APPENDIX A

EXAMPLE *MATHEMATICA* NOTEBOOKS

Filter Design

For Signal Processing

Using MATLAB and *Mathematica*

Miroslav D. Lutovac, Ph. D.

Dejan V. Tošić, Ph. D.

Brian L. Evans, Ph. D.

A.1 Analysis by Transform Method of Analog LTI Circuits

Miroslav D. Lutovac, Dejan V. Tosic and Brian L. Evans
lutovac@iritel.bg.ac.yu tosic@galeb.etf.bg.ac.yu bevans@ece.utexas.edu

A.1.1 References

1. G. S. Moschytz and P. Horn,
Active Filter Design Handbook
For Use with Programmable Pocket Calculators and Mini Computers.
John Wiley and Sons, Ltd., New York, 1981, pp. 38–39

2. R. P. Sallen and E. L. Key,
"A practical method of designing RC active filters,"
IRE Trans. Circuit Theory, CT-2, Mar. 1955, pp. 74–85.

3. D.V. Tosic, M.D. Lutovac, B.L. Evans, I.M. Markoski,
"A tool for symbolic analysis and design of analog active filters,"
5th Int. Workshop, Symbolic Methods, Applications, Circuit Design,
SMACD'98, Kaiserslautern, Germany, pp. 71–74, Oct. 1998.

4. D. V. Tosic, M. D. Lutovac,
"Symbolic computation of impulse, step and sine responses of linear time-invariant systems,"
Proc. 10th Int. Symp. Theor. Electrical Engineering,
ISTET99, Magdeburg, Germany, Sep. 1999, pp. 653–657.

5. D. V. Tosic,
"SALECAS - a package for symbolic analysis of linear circuits and systems,"
4th Int. Workshop, Symbolic Methods, Applications, Circuit Design,
SMACD, Leuven, Belgium, pp. 227–230, Oct. 1996.

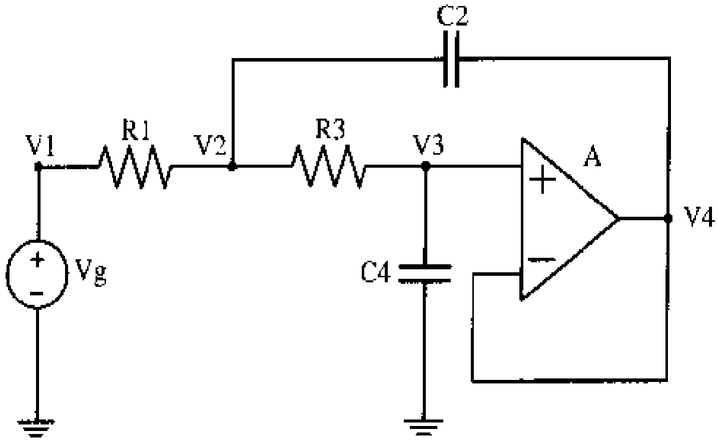
6. M. D. Lutovac, D. V. Tosic and B. L. Evans,
"Advanced Filter Design for Signal Processing using MATLAB and Mathematica,"
http://iritel.iritel.bg.ac.yu/~lutovac/www/afdhome.htm
http://galeb.etf.bg.ac.yu/~tosic/afdhome.htm
http://www.ece.utexas.edu/~bevans/

A.1.2 Initialization

```
SetDirectory[HomeDirectory[]];
<<Calculus'LaplaceTransform'
<<afd\math\m\clearall.m
<<afd\math\m\drawafil.m
<<afd\math\m\drawacc.m
```

A.1.3 Schematic

```
DrawLPSK[0, 0, 1/2, 1.25, 8];
```



A.1.4 Circuit Analysis

Reduced Modified Nodal Analysis (RMNA)

```
CircuitEquations = {
 V1 == Vg
,(V2-V1)/R1 + (V2-V3)/R3 + (V2-V4)*(s*C2) == 0
,(V3-V2)/R3 + V3*(s*C4) == 0
,V3 == V4
};
NodeVoltages = {V1,V2,V3,V4};
CircuitResponse = Together[Flatten[
 Solve[CircuitEquations,NodeVoltages]
]];
```

A.1.5 Response

```
V1s = Simplify[V1 /. CircuitResponse];
Collect[Numerator[%],s]/Collect[Denominator[%],s]
```

```
Vg
```

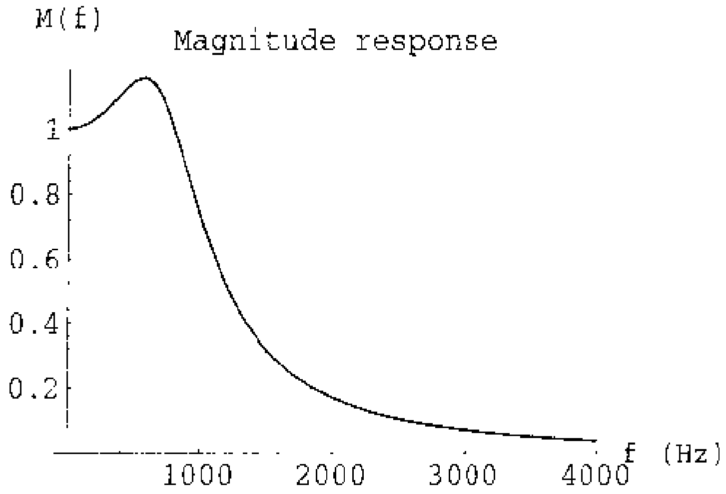
```
V2s = Simplify[V2 /. CircuitResponse];
Collect[Numerator[%],Vg]/Collect[Denominator[%],s]
```

```
            (1 + C4 R3 s) Vg
----------------------------------------
                                       2
1 + (C4 R1 + C4 R3) s + C2 C4 R1 R3 s
```

```
V3s = Simplify[V3 /. CircuitResponse];
Collect[Numerator[%],s]/Collect[Denominator[%],s]

                   Vg
-------------------------------------
                                    2
1 + (C4 R1 + C4 R3) s + C2 C4 R1 R3 s

V4s = Simplify[V4 /. CircuitResponse];
Collect[Numerator[%],s]/Collect[Denominator[%],s]

                   Vg
-------------------------------------
                                    2
1 + (C4 R1 + C4 R3) s + C2 C4 R1 R3 s
```

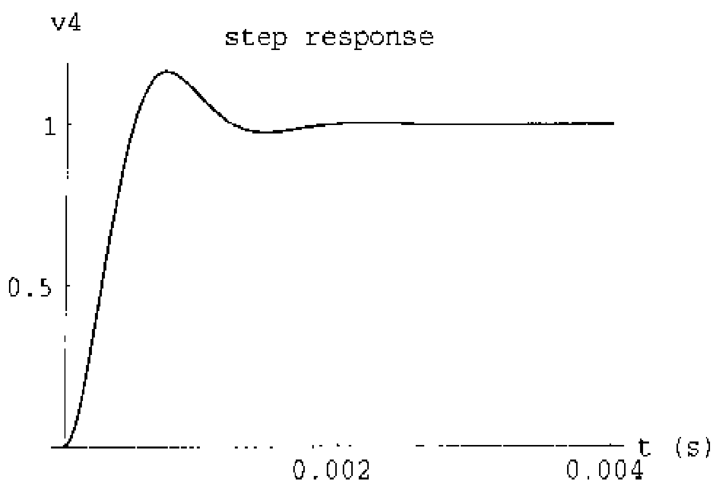
A.1.6 Voltage Transfer Function

```
H = V4s/Vg;
Collect[Numerator[%],s]/Collect[Denominator[%],s]

                   1
-------------------------------------
                                    2
1 + (C4 R1 + C4 R3) s + C2 C4 R1 R3 s

M = Abs[H] /. {C2->4*C, C4->C, R1->R, R3->R} ./ s -> I*2*Pi*f

                   1
Abs[---------------------------------]
                         2 2   2 2
     1 + 4 * C f Pi R - 16 C F Pi R

Plot[ {M ./ {C->10^(-8), R->10^4}}, {f,0,4000}
, PlotRange -> All
, AxesLabel -> {"f (Hz)", "M(f)"}
, AxesOrigin -> {0,0}
, PlotLabel -> "Magnitude response"];
```



