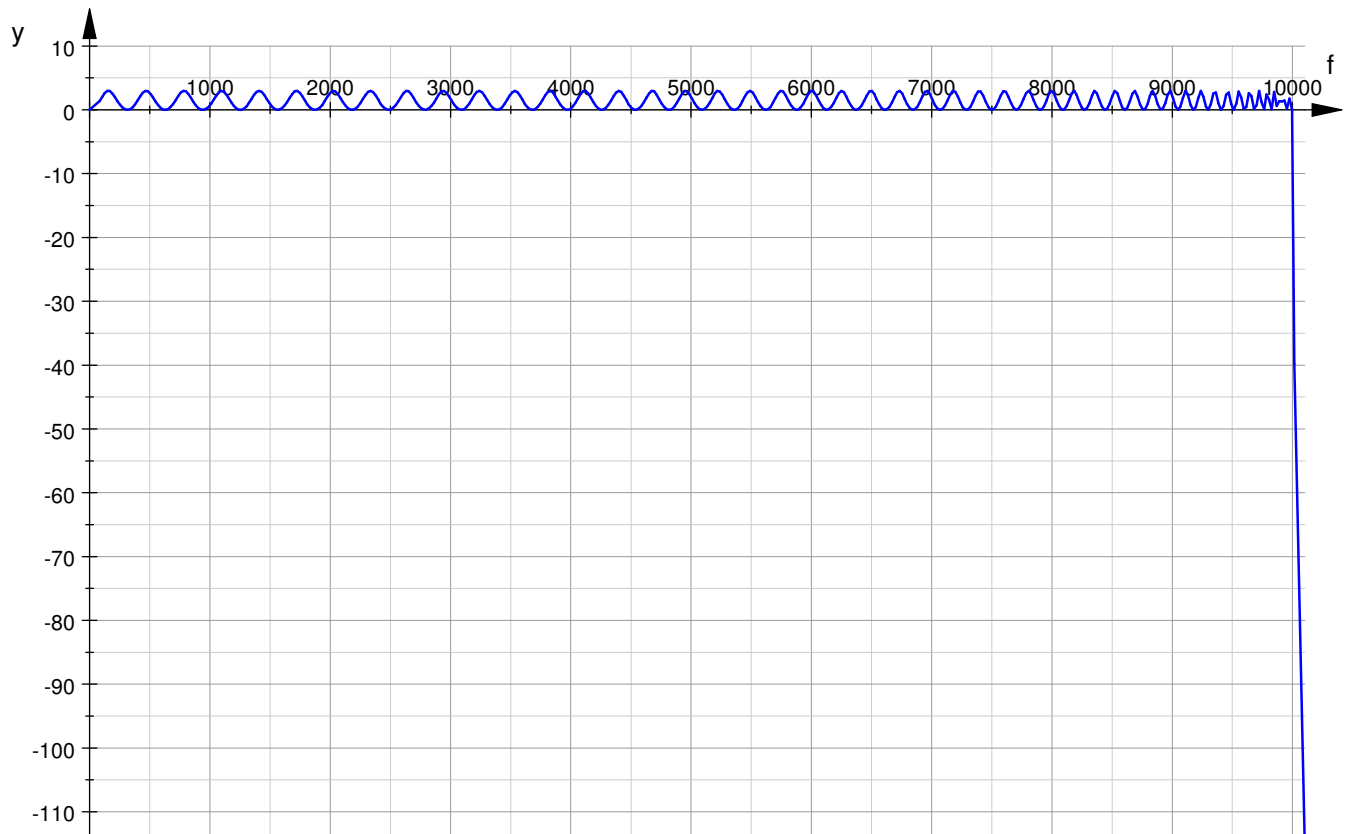
- `delete f:plotfunc2d(U2U0dB(f)+6.02, f=1000..1.01*fg, LegendVisible=FALSE, CoordinateType=LogLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Amplitudenfunktion", YMax=10):`



Ausschnittsvergrößerung aus dem Betrag der Übertragungsfunktion, einfach logarithmisch

- `plotfunc2d(U2U0dB(f)+6.02, f=0..1.01*fg, LegendVisible=FALSE, CoordinateType=LinLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Vergrößerung Amplitudenfunktion", YMax=10):`