A.1.7 Step Response

```
v4t = InverseLaplaceTransform[
H/s //. {C2->4*C, C4->C, R1->R, R3->R}
, s, t];
v4t //Together //Simplify

          Sqrt[3] t            Sqrt[3] t
      Cos[---------]       Sin[---------]
            4 C R                4 C R
1 - -------------- - ------------------
        t/(4 C R)              t/(4 C R)
       E                Sqrt[3] E

Plot[v4t /. {C->10^(-8), R->10^4}
, {t,0,4*10^(-3)}
, PlotRange -> All
, AxesLabel -> {"t (s)", "v4"}
, Ticks -> {{0,0.002,0.004},{0,0.5,1}}
, PlotLabel -> "step response"];
```



A.2 Analysis by Transform Method of Discrete-Time LTI Systems

Miroslav D. Lutovac, Dejan V. Tosic and Brian L. Evans
lutovac@iritel.bg.ac.yu tosic@galeb.etf.bg.ac.yu bevans@ece.utexas.edu

A.2.1 References

1. Alan Oppenheim, Ronald Schafer, "Digital Signal Processing," Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

2.Sanjit Mitra, James Kaiser, "Handbook for Digital Signal Processing," John Wiley, New York, 1993, pp. 127–128.

3. D. V. Tosic, M. D. Lutovac, "Symbolic computation of impulse, step and sine responses of linear time-invariant systems," Proc. 10th Int. Symp. Theor. Electrical Engineering, ISTET99, Magdeburg, Germany, Sep. 1999, pp. 653–657.

4. John M. Novak and Brian L. Evans, "Mathematica, Signals and Systems," Georgia Tech Research Corp., Atlanta, Georgia, 1995.

5. D. V. Tosic, M. D. Lutovac, and I. M. Markoski, "Symbolic derivation of transfer functions of discrete-time systems," ISTET'97, Palermo, Italia, 9–11 June 1997, pp. 311–314.

6. D. V. Tosic, "SALECAS - a package for symbolic analysis of linear circuits and systems," SMACD, Leuven, Belgium, pp. 227–230, Oct. 1996.

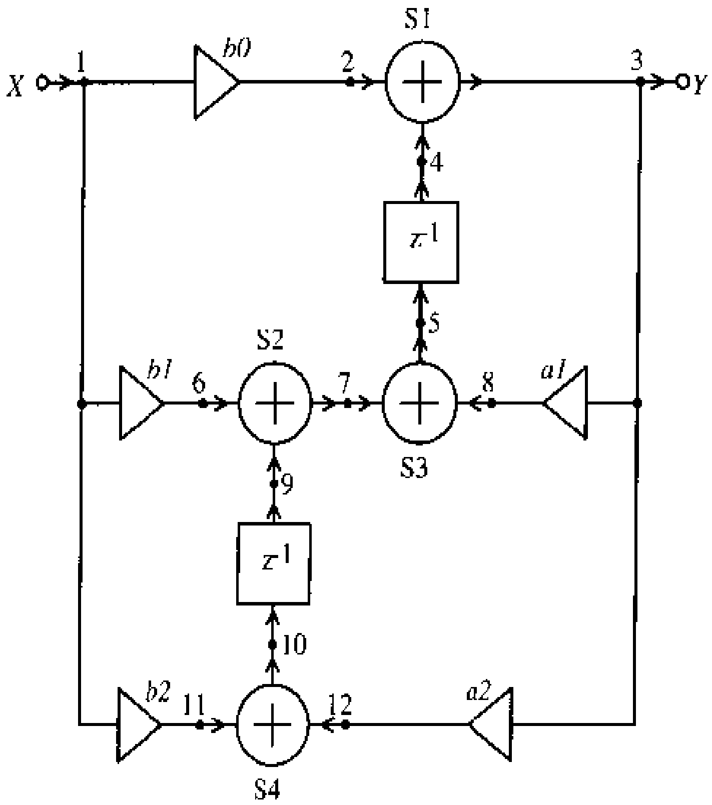
7. M. D. Lutovac, D. V. Tosic and B. L. Evans, "Advanced Filter Design for Signal Processing using MATLAB and Mathematica," http://iritel.iritel.bg.ac.yu/~lutovac/www/afdhome.htm http://galeb.etf.bg.ac.yu/~tosic/afdhome.htm http://www.ece.utexas.edu/~bevans/

A.2.2 Initialization

```
SetDirectory[HomeDirectory[]];
<<afd\math\m\clearall.m
AppendTo[$Path, "APPS"];"
Needs["SignalProcessing`Master`"]
<<afd\math\m\drawdfil.m
<<afd\math\m\drawiirf.m
```

■ A.2.3 Block Diagram

```
DrawTDF2[0,0,1,1/0.8,10];
```



■ A.2.4 Analysis (v=1/z)

```
ElementEquations = {
  Y1 == X
, Y2 == b0*Y1 + Xb0
, Y3 == Y2 + Y4
, Y4 == v*Y5
, Y5 == Y7 + Y8
, Y6 == b1*Y1 + Xb1
, Y7 == Y6 + Y9
, Y8 == a1*Y3 + Xa1
, Y9 ==  v*Y10
, Y10 == Y11 + Y12
, Y11 == b2*Y1 + Xb2
, Y12 == a2*Y3 + Xa2
};
```

```
NodeSignals = {Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11,Y12};
Response = Flatten[Solve[ElementEquations,NodeSignals]];
Y = Together[Y3/.Response]

                         2                2                        2
-(b0 X) - b1 v X - b2 v   X - v Xa1 - v   Xa2 - Xb0 - v Xb1 - v   Xb2
--------------------------------------------------------------------
                                            2
                           -1 + a1 v + a2 v
```

A.2.5 Transfer Function and Noise Transfer Functions

```
Hv   = Y /. {X -> 1, Xa1 -> 0, Xa2 -> 0, Xb0 -> 0, Xb1 -> 0, Xb2 -> 0};
Ha1v = Y /. {X -> 0, Xa1 -> 1, Xa2 -> 0, Xb0 -> 0, Xb1 -> 0, Xb2 -> 0};
Ha2v = Y /. {X -> 0, Xa1 -> 0, Xa2 -> 1, Xb0 -> 0, Xb1 -> 0, Xb2 -> 0};
Hb0v = Y /. {X -> 0, Xa1 -> 0, Xa2 -> 0, Xb0 -> 1, Xb1 -> 0, Xb2 -> 0};
Hb1v = Y /. {X -> 0, Xa1 -> 0, Xa2 -> 0, Xb0 -> 0, Xb1 -> 1, Xb2 -> 0};
Hb2v = Y /. {X -> 0, Xa1 -> 0, Xa2 -> 0, Xb0 -> 0, Xb1 -> 0, Xb2 -> 1};
  H = Collect[-Numerator[Hv],v] / (-Collect[Denominator[Hv],v]);
Ha1 = Collect[-Numerator[Ha1v],v]/(-Collect[Denominator[Ha1v],v]);
Ha2 = Collect[-Numerator[Ha2v],v]/(-Collect[Denominator[Ha2v],v]);
Hb0 = Collect[-Numerator[Hb0v],v]/(-Collect[Denominator[Hb0v],v]);
Hb1 = Collect[-Numerator[Hb1v],v]/(-Collect[Denominator[Hb1v],v]);
Hb2 = Collect[-Numerator[Hb2v],v]/(-Collect[Denominator[Hb2v],v]);
v2invz = {v->z^"-1", v^2->z^("-2")};
Print["H(z) = ", H /. v2invz ]
Print["Ha1(z) = ", Ha1 /. v2invz ]
Print["Ha2(z) = ", Ha2 /. v2invz ]
Print["Hb0(z) = ", Hb0 /. v2invz ]
Print["Hb1(z) = ", Hb1 /. v2invz ]
Print["Hb2(z) = ", Hb2 /. v2invz ]

                  -1       -2
         b0 + b1 z   + b2 z
H(z) = --------------------
                -1       -2
         1 - a1 z   - a2 z
                   -1
                  z
Ha1(z) = --------------------
                  -1       -2
           1 - a1 z   - a2 z
                   -2
                  z
Ha2(z) = --------------------
                  -1       -2
           1 - a1 z   - a2 z
```

```
                     1
HbO(z) = ------------------
                -1        -2
          1 - a1 z   - a2 z
                 -1
                z
Hb1(z) = ------------------
                -1        -2
          1 - a1 z   - a2 z
                 -2
                z
Hb2(z) = ------------------
                -1        -2
          1 - a1 z   - a2 z
```

■ A.2.6 Complex Response in Terms of v=z^(-1)

```
Y3z = Collect[-Numerator[Y],X]/(-Denominator[Y])
```

```
                 2              2                        2
(b0 + b1 v + b2 v ) X + v Xa1 + v  Xa2 + Xb0 + v Xb1 + v  Xb2
-------------------------------------------------------------
                                            2
                          1 - a1 v - a2 v
```

■ A.2.7 Transfer Function

```
H3z = H /. {b0->1, b1->2, b2->1, a1->0, a2->-1/2, v->z^(-1)}
```

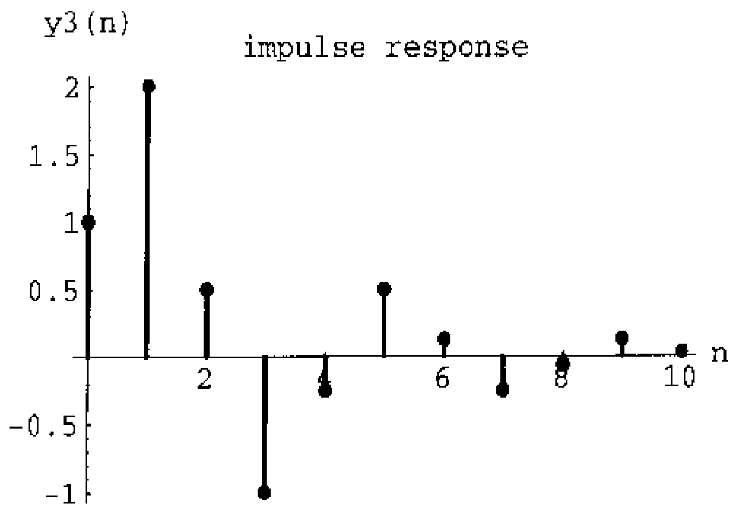
```
     -2   2
1 + z   + -
          z
-----------
      1
 1 + ----
        2
     2 z
```

■ A.2.8 Impulse Response

```
y3n = InverseZTransform[H3z,z,n]
```

```
     -I     n
-((-------)  (1 - 2 I Sqrt[2]) DiscreteStep[n])
   Sqrt[2]
------------------------------------------------ -
                          2

     I     n
 (-------)  (1 + 2 I Sqrt[2]) DiscreteStep[n]
  Sqrt[2]
--------------------------------------------- + 2 KroneckerDelta[n]
                          2
```

```
y3nDS = ComplexExpand[Coefficient[y3n,DiscreteStep[n]]]

        n Pi
  Cos[----]
         2            1/2 - n/2      n Pi
-(---------) + 2 2              Sin[----]
       n/2                            2
      2

y3nKD = Coefficient[y3n,KroneckerDelta[n]]

2

h3n = y3nDS*DiscreteStep[n] + y3nKD*KroneckerDelta[n]

                                                    n Pi
                                               Cos[----]
                                                     2            1/2 - n/2      n Pi
2 KroneckerDelta[n] + DiscreteStep[n] (-(---------) + 2 2              Sin[----])
                                                   n/2                            2
                                                  2

DiscreteSignalPlot[h3n
,{n,0,10}
,AxesLabel -> {"n","y3(n)"}
,PlotLabel -> "impulse response"
];
```



```
Table[{n,h3n}, {n,0,10}]
% //TableForm //N
```

```
                        1                        1          1         1           1
{{0, 1}, {1, 2}, {2, -}, {3, -1}, {4, -(-)}, {5, -}, {6, -}, {7, -(-)},
                        2                        4          2         8           4

       1           1          1
{8, -(--)}, {9, -}, {10, --}}
       16          8          32
```

```
0       1.
1.      2.
2.      0.5
3.      -1.
4.      -0.25
5.      0.5
6.      0.125
7.      -0.25
8.      -0.0625
9.      0.125
10.     0.03125
```

A.3 Switched Capacitor Filter - Mode 3a Analysis and Design

Miroslav D. Lutovac, Dejan V. Tosic and Brian L. Evans
lutovac@iritel.bg.ac.yu tosic@galeb.etf.bg.ac.yu bevans@ece.utexas.edu

■ A.3.1 References

1. Linear Technology Corporation
Monolithic Filter Handbook, 1990.

2. M. D. Lutovac, D. Novakovic, I. Markoski,
"Selective SC-filters with low passive sensitivity,"
Electronics Letters vol. 33, no. 8, pp. 674–675, Apr. 1997.

3. M.D.Lutovac, D. V. Tosic, D.Novakovic,
"Programmable low-pass/high-pass SC-filters,"
Proc.9th Conf. MELECON '98, Tel-Aviv, Israel May 1998, pp.673–677

4. D.V. Tosic, M.D. Lutovac, B.L. Evans, I.M. Markoski,
"A tool for symbolic analysis and design of analog active filters,"
5th Int. Workshop, Symbolic Methods, Applications, Circuit Design,
SMACD'98, Kaiserslautern, Germany, pp. 71–74, Oct. 1998.

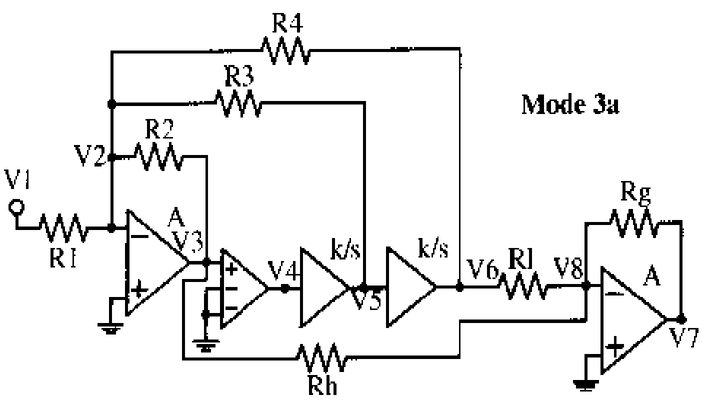
5. M. D. Lutovac, D. V. Tosic and B. L. Evans,
"Advanced Filter Design for Signal Processing using MATLAB and Mathematica,"
http://iritel.iritel.bg.ac.yu/~lutovac/www/afdhome.htm
http://galeb.etf.bg.ac.yu/~tosic/afdhome.htm
http://www.ece.utexas.edu/~bevans/

■ A.3.2 Initialization

```
SetDirectory[HomeDirectory[]];
<<afd\math\m\clearall.m
<<afd\math\m\drawafil.m
<<afd\math\m\drawasc.m
```