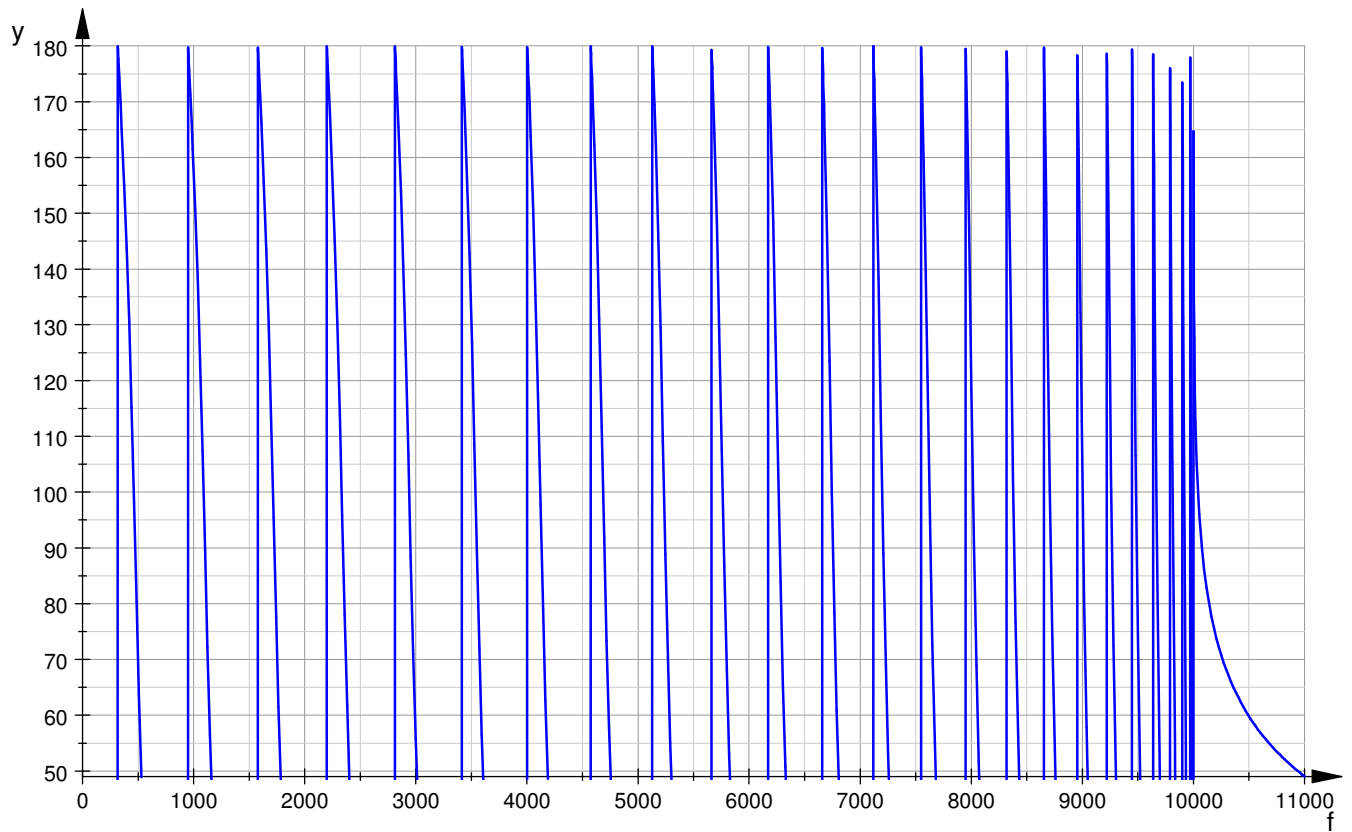
Vergößerung Amplitudenfunktion



die Phasenverschiebung des Filters

- ```
plotfunc2d(Winkel(f), f=0..1.1*fg, LegendVisible=FALSE,
 GridVisible=TRUE, SubgridVisible=TRUE,
 Height=120*unit::mm, Width=180*unit::mm, Mesh=500, AdaptiveMesh=4,
 Header="Phasenfunktion"):
```

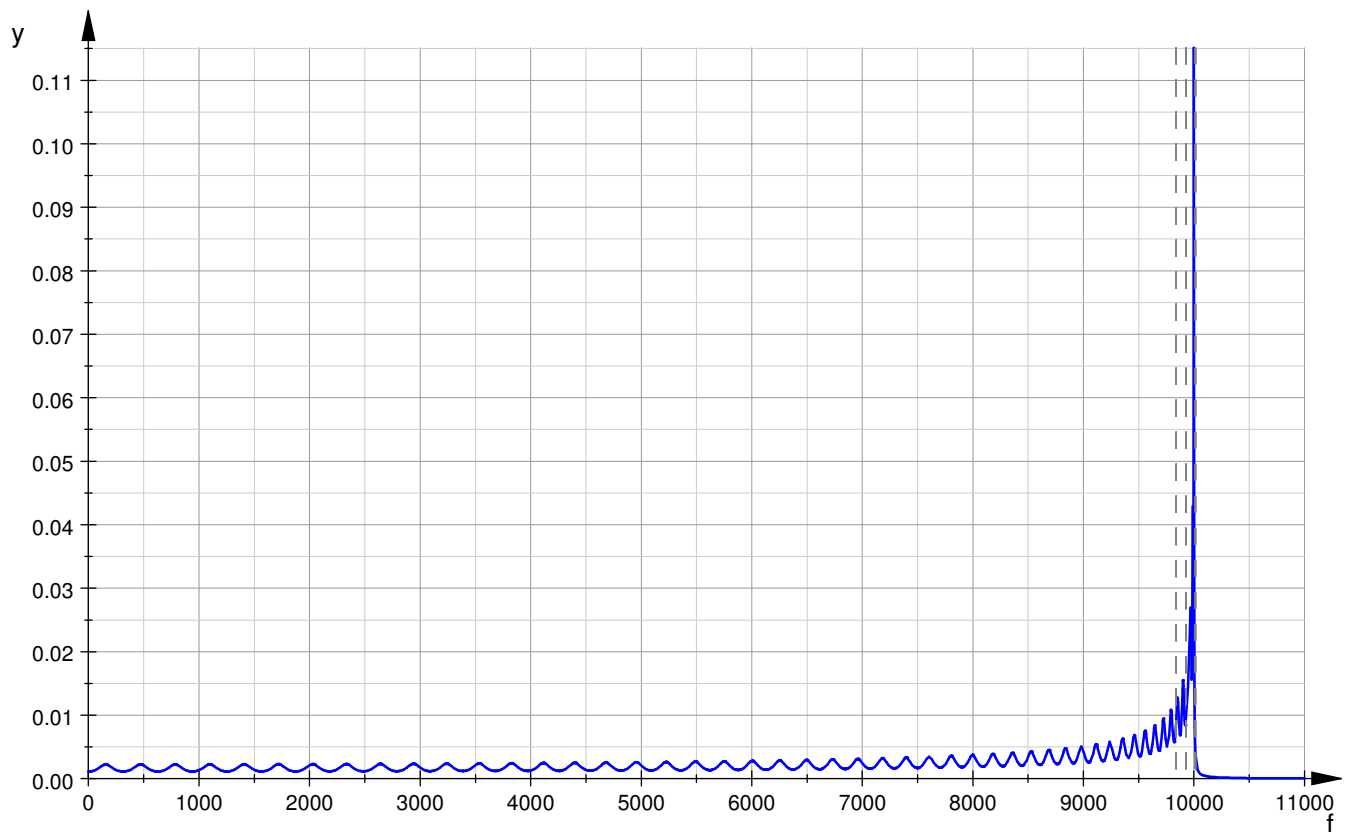
## Phasenfunktion



### Berechnete Gruppenlaufzeit

- `plotfunc2d(tg(f), f=0..1.1*fg, LegendVisible=FALSE, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Gruppenlaufzeit", Mesh=500, AdaptiveMesh=4):`