■ A.3.3 Schematic

```
DrawMode3a[0,0,1/2,1/0.8,8];
```



■ A.3.4 Circuit Analysis

■ Reduced Modified Nodal Analysis

```
CircuitEquations = {
  V1 == Vg
, (V2-V1)/R1 + (V2-V3)/R2 + (V2-V5)/R3 + (V2-V6)/R4 == 0
, V3 == -A*V2
, V4 == V3
, V5 == V4*k/s
, V6 == V5*k/s
, (V8-V6)/Rl + (V8-V3)/Rh + (V8-V7)/Rg == 0
, V7 == -A*V8
};
NodeVoltages = {V1,V2,V3,V4,V5,V6,V7,V8};
CircuitResponse = Together[Flatten[
 Solve[CircuitEquations,NodeVoltages]
]];
Print["V1 = ", V1 /. CircuitResponse]
Print["V7 = ", V7 /. CircuitResponse]

V1 = Vg
           2                  2                   2
V7 = -((A   R2 R3 R4 s (-(k   Rg Rh) - Rg Rl s ) Vg) /
    ((Rg Rh + Rg Rl + Rh Rl + A Rh Rl)
                                                      3
      ((R1 R2 R3 + R1 R2 R4 + R1 R3 R4 + R2 R3 R4) s   -
                2                2
        A (-(k   R1 R2 R3 s) + s   (-(k R1 R2 R4) - R1 R3 R4 s))))
```

■ A.3.5 Voltage Transfer Function

```
H = V7/V1 /. CircuitResponse //Together //Simplify;
H3a = Limit[H, A->Infinity];
Print["H(s) = ",
 Factor[Collect[Numerator[H3a],s]]/Factor[Collect[Denominator[H3a],s]]
]

                          2          2
             Rg R2 R3 R4 (k   Rh + Rl s )
H(s) = ----------------------------------------
                   2                         2
       Rh Rl R1 (k   R2 R3 + k R2 R4 s + R3 R4 s )
```

■ A.3.6 Definitions and Procedures

```
PoleQpole[H_,s_] := Module[{den,fp,Qp},
  den = Denominator[H];
  fp = Sqrt[Coefficient[den,s,0]/Coefficient[den,s,2]]/(2*Pi);
  Qp = (Coefficient[den,s,2]/Coefficient[den,s,1])*(2*Pi*fp);
  Simplify[{fp, Qp}]];
```

```
ZeroQzero[H_,s_] := Module[{fz,num,Qz0},
  num = Numerator[H];
  Qz0 = (Coefficient[num,s,2]/Coefficient[num,s,1]);
  fz = Sqrt[Coefficient[num,s,0]/Coefficient[num,s,2]]/(2*Pi);
  Simplify[{fz, Qz0*fz}]];
Sensitivity[F_,x_] := (x/F)*D[F,x];
GSP[F_,A_] := Limit[A*Sensitivity[F,A],A->Infinity]//Simplify;
PrintLabeledList[expressions_List,labels_List] := Map[
 Print[#[[1]]," = ",#[[2]]]&
,Transpose[{labels,expressions}]
];
```

■ A.3.7 Poles, Zeros, Q-Factors

```
{fp,Qp} = Simplify[PoleQpole[H,s]];
PrintLabeledList[{fp,Qp},{"fp","Qp"}];
```

```
                                2
                             A k   R1 R2 R3
        Sqrt[-----------------------------------------------------]
             R1 R2 R3 + R1 R2 R4 + R1 R3 R4 + A R1 R3 R4 + R2 R3 R4
fp = -------------------------------------------------------------
                                2 Pi
                                 2
                              A k   R1 R2 R3
Qp = (Sqrt[-----------------------------------------------------]
           R1 R2 R3 + R1 R2 R4 + R1 R3 R4 + A R1 R3 R4 + R2 R3 R4
    (R1 R2 R3 + R1 R2 R4 + R1 R3 R4 + A R1 R3 R4 + R2 R3 R4)) / (A k R1 R2 R4)
```

```
fp0 = Limit[fp, A -> Infinity];
Qp0 = Simplify[Limit[Qp, A -> Infinity]/.k->1];
PrintLabeledList[{fp0,Qp0},{"fp","Qp"}];
```

```
            2
           k   R2
      Sqrt[-----]
            R4
fp = -----------
        2 Pi
             R2
      R3 Sqrt[--]
             R4
Qp = -----------
         R2
```

```
{fz,Qz} = Simplify[ZeroQzero[H,s]];
PrintLabeledList[{fz,Qz},{"fz","Qz"}];

                                          1
Power::infy: Infinite expression - encountered.
                                          0
             2
            k   Rh
      Sqrt[-----]
             Rl
fz = -----------
         2 Pi
Qz = ComplexInfinity
```

■ A.3.8 Gain-Sensitivity Product (GSP)

```
GSPfp = GSP[fp,A];
GSPQp = GSP[Qp,A];
PrintLabeledList[{GSPfp,GSPQp},{"GSPfp","GSPQp"}];

          1    R2      R2      R2
GSPfp = - + ---- + ---- + ----
          2   2 R1    2 R3    2 R4
            1      R2      R2      R2
GSPQp = -(-) - ---- - ---- - ----
            2     2 R1    2 R3    2 R4
```

■ A.3.9 Design

■ Find Element Values

```
DesignMode3a[K_,Qp_,wp_,wz_,fclk_,P_:100,R1_:R1nom,R2_:R2nom,Rh_:Rhnom]:=
 Module[{R3,R4,Rl,Rg},
 R4 = R2*(2*Pi*fclk/(P*wp))^2;
 R3 = Qp*Sqrt[R2*R4];
 Rl = Rh*(2*Pi*fclk/(P*wz))^2;
 Rg = K*Rh*R1/R2;
{R1,R2,R3,R4,Rl,Rh,Rg}
];
{R1,R2,R3,R4,Rl,Rh,Rg} = Together[
                              DesignMode3a[K,Q,W,Z,Fc,P,R1n,R2n,Rhn]];
PrintLabeledList[{R1,R2,R3,R4,Rl,Rh,Rg}
                  ,{"R1","R2","R3","R4","Rl","Rh","Rg"}];

R1 = R1n
R2 = R2n
                   2    2     2
                 Fc   Pi   R2n
R3 = 2 Q Sqrt[------------]
                    2   2
                   P   W
```

```
          2    2
      4 Fc   Pi   R2n
R4 = -------------
           2   2
          P   W
          2    2
      4 Fc   Pi   Rhn
R1 = -------------
           2   2
          P   Z
Rh = Rhn
      K Rhn R1n
Rg = ---------
         R2n
```

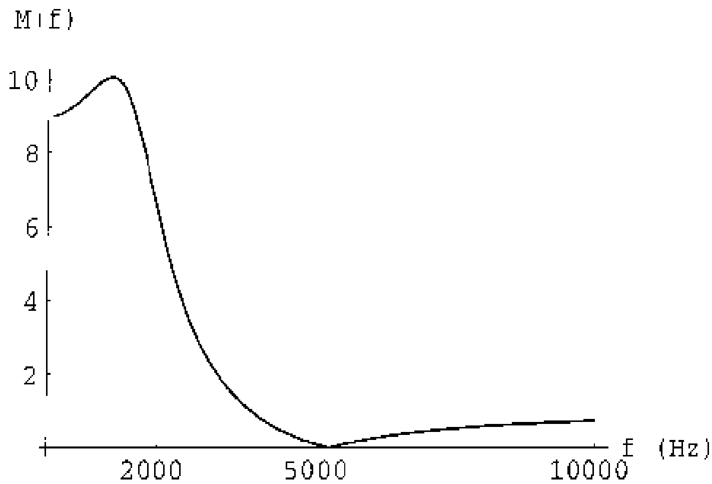
■ **Test**

```
H3atest  = Simplify[ExpandAll[Together[Limit[H , A -> Infinity]] /.
 {Sqrt[x_^2*y_^2/z_^2] -> x*y/z, Sqrt[x_^2*y_^2*p_^2/z_^2] -> x*y*p/z
, Sqrt[x_^2*y_^2*p_^2/(z_^2*n_^2)] -> x*y*p/(z*n)} ]];
num = Numerator[H3atest];
den = Denominator[H3atest];
numlist = CoefficientList[num,s];
denlist = CoefficientList[den,s];
K3at =(numlist[[3]]/denlist[[3]]);
H3at =  K3at *
Simplify[num/numlist[[3]]]/Simplify[den/denlist[[3]]] /. k-> 2*Pi*Fc/P
```

```
     2     2
 K (s  + Z )
-------------
 2    s W     2
s  + --- + W
       Q
```

Design Examples

```
values = {K ->1, Q -> 1.0349, W -> 2*Pi*1710.9457, Z -> 2*Pi*5129.3034
  , Fc -> 256.*10^3, P -> 100, R12 ->23.16*10^3
  , R22 ->10.*10^3, Rh2 ->238.6*10^3} //N;
h1 = H  /. k-> 2*Pi*Fc/P /.values //N;
H3atest = Limit[h1, A->Infinity];
H3atest =H3atest /. Sqrt[x_^2] -> x;
num = Numerator[H3atest];
den = Denominator[H3atest];
numlist = CoefficientList[num,s];
denlist = CoefficientList[den,s];
K3at =(numlist[[3]]/denlist[[3]]);
H3at =  K3at * Simplify[num/numlist[[3]]]/Simplify[den/denlist[[3]]]

                    9         2
   1. (1.03867 10   + 1. s )
-------------------------------
            8                       2
1.15567 10   + 10387.7 s + 1. s

Rexample1 = N[{R1,R2,R3,R4,R1,Rh,Rg}/. values] /. Sqrt[R2n^2]->R2n;
PrintLabeledList[Rexample1,{"R1","R2","R3","R4","Rl","Rh","Rg"}];

R1 = R1n
R2 = R2n
R3 = 1.54847 R2n
R4 = 2.23876 R2n
Rl = 0.249094 Rhn
Rh = Rhn
      1. Rhn R1n
Rg = ----------
         R2n

Plot[{Abs[H3at] /. s -> I*2*Pi*f}
, {f, 100, 10000}
,PlotRange -> All
,Ticks -> {{0,2000,5000,10000},{0,2,4,6,8,10}}
, AxesLabel -> {"f (Hz)","M(f)"}
];
```



■ A.3.10 Optimization

■ Find R2/R1 for Low Gain-Sensitivity Product

```
sf = (Simplify[N[Together[GSPfp /. values]]
      /. Sqrt[x_^2] -> x/. Sqrt[x_^2*y_^2] -> x*y ]
      /. Sqrt[R2n^2] -> R2n)

           0.5 R2n
1.04624 + -------
             R1n
```

■ Remark

We have to choose R1 > R2 to minimize GSP.

A.4 OTA-C General Biquad Analysis and Design

Miroslav D. Lutovac, Dejan V. Tosic and Brian L. Evans
lutovac@iritel.bg.ac.yu tosic@galeb.etf.bg.ac.yu bevans@ece.utexas.edu

A.4.1 References

1. Wai-Kai-Chen, Ed.,
"The Circuits and Filters Handbook," p. 2479,
"CRC Press," "Boca Raton, Florida," "1995."

2. D.V. Tosic, M.D. Lutovac, B.L. Evans, I.M. Markoski,
"A tool for symbolic analysis and design of analog active filters,"
5th Int. Workshop, Symbolic Methods, Applications, Circuit Design,
SMACD'98, Kaiserslautern, Germany, pp. 71–74, Oct. 1998.

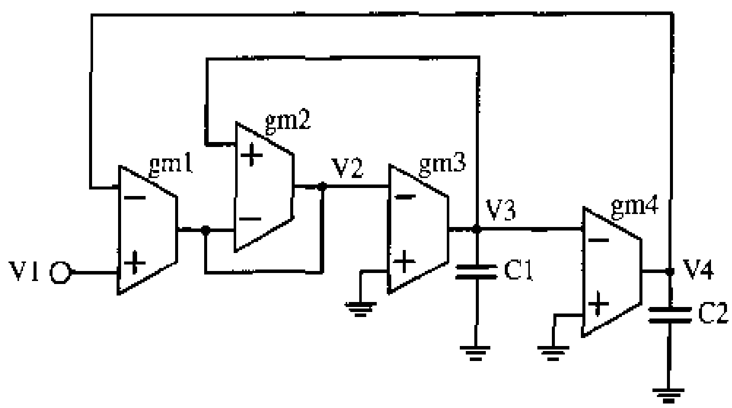
3. M. D. Lutovac, D. V. Tosic and B. L. Evans,
"Advanced Filter Design for Signal Processing using MATLAB and Mathematica,"
http://iritel.iritel.bg.ac.yu/~lutovac/www/afdhome.htm
http://galeb.etf.bg.ac.yu/~tosic/afdhome.htm
http://www.ece.utexas.edu/~bevans/

A.4.2 Initialization

```
SetDirectory[HomeDirectory[]];
<<afd\math\m\clearall.m
<<afd\math\m\drawafil.m
<<afd\math\m\drawaota.m
```

A.4.3 Schematic

```
DrawOTAb[0, 0, 1/2, 1/08, 8]
```



A.4.4 Circuit Analysis

Reduced Modified Nodal Analysis

```
CircuitEquations = {
  V1 == Vg
, -(V1-V4)*gm1 - (V3-V2)*gm2 == 0
, -(-V2)*gm3 + V3/(1/(s*C1)) == 0
, -(-V3)*gm4 + V4/(1/(s*C2)) == 0
};

NodeVoltages = {V1,V2,V3,V4};
CircuitResponse = Together[Flatten[
 Solve[CircuitEquations,NodeVoltages]
]];
```

A.4.5 Voltage Transfer Function

```
H = V4/V1 /. CircuitResponse //Together //Simplify;
Print["H(s) = ", H]
Hhp = V2/V1 /. CircuitResponse //Together //Simplify;
Print["Hhp(s) = ",  Hhp]
Hbp = V3/V1 /. CircuitResponse //Together //Simplify;
Print["Hbp(s) = ", Hbp]

                        gm1 gm3 gm4
H(s) = -----------------------------------------
                                              2
       gm1 gm3 gm4 + C2 gm2 gm3 s + C1 C2 gm2 s
                                    2
                        C1 C2 gm1 s
Hhp(s) = -----------------------------------------
                                                2
         gm1 gm3 gm4 + C2 gm2 gm3 s + C1 C2 gm2 s
                        C2 gm1 gm3 s
Hbp(s) = -(-----------------------------------------)
                                                  2
           gm1 gm3 gm4 + C2 gm2 gm3 s + C1 C2 gm2 s
```

A.4.6 Definitions and Procedures

```
PoleQpole[H_,s_] := Module[{den,fp,Qp},
  den = Denominator[H];
  fp = Sqrt[Coefficient[den,s,0]/Coefficient[den,s,2]]/(2*Pi);
  Qp = (Coefficient[den,s,2]/Coefficient[den,s,1])*(2*Pi*fp);
  Simplify[{fp, Qp}]];
ZeroQzero[H_,s_] := Module[{fz,num,Qz0},
  num = Numerator[H];
  Qz0 = (Coefficient[num,s,2]/Coefficient[num,s,1]);
  fz = Sqrt[Coefficient[num,s,0]/Coefficient[num,s,2]]/(2*Pi);
  Simplify[{fz, Qz0*fz}]];
```

```
PrintLabeledList[expressions_List,labels_List] := Map[
 Print[#[[1]]," = ",#[[2]]]&
,Transpose[{labels,expressions}]
];
```