## Gruppenlaufzeit



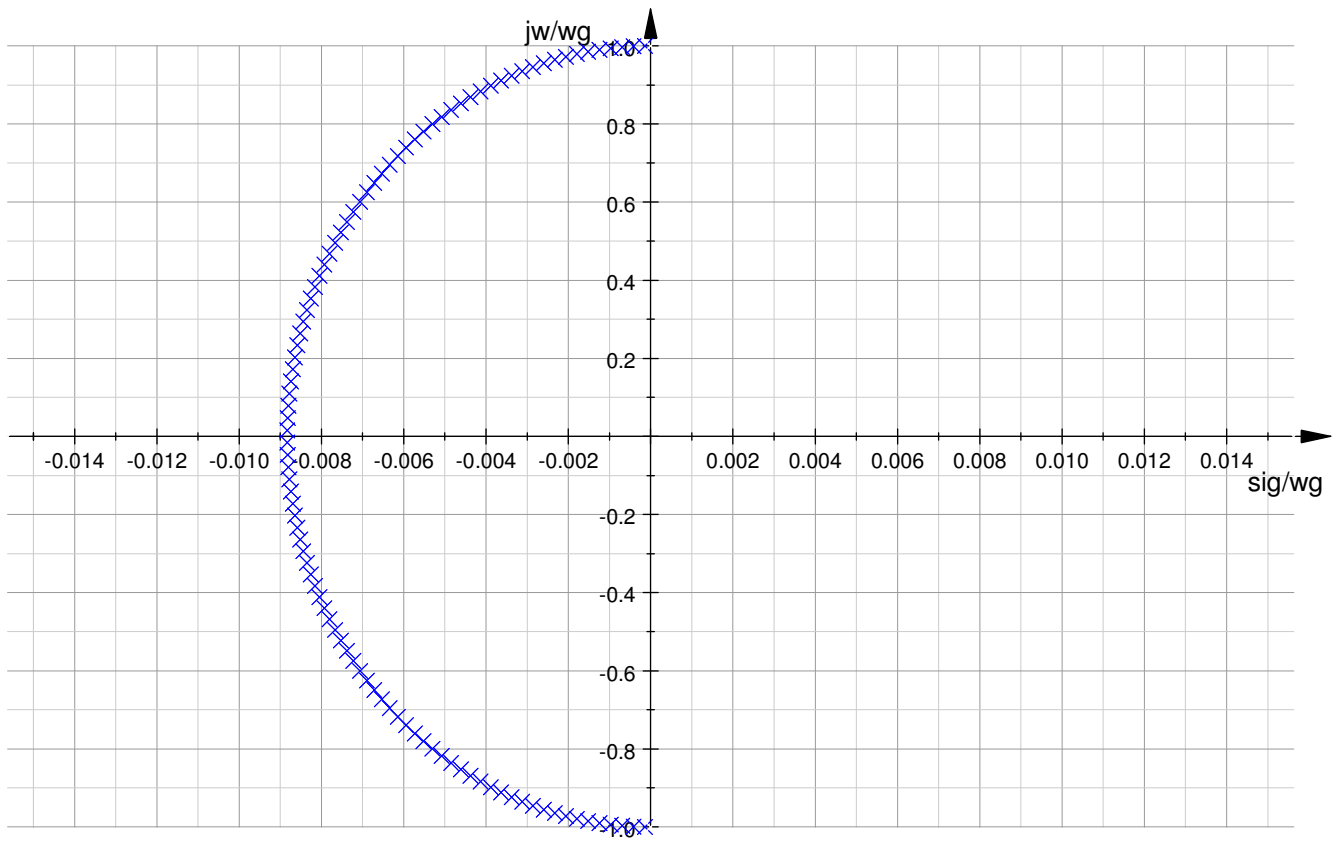
Lage der Pol-Nullstellen des Filters in der komplexen Ebene durch Suche der Nullstellen jedes Linearfaktors einzeln

- `Pol:=solve(prod(f)=0,f)/fg:`
- `delete PolTab:for i from 1 to n do`  
    `PolTab[i]:=-Im(op(Pol,i))-I*Re(op(Pol,i)):`  
  `end_for:`

(bei den Nullstellen wurden Im und Re vertauscht und mit -1 multipliziert, sonst liegen die komischerweise in der sig-Ebene mit der Hauptachse der Ellipse !)

- `Liste:=[[Re(op(op(PolTab,i),2)),Im(op(op(PolTab,i),2)),RGB::Blue] $`  
  `i=1..n]:`
- `Breite:=1/64>Liste:=Liste.[[Breite,0,RGB::White]].[[0,1,RGB::White]]`  
  `.[[-Breite,0,RGB::White]].[[0,-1,RGB::White]]:`
- `plot(plot::PointList2d(Liste, PointStyle=XCrosses, PointSize=2,`  
  `Color=RGB::Blue, GridVisible=TRUE, SubgridVisible=TRUE,`  
    `Scaling=Unconstrained,`  
  `AxesTitles=["sig/wg","jw/wg"], Height=120*unit::mm,`  
  `Width=180*unit::mm, Header="Pol-Null-Stellen"):`