A.4.7 Poles, Zeros, Q-Factors

```
{fp,Qp} = Simplify[PoleQpole[H,s]];
Klp = H /. s-> 0
PrintLabeledList[{fp,Qp},{"fp","Qp"}];

1
          gm1 gm3 gm4
     Sqrt[-----------]
           C1 C2 gm2
fp = -----------------
           2 Pi
              gm1 gm3 gm4
     C1 Sqrt[-----------]
              C1 C2 gm2
Qp = --------------------
             gm3
```

A.4.8 Design

Find Element Values

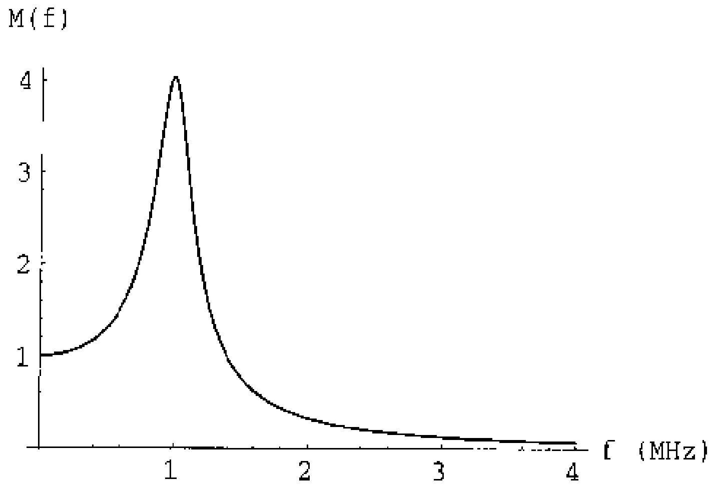
```
DesignOTA1[Qp_,wp_,C1_,C2_,gm1_,gm2_] :=
 Module[{gm3,gm4},
 gm3 = C1*wp/Qp;
 gm4 = C2*wp*Qp*gm2/gm1;
{C1,C2,gm1,gm2,gm3,gm4}
];
{C1,C2,gm1,gm2,gm3,gm4} = Together[DesignOTA1[Q,W,c1,c2,g1,g2]];
PrintLabeledList[{C1,C2,gm1,gm2,gm3,gm4},{"C1","C2","gm1","gm2","gm3","gm4"}];

C1 = c1
C2 = c2
gm1 = g1
gm2 = g2
      c1 W
gm3 = ----
       Q
      c2 g2 Q W
gm4 = ---------
         g1
```

■ A.4.9 Test

```
Simplify[H]

          2
       Q W
----------------
   2           2
Q s  + s W + Q W
```

■ Design Examples

```
values = {Q -> 4., W -> N[2*Pi*10^6] ,
  c1 -> 10.*10^(-12), c2 ->10.*10^(-12), g1 -> g , g2 -> g} //N;
h1 = Together[H  /. values] /. 1. -> 1;
Print["gm3 = ",10^3*gm3 /. values, " mS "]
Print["gm4 = ",10^3*gm4 /. values, " mS "]
h = (Numerator[h1]/g)/ (Simplify[Denominator[h1]/g])

gm3 = 0.015708 mS
gm4 = 0.251327 mS
                        13
          3.94784 10
--------------------------------
          13             6      2
3.94784 10   + 1.5708 10  s + s

PrintLabeledList[N[{Q,W/(2*Pi)} /. values]
,{"Qp","fp (Hz)"}];
Print["-----------------------"]
Rexample1 = N[{C1*10^(12),C2*10^(12),gm1,gm2,gm3*10^(3),gm4*10^(3)}/. values] ;
PrintLabeledList[Rexample1
                ,{"C1 (pF)","C2 (pF)","gm1","gm2","gm3 (mS)","gm4 (mS)"}];

Qp = 4.
                6
fp (Hz) = 1. 10
------------------------
C1 (pF) = 10.
C2 (pF) = 10.
gm1 = g
gm2 = g
gm3 (mS) = 0.015708
gm4 (mS) = 0.251327

Plot[{Abs[h] /. s -> I*2*Pi*f*10^6}
, {f, 0.01, 4}
,PlotRange -> All
, AxesLabel -> {"f (MHz)","M(f)"}
];
```


M(f)
4
3
2
1
1
2
3
4
f (MHz)

A.5 Lowpass-Medium-Q-Factor Active RC Filter Analysis and Design

Miroslav D. Lutovac, Dejan V. Tosic and Brian L. Evans

lutovac@iritel.bg.ac.yu tosic@galeb.etf.bg.ac.yu bevans@ece.utexas.edu

■ A.5.1 References

1. G. S. Moschytz and P. Horn,
Active Filter Design Handbook
For Use with Programmable Pocket Calculators and Mini Computers
John Wiley and Sons, Ltd., New York, 1981, pp. 38–39.

2. R. P. Sallen and E. L. Key,
"A practical method of designing RC active filters,"
IRE Trans. Circuit Theory, CT-2, Mar. 1955, pp. 74–85.

3. D.V. Tosic, M.D. Lutovac, B.L. Evans, I.M. Markoski,
"A tool for symbolic analysis and design of analog active filters,"
5th Int. Workshop, Symbolic Methods, Applications, Circuit Design,
SMACD'98, Kaiserslautern, Germany, pp. 71–74, Oct. 1998.

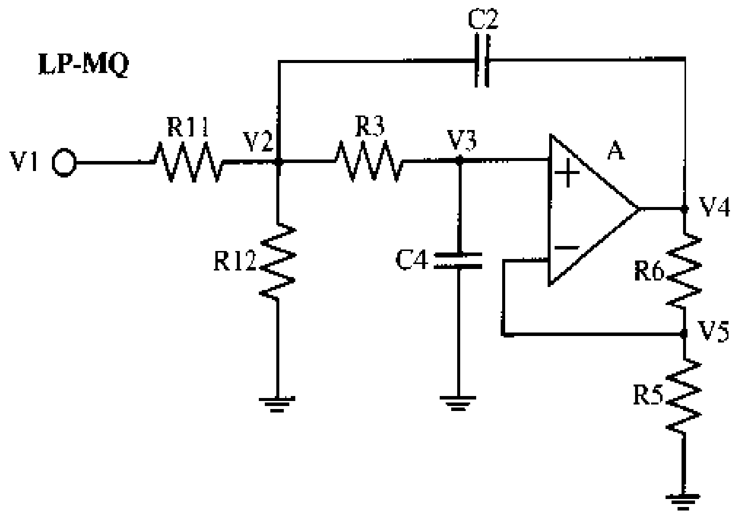
4. M. D. Lutovac, D. V. Tosic and B. L. Evans,
"Advanced Filter Design for Signal Processing using MATLAB and Mathematica,"
http://iritel.iritel.bg.ac.yu/~lutovac/www/afdhome.htm
http://galeb.etf.bg.ac.yu/~tosic/afdhome.htm
http://www.ece.utexas.edu/~bevans/

■ A.5.2 Initialization

```
SetDirectory[HomeDirectory[]];
<<afd\math\m\clearall.m
<<afd\math\m\drawafil.m
<<afd\math\m\drawarc.m
```

A.5.3 Schematic

```
DrawLPMQ[0,0,1/2,1/0.8];
```



A.5.4 Circuit Analysis

Reduced Modified Nodal Analysis

```
CircuitEquations = {V1 == Vg
, (V2-V1)/R11 + V2/R12 + (V2-V3)/R3 + (V2-V4)/(1/(s*C2)) == 0
, (V3-V2)/R3 + V3/(1/(s*C4)) == 0
, (V5-V4)/R6 + V5/R5 == 0
, (V3-V5)*A == V4};
NodeVoltages = {V1,V2,V3,V4,V5};
CircuitResponse = Together[Flatten[
 Solve[CircuitEquations,NodeVoltages]
]];
Print["V1 = ", V1 /. CircuitResponse]
Print["V4 = ", V4 /. CircuitResponse]
```

```
V1 = Vg
               2
V4 = -((A  R12 R3 R5 Vg) /
          2
      (A  C2 R11 R12 R3 R5 s - (-R5 - A R5 - R6) (1 + C4 R3 s)
         (R11 R12 + R11 R3 + R12 R3 + C2 R11 R12 R3 s) +
        (-R5 - A R5 - R6) (R11 R12 + A C2 R11 R12 R3 s))) -
   (A R12 R3 (-R5 - A R5 - R6) Vg) /
       2
    (A  C2 R11 R12 R3 R5 s - (-R5 - A R5 - R6) (1 + C4 R3 s)
       (R11 R12 + R11 R3 + R12 R3 + C2 R11 R12 R3 s) +
      (-R5 - A R5 - R6) (R11 R12 + A C2 R11 R12 R3 s))
```

■ A.5.5 Voltage Transfer Function

```
H = V4/VI /. CircuitResponse //Together //Simplify;
Ha = Limit[H, A->Infinity];
Print["H(s) = ",
 Collect[Numerator[Ha],s]/Collect[Denominator[Ha],s]]
```

```
H(s) = (R12 (R5 + R6)) /
  (R11 R5 + R12 R5 + (C4 R11 R12 R5 + C4 R11 R3 R5 + C4 R12 R3 R5 - C2 R11 R12 R6) s +

                                2
     C2 C4 R11 R12 R3 R5 s )
```

■ A.5.6 Definitions and Procedures

```
PoleQpole[H_,s_] := Module[{den,fp,Qp},
  den = Denominator[H];
  fp = Sqrt[Coefficient[den,s,0]/Coefficient[den,s,2]]/(2*Pi);
  Qp = (Coefficient[den,s,2]/Coefficient[den,s,1])*(2*Pi*fp);
  Simplify[{fp, Qp}]];
ZeroQzero[H_,s_] := Module[{fz,num,Qz0},
  num = Numerator[H];
  Qz0 = (Coefficient[num,s,2]/Coefficient[num,s,1]);
  fz = Sqrt[Coefficient[num,s,0]/Coefficient[num,s,2]]/(2*Pi);
  Simplify[{fz, Qz0*fz}]];
Sensitivity[F_,x_] := (x/F)*D[F,x];
GSP[F_,A_] := Limit[A*Sensitivity[F,A],A->Infinity]//Simplify;
PrintLabeledList[expressions_List,labels_List] := Map[
 Print[#[[1]]," = ",#[[2]]]&
,Transpose[{labels,expressions}]
];
```

■ A.5.7 Poles, Zeros, Q-Factors

```
{fp,Qp} = Simplify[PoleQpole[H,s]];
PrintLabeledList[{fp,Qp},{"fp","Qp"}];
```

```
              R11 + R12
     Sqrt[----------------]
          C2 C4 R11 R12 R3
fp = ----------------------
              2 Pi
                               R11 + R12
Qp = (C2 C4 R11 R12 Sqrt[----------------] R3 (R5 + A R5 + R6)) /
                          C2 C4 R11 R12 R3

(C2 R11 R12 R5 + C4 R11 R12 R5 + A C4 R11 R12 R5 + C4 R11 R3 R5 + A C4 R11 R3 R5 +
  C4 R12 R3 R5 + A C4 R12 R3 R5 + C2 R11 R12 R6 - A C2 R11 R12 R6 + C4 R11 R12 R6 +
  C4 R11 R3 R6 + C4 R12 R3 R6)
```

A.5.8 Gain-Sensitivity Product (GSP)

```
GSPfp = GSP[fp,A];
GSPQp = GSP[Qp,A];
PrintLabeledList[{GSPfp,GSPQp},{"GSPfp","GSPQp"}];

GSPfp = 0
                                              2
                               C2 R11 R12 (R5 + R6)
GSPQp = -----------------------------------------------------------------
        R5 (C4 R11 R12 R5 + C4 R11 R3 R5 + C4 R12 R3 R5   C2 R11 R12 R6)
```

A.5.9 Design

Find Element Values

```
DesignLPMQ[K_,Qp_,wp_,P_,C2x_,C4x_,R5x_] :=  Module[
{C2,C4,R1,R11,R12,R3,R5,R6,Ko},
 C2 = C2x;
 C4 = C4x;
 R1 = 1/(wp*Sqrt[C2x*C4x*P]);
 R3 = P*R1;
 R5 = R5x;
 R6 = R5*((1+P)*C4/C2-Sqrt[P*C4/C2]/Qp);
 Ko = 1+R6/R5;
 R11 = R1*Ko/K;
 R12 = R1*Ko/(Ko-K);
{R11,R12,C2,R3,C4,R5,R6}
];
{R11,R12,C2,R3,C4,R5,R6} = Together[DesignLPMQ[K,Q,W,P,C2x,C4x,R5x]];
```

A.5.10 Design Example

```
values = {K -> 1., Q -> 7.5, W -> 2*Pi*2500., P -> 1.5333078
 , C2x -> 33.*10^(-9), C4x -> 10.*10^(-9), R5x -> 6800.} //N;
PrintLabeledList[N[{K,Q,W/(2*Pi),P} /. values]
,{"K","Qp","fp (Hz)","P"} ];
Print["----------------"]
PrintLabeledList[Together[{R11,R12,C2*10^9,R3,C4*10^9,R5,R6} /. values],
{"R11 (ohm)","R12 (ohm)","C2 (nF)","R3 (ohm)","C4 (nF)","R5 (ohm)","R6 (ohm)"}];

K = 1.
Qp = 7.5
fp (Hz) = 2500.
P = 1.53331
----------------
R11 (ohm) = 4745.54
R12 (ohm) = 7011.9
C2 (nF) = 33.
R3 (ohm) = 4339.48
C4 (nF) = 10.
R5 (ohm) = 6800.
R6 (ohm) = 4602.13
```

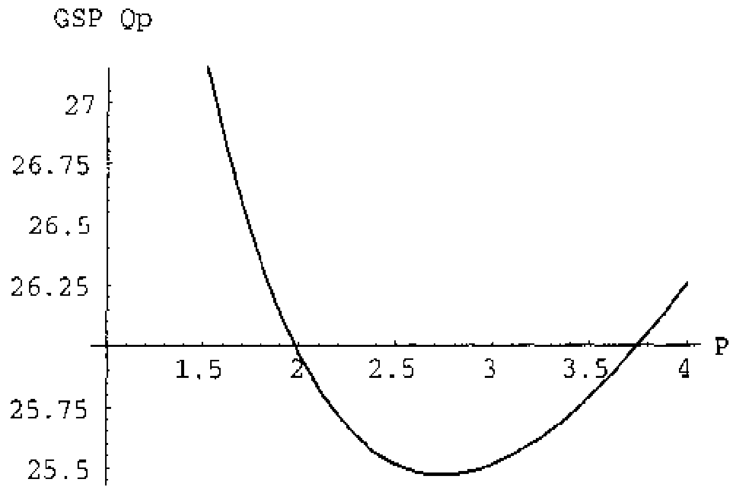
A.5.11 Optimization

Find P for Low Gain-Sensitivity Product

```
values = {K -> 1.476288, Q -> 7.5, W -> 2*Pi*2500.
 , C2x -> 100.*10^(-9), C4x -> 15.*10^(-9), R5x -> 10000.} //N;
gspQp = Together[Simplify[GSPQp] /. values]
```

```
                                              2
0.435711 (7.66667 - 0.344265 Sqrt[P] + 1. P)
--------------------------------------------
                  Sqrt[P]
```

```
P1 = 1.;
P2 = 4.;
Plot[gspQp /. values
, {P, P1, P2}
, AxesLabel -> {"P","GSP Qp"}
];
```