# Pol-Null-Stellen



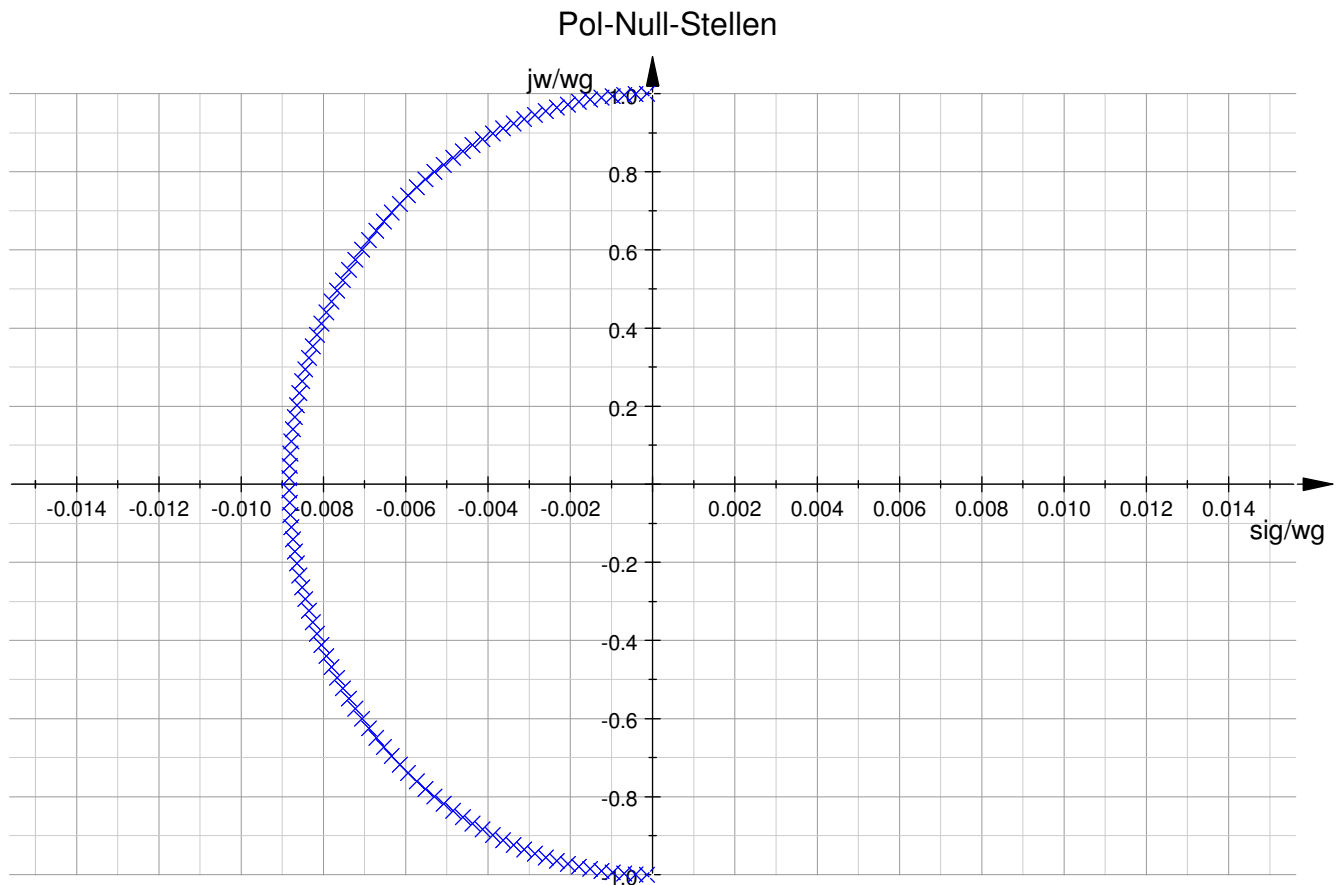
- PolTab;



37 = -0.0070617358486402257916226004297806 - 0.60044363537144597795007942494087 · i  
74 = -0.0068917082832331780651493564570422 + 0.62526703419383634346485952950988 · i  
1 = -0.0065314238090560905265968505002163 - 0.67303875388766277487474685007005 · i  
38 = -0.0070617358486402257916226004297806 + 0.60044363537144597795007942494087 · i  
75 = -0.0038849491159054040405472199268236 - 0.89806258934872887871048115489918 · i  
2 = -0.0065314238090560905265968505002163 + 0.67303875388766277487474685007005 · i  
39 = -0.0082605794381566547273719623416424 - 0.35348862556382871638415196535541 · i  
76 = -0.0038849491159054040405472199268236 + 0.89806258934872887871048115489918 · i  
3 = -0.0086471745483480381762853631487384 - 0.20279520191865585481173698015928 · i  
40 = -0.0082605794381566547273719623416424 + 0.35348862556382871638415196535541 · i  
77 = -0.00096902536547783815294103135802599 - 0.99399970943159904325594425506485 · i  
4 = -0.0086471745483480381762853631487384 + 0.20279520191865585481173698015928 · i  
41 = -0.0086991563105612395505049609609676 - 0.17193580369795074874850502788669 · i  
78 = -0.00096902536547783815294103135802599 + 0.99399970943159904325594425506485 · i  
5 = -0.0031214128587022021839786950531898 - 0.93548050326443675555650397167463 · i  
42 = -0.0086991563105612395505049609609676 + 0.17193580369795074874850502788669 · i  
79 = -0.0076705814469064846866504723242481 - 0.49547798608613207675204984594262 · i  
6 = -0.0031214128587022021839786950531898 + 0.93548050326443675555650397167463 · i  
43 = -0.0043752224120542703384568774611744 - 0.86866538189047183654768849101649 · i  
80 = -0.0076705814469064846866504723242481 + 0.49547798608613207675204984594262 · i  
7 = -0.0057350488296215379301568305792472 - 0.7604356133992712068977868916442 · i  
44 = -0.0043752224120542703384568774611744 + 0.86866538189047183654768849101649 · i  
81 = -0.0020614744538915285697752186680137 - 0.97240783255183030830776307398107 · i  
8 = -0.0057350488296215379301568305792472 + 0.7604356133992712068977868916442 · i  
45 = -0.0072247943333383667461681771265209 - 0.57502767117338261113968798471341 · i  
82 = -0.0020614744538915285697752186680137 + 0.97240783255183030830776307398107 · i  
9 = -0.008820847522832112007156352427331 - 0.047108287363557466860656921170617 · i  
46 = -0.0072247943333383667461681771265209 + 0.57502767117338261113968798471341 · i  
83 = -0.0025965673889094775881735645363 - 0.95583028062829456075059484095344 · i  
10 = -0.008820847522832112007156352427331 + 0.047108287363557466860656921170617 · i  
47 = -0.0017907437613935360228764703473012 - 0.97926098996623988292931270480788 · i  
84 = -0.0025965673889094775881735645363 + 0.95583028062829456075059484095344 · i  
11 = -0.0085866590688416523125984999687602 - 0.2334544657588286876849313084357 · i  
48 = -0.0017907437613935360228764703473012 + 0.97926098996623988292931270480788 · i  
85 = -0.0059431383893582014600300060792457 - 0.73965993277737483784832388156411 · i  
12 = -0.0085866590688416523125984999687602 + 0.2334544657588286876849313084357 · i  
49 = -0.0063415224575643917952597123793264 - 0.69593992983931064818967168947213 · i  
86 = -0.0059431383893582014600300060792457 + 0.73965993277737483784832388156411 · i  
13 = -0.0079301677949078597678757506315331 - 0.43995632283573892001013833797092 · i  
50 = -0.0063415224575643917952597123793264 + 0.69593992983931064818967168947213 · i  
87 = -0.0075293674209044328037815367200271 - 0.52251893663996159050929734863188 · i  
14 = -0.0079301677949078597678757506315331 + 0.43995632283573892001013833797092 · i  
51 = -0.0088295612337639906589429599371226 - 0.015707929731253625080813003650945 · i  
88 = -0.0075293674209044328037815367200271 + 0.52251893663996159050929734863188 · i  
15 = -0.00552129946909087417279609144667 - 0.78046083587924670495316832979818 · i  
52 = -0.0088295612337639906589429599371226 + 0.015707929731253625080813003650945 · i  
89 = -0.0080482839365219452076532658886931 - 0.41153040331759177632833113735646 · i  
16 = -0.00552129946909087417279609144667 + 0.78046083587924670495316832979818 · i  
53 = -0.0036339395379795174769568593914479 - 0.91143881173458309968942899588508 · i  
90 = -0.0080482839365219452076532658886931 + 0.41153040331759177632833113735646 · i  
17 = -0.0061453627890283616812674131445727 - 0.71815429710205309085027635371742 · i  
54 = -0.0036339395379795174769568593914479 + 0.91143881173458309968942899588508 · i  
91 = -0.00815845739357177870071989276172 - 0.38269835297603787799123124562226 · i  
18 = -0.0061453627890283616812674131445727 + 0.71815429710205309085027635371742 · i  
55 = -0.0028604015593051242906742530828789 - 0.9461222461614848190210164043771 · i  
92 = -0.00815845739357177870071989276172 + 0.38269835297603787799123124562226 · i  
19 = -0.0033793437004604824177020797102523 - 0.9239155540525965667520525395529 · i  
56 = -0.0028604015593051242906742530828789 + 0.9461222461614848190210164043771 · i  
93 = -0.0041321247180894828570551267216505 - 0.88380008761159784234887680104228 · i

### direkte Berechnung der Pole nach HERPY/BERKA

- `sig/wg:=[-sin(PI/2/n*(2*i-1))*sinh(1/n*arcsinh(1/epsilon)) $  
i=1..n]:wiwg:=[I*cos(PI/2/n*(2*i-1))*cosh(1/n*arcsinh(1/epsilon)) $  
i=1..n]:`
- `delete Liste:for i from 1 to n do  
    PolTabdirekt[i]:=float(op(sig/wg,i))+float(op(wiwg,i)):  
end_for:`
- `Liste:=[[Re(op(op(PolTabdirekt,i),2)),Im(op(op(PolTabdirekt,i),2)),R  
GB::Blue] $ i=1..n]:`
- `Breite:=1/64>Liste:=Liste.[[Breite,0,RGB::White]].[[0,1,RGB::White]]  
.[[-Breite,0,RGB::White]].[[0,-1,RGB::White]]:`
- `plot(plot::PointList2d(Liste, PointStyle=XCrosses, PointSize=2,  
Color=RGB::Blue, GridVisible=TRUE, SubgridVisible=TRUE,  
    Scaling=Unconstrained,  
AxesTitles=["sig/wg", "jw/wg"], Height=120*unit::mm,  
Width=180*unit::mm, Header="Pol-Null-Stellen")):`



- `PolTabdirekt;`

37 = -0.0080482839365219452076532658884153 + 0.41153040331759177632833113735647 · i  
74 = -0.0065314238090560905265968504999909 - 0.67303875388766277487474685007005 · i  
1 = -0.00013870583181713233697684439043038 + 0.99991561710698696656809888828276 · i  
38 = -0.0081584573935717787007198927614385 + 0.38269835297603787799123124562227 · i  
75 = -0.0063415224575643917952597123791075 - 0.69593992983931064818967168947213 · i  
2 = -0.00041598060954155151904092900701512 + 0.99892882111417781429822207740412 · i  
39 = -0.0082605794381566547273719623413573 + 0.35348862556382871638415196535542 · i  
76 = -0.0061453627890283616812674131443606 - 0.71815429710205309085027635371742 · i  
3 = -0.00069284486462630080676135864744134 + 0.99695620297706708838487628411713 · i  
40 = -0.0083545492881479118738965335888183 + 0.32393004755538701431243405768366 · i  
77 = -0.0059431383893582014600300060790406 - 0.73965993277737483784832388156411 · i  
4 = -0.00096902536547783815294103135799255 + 0.99399970943159904325594425506485 · i  
41 = -0.008440274206648726247537691995764 + 0.29405178969855673320751053358935 · i  
78 = -0.0057350488296215379301568305790492 - 0.7604356133992712068977868916442 · i  
5 = -0.0012442495552854990415830760934448 + 0.99006225817996126444050978890208 · i  
42 = -0.0085176695935142323804404989129067 + 0.26388333822659677820430489441878 · i  
79 = -0.0055212994690908741727960914464794 - 0.78046083587924670495316832979818 · i  
6 = -0.0015182458210021624312349287572988 + 0.98514773501116486787597865403066 · i  
43 = -0.0085866590688416523125984999684639 + 0.23345446575882868768493130843571 · i  
80 = -0.0053021012525787809165270895113106 - 0.79971583774038800538872064996124 · i  
7 = -0.0017907437613935360228764703472394 + 0.97926098996623988292931270480788 · i  
44 = -0.00864717454834803817628536314844 + 0.20279520191865585481173698015929 · i  
81 = -0.0050776705022608342059090867139344 - 0.81818161662054270869932282959453 · i  
8 = -0.0020614744538915285697752186679426 + 0.97240783255183030830776307398107 · i  
45 = -0.0086991563105612395505049609606673 + 0.17193580369795074874850502788671 · i  
82 = -0.0048482287041917450006866258679149 - 0.83583994902535660393351060738014 · i  
9 = -0.0023301707199883571684368224111703 + 0.96459502600691244466924659170037 · i  
46 = -0.0087425530557577862127448511993719 + 0.14090672559705690743158148020008 · i  
83 = -0.0046140022897253635115168647245825 - 0.8526734083126624471632519465309 · i  
10 = -0.0025965673889094775881735645362104 + 0.95583028062829456075059484095344 · i  
47 = -0.0087773219565895217134139930854124 + 0.10973858956987391488281329520463 · i  
84 = -0.0043752224120542703384568774610234 - 0.8686653818904718365476884910165 · i  
11 = -0.0028604015593051242906742530827802 + 0.9461222461614848190210164043771 · i  
48 = -0.0088034287003490253978542727346544 + 0.078462154803685935617825076684156 · i  
85 = -0.0041321247180894828570551267215078 - 0.88380008761159784234887680104228 · i  
12 = -0.0031214128587022021839786950530821 + 0.93548050326443675555650397167463 · i  
49 = -0.0088208475228321120071563524270265 + 0.047108287363557466860656921170674 · i  
86 = -0.0038849491159054040405472199266895 - 0.89806258934872887871048115489918 · i  
13 = -0.0033793437004604824177020797101356 + 0.9239155540525965667520525395529 · i  
50 = -0.0088295612337639906589429599368179 + 0.015707929731253625080813003651116 · i  
87 = -0.0036339395379795174769568593913224 - 0.91143881173458309968942899588508 · i  
14 = -0.0036339395379795174769568593913224 + 0.91143881173458309968942899588508 · i  
51 = -0.0088295612337639906589429599368179 - 0.015707929731253625080813003651116 · i  
88 = -0.0033793437004604824177020797101356 - 0.9239155540525965667520525395529 · i  
15 = -0.0038849491159054040405472199266895 + 0.89806258934872887871048115489918 · i  
52 = -0.0088208475228321120071563524270265 - 0.047108287363557466860656921170674 · i  
89 = -0.0031214128587022021839786950530821 - 0.93548050326443675555650397167463 · i  
16 = -0.0041321247180894828570551267215078 + 0.88380008761159784234887680104228 · i  
53 = -0.0088034287003490253978542727346544 - 0.078462154803685935617825076684156 · i  
90 = -0.0028604015593051242906742530827802 - 0.9461222461614848190210164043771 · i  
17 = -0.0043752224120542703384568774610234 + 0.8686653818904718365476884910165 · i  
54 = -0.0087773219565895217134139930854124 - 0.10973858956987391488281329520463 · i  
91 = -0.0025965673889094775881735645362104 - 0.95583028062829456075059484095344 · i  
18 = -0.0046140022897253635115168647245825 + 0.8526734083126624471632519465309 · i  
55 = -0.0087425530557577862127448511993719 - 0.14090672559705690743158148020008 · i  
92 = -0.0023301707199883571684368224111703 - 0.96459502600691244466924659170037 · i  
19 = -0.0048482287041917450006866258679149 + 0.83583994902535660393351060738014 · i  
56 = -0.0086991563105612395505049609606673 - 0.17193580369795074874850502788671 · i  
93 = -0.0020614744538915285697752186679426 - 0.97240783255183030830776307398107 · i

die Grunddämpfung durch  $\ddot{u}^2$

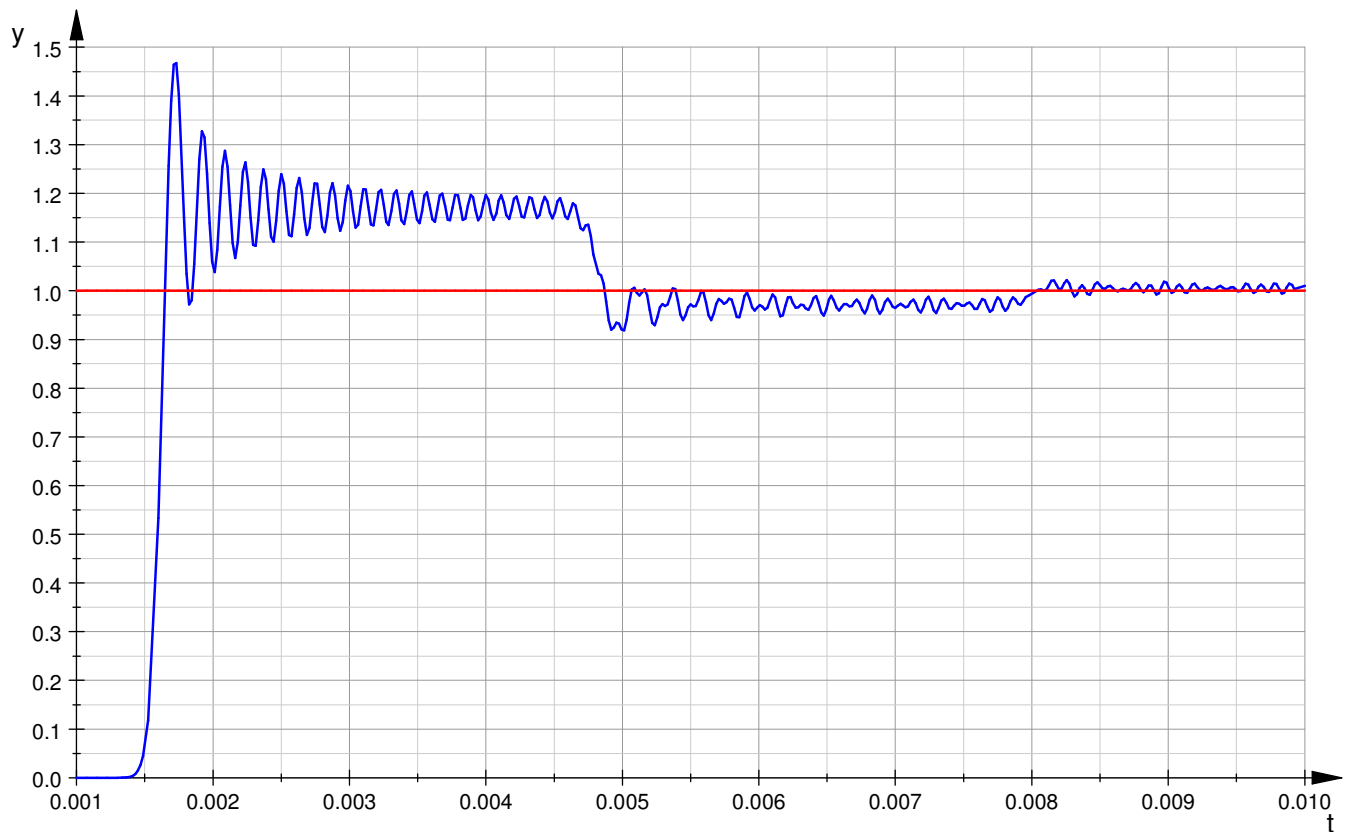
- $1/(1+1/\ddot{u}^2)$ ;

$$\frac{1}{2}$$

Sprungantwort des Filters  $u_a(t)=\text{invlaplace}(2/p*T(p))$

- `delete i:prodp:=(p)->product(1+a[i]*p/wg+b[i]*(p/wg)^2, i=1..k):`
- `ua:=(t)->Re(transform::invlaplace(a0/(1+ue2)*2/p/prodp(p),p,t)):`
- `plotfunc2d(ua(t), 1, t=1/1000..100/fg, LegendVisible=FALSE, CoordinateType=LinLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Sprungantwort", YMax=1.5):`

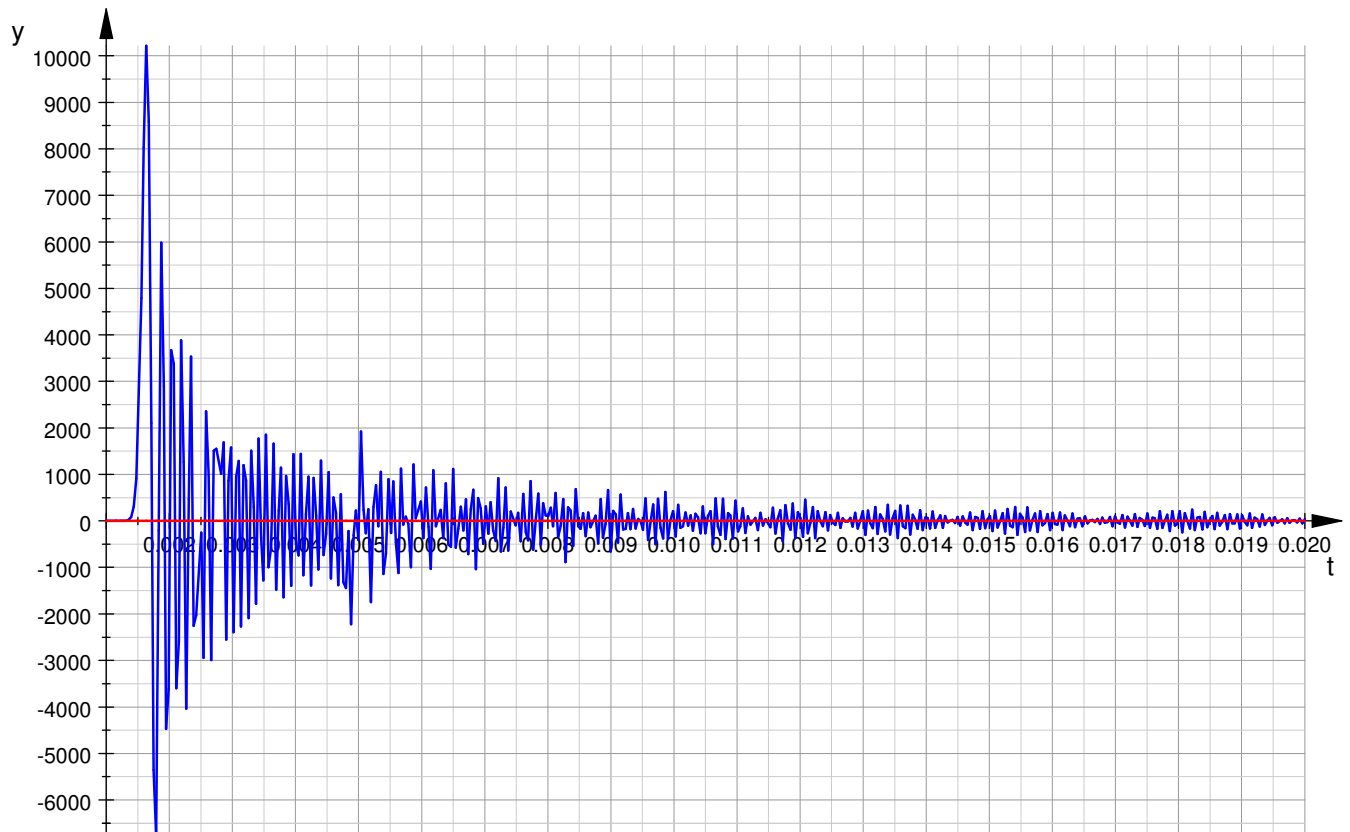
Sprungantwort



Impulsantwort des Filters  $u_a(t)=\text{invlaplace}(T(p))$

- `ua1:=(t)->Re(transform::invlaplace(a0/(1+ue2)*2/prodp(p),p,t)):`
- `plotfunc2d(ua1(t), 1, t=1/1000..200/fg, LegendVisible=FALSE, CoordinateType=LinLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Impulsantwort"):`

## Impulsantwort



### Suchbereich definieren

- `anf:=1/1000:ende:=0.00175:`

### Überschwingen in % bei t in us

- `maximum:=op(numeric::solve(diff(ua(t),t)=0,t=anf..ende,RestrictedSearch),1):`

- `(ua(maximum)-1)*100;maximum/1e-6;`

47.455324143379610873147456385432

1722.7189326288190957022869175304

### Ausschnittsvergrößerung der Sprungantwort

to für  $ua(t)=1/2$  in us

- `tx:=op(numeric::solve(Re(ua(t))=1/2,t=anf..maximum,RestrictedSearch),1):tx/1e-6;`

1596.0153246828824413181926115405

### die Einschwingzeit tau in us und die daraus resultierende Grenzfrequenz in kHz

- `m:=ua'(t):t:=tx:m:=float(m):delete t:yt:=t->1/2-m*(tx-t):`
- `tau:=op(solve(yt(t)=1,t),1)-op(solve(yt(t)=0,t),1):tau/1e-6;1/2/tau/1e3;`

121.85777385157075530238490136953

4.1031440522541185805805919244038

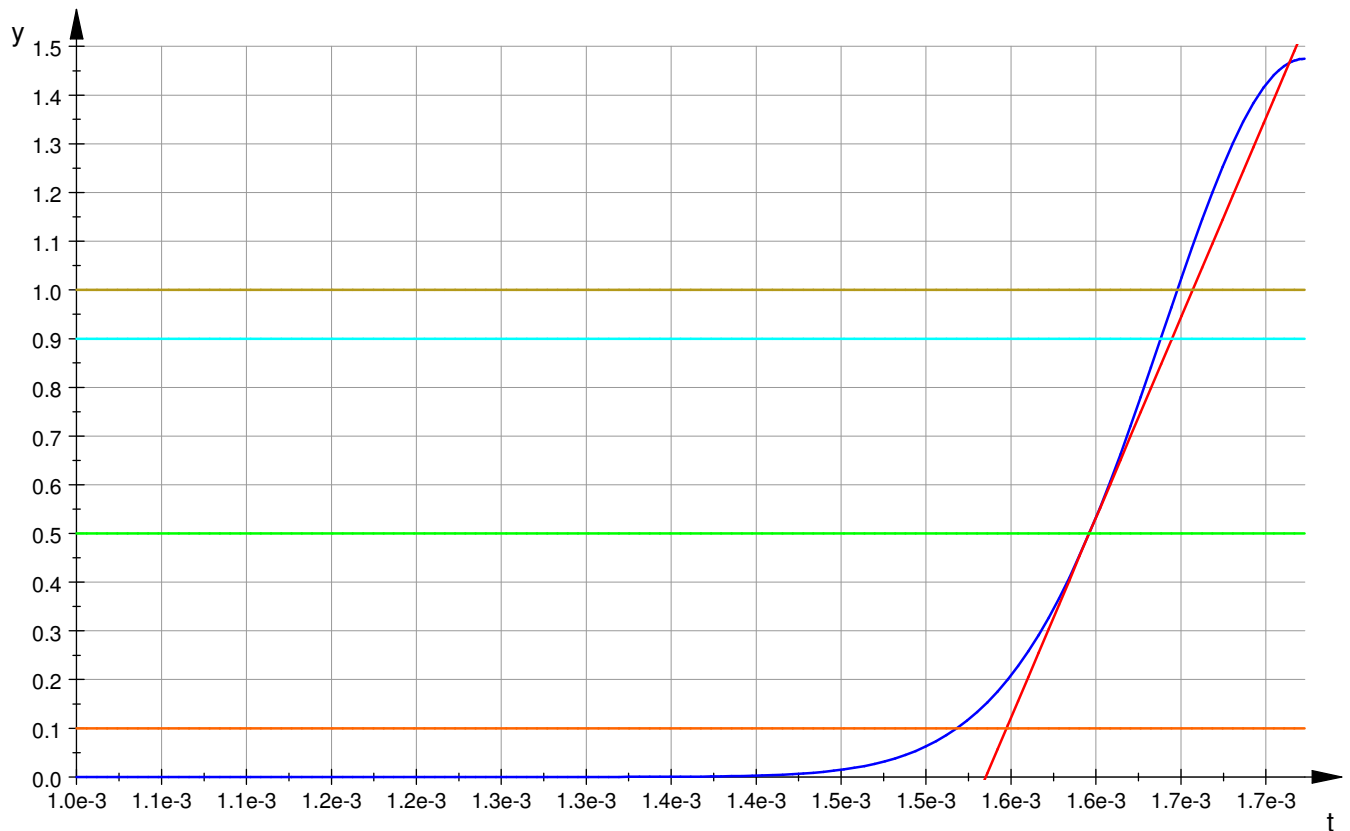
tr, Rise-Time in us

- `tr:=op(numeric::solve(ua(t)=9/10,t=anf..ende,RestrictedSearch),1)-op(numeric::solve(ua(t)=1/10,t=anf..ende,RestrictedSearch),1):tr/1e-6;`

120.03701303237661995980213177027

- `plotfunc2d(ua(t), yt(t), 1/2, 1, 1/10, 9/10, t=0.001..maximum, LegendVisible=FALSE, CoordinateType=LinLin, GridVisible=TRUE, SubgridVisible=FALSE, Height=120*unit::mm, Width=180*unit::mm, Header="Vergrößerung Sprungantwort", YRange=0..1.5):`

Vergrößerung Sprungantwort



CPU-Zeit in Sekunden und in Minuten

- `te:=time():float((te-ta)/1000);float((te-ta)/1000/60);`

98.593