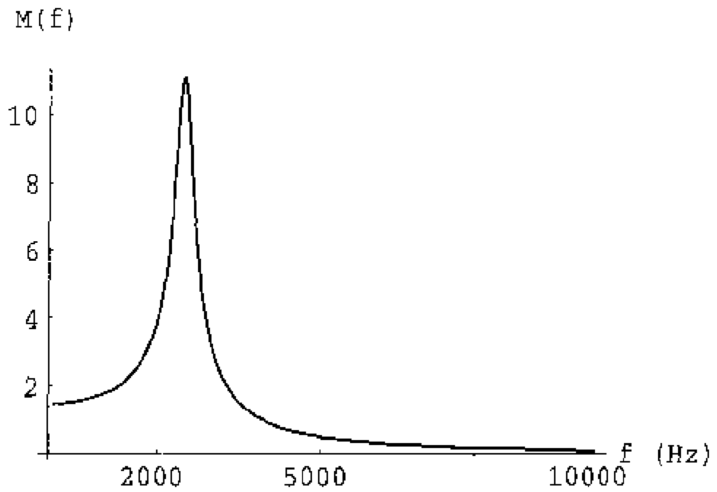

```
{GSPmin,Pset} = FindMinimum[gspQp,{P,P1,P2}]
```

```
{25.4701, {P -> 2.74571}}
```

```
PrintLabeledList[N[{K,Q,W/(2*Pi),P} /. values /.Pset]
,{"K","Qp","fp (Hz)","P"} ];
Print["---------------"]
PrintLabeledList[Together[{R11,R12,C2*10^9,R3,C4*10^9,R5,R6} /. values /.Pset],
{"R11 (ohm)","R12 (ohm)","C2 (nF)","R3 (ohm)","C4 (nF)","R5 (ohm)","R6 (ohm)"}]
```

```
K = 1.47629
Qp = 7.5
fp (Hz) = 2500.
P = 2.74571
---------------
R11 (ohm) = 991.99
```

```
                        9
R12 (ohm) = 1.81368 10
C2 (nF) = 100.
R3 (ohm) = 2723.72
C4 (nF) = 15.
R5 (ohm) = 10000.
R6 (ohm) = 4762.89

Hhpmq = Together[Ha  /.Pset/. values] ;
num = Numerator[Hhpmq];
den = Denominator[Hhpmq];
numlist = CoefficientList[num,s];
denlist = CoefficientList[den,s];
Hhpmq = (1/denlist[[3]]) * num/(den/denlist[[3]])//Factor

                   8
         3.64259 10
-----------------------------
         8                  2
2.4674 10  + 2094.4 s + 1. s

Plot[{Abs[Hhpmq] /. s -> N[I*2*Pi*f]}
, {f, 100, 10000}
, PlotRange -> All
, Ticks -> {{0,2000,5000,10000},{0,2,4,6,8,10}}
, AxesLabel -> {"f (Hz)","M(f)"} ];
```



- **Remark**

We choose P=2.7457 as a good choice because we obtain small GSP and 1/R12=0.

A.6 General Purpose High-Q-Factor RC Filter Analysis and Design

Miroslav D. Lutovac, Dejan V. Tosic and Brian L. Evans
lutovac@iritel.bg.ac.yu tosic@galeb.etf.bg.ac.yu bevans@ece.utexas.edu

A.6.1 References

1. G. S. Moschytz and P. Horn,
Active Filter Design Handbook
For Use with Programmable Pocket Calculators and Mini Computers
John Wiley and Sons, Ltd., New York, 1981 pp. 64–65.

2. W. J. Kerwin, L. P. Huelsman, and R. W. Newcomb,
State-variable synthesis for insensitive integrated circuit transfer functions,
IEEE J. Solid-State Circuits, SC–2, pp. 87–92, September, 1967.

3. D.V. Tosic, M.D. Lutovac, B.L. Evans, I.M. Markoski,
"A tool for symbolic analysis and design of analog active filters,"
5th Int. Workshop, Symbolic Methods, Applications, Circuit Design,
SMACD'98, Kaiserslautern, Germany, pp. 71–74, Oct. 1998.

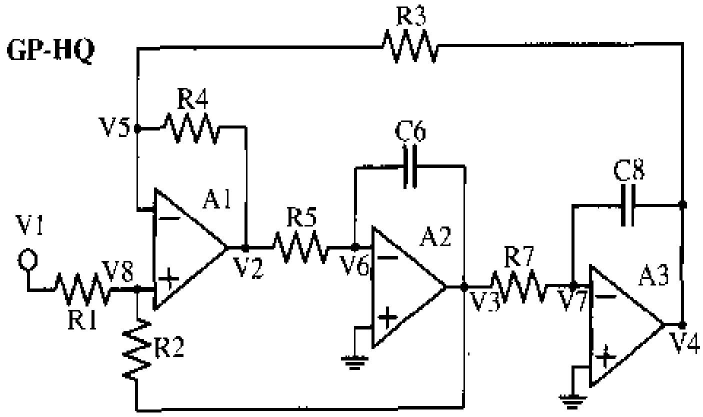
4. M. D. Lutovac, D. V. Tosic and B. L. Evans,
"Advanced Filter Design for Signal Processing using MATLAB and Mathematica,"
http://iritel.iritel.bg.ac.yu/~lutovac/www/afdhome.htm
http://galeb.etf.bg.ac.yu/~tosic/afdhome.htm
http://www.ece.utexas.edu/~bevans/

A.6.2 Initialization

```
SetDirectory[HomeDirectory[]];
<<afd\math\m\clearall.m
<<afd\math\m\drawafil.m
<<afd\math\m\drawarc.m
```

A.6.3 Schematic

```
DrawGPHQ[0,0,1/2,1/0.8];
```



A.6.4 Circuit Analysis

Reduced Modified Nodal Analysis

```
CircuitEquations = {V1 == Vg
, V2 == A*(V8 - V5)
, V3 == -A*V6
, V4 == -A*V7
, (V5-V2)/R4 + (V5-V4)/R3 == 0
, (V6-V2)/R5 + (V6-V3)/(1/(s*C6)) ==0
, (V7-V3)/R7 + (V7-V4)/(1/(s*C8)) ==0
, (V8-V1)/R1 + (V8-V3)/R2 ==0};
NodeVoltages = {V1,V2,V3,V4,V5,V6,V7,V8};
CircuitResponse = Together[Flatten[
 Solve[CircuitEquations,NodeVoltages]
]];
```

A.6.5 Voltage Transfer Function

```
H = V4/V1 /. CircuitResponse //Together //Simplify;
Ha = Limit[H, A->Infinity];
Print["H(s) = ",
 Collect[Numerator[Ha],s]/Collect[Denominator[Ha],s]]

H(s) = (R2 (R3 + R4)) /
  (R1 R4 + R2 R4 + (C8 R1 R3 R7 + C8 R1 R4 R7) s +
                                                2
    (C6 C8 R1 R3 R5 R7 + C6 C8 R2 R3 R5 R7) s )

Klp = Factor[Ha /. s->0]

R2 (R3 + R4)
------------
(R1 + R2) R4

Hh = V2/V1 /. CircuitResponse //Together //Simplify;
Hha = Limit[Hh, A->Infinity];
        Print["Hh(s) = ",
 Collect[Numerator[Hha],s]/Collect[Denominator[Hha],s]]

                                      2
Hh(s) = (C6 C8 R2 (R3 + R4) R5 R7 s ) /
  (R1 R4 + R2 R4 + (C8 R1 R3 R7 + C8 R1 R4 R7) s +
                                                2
    (C6 C8 R1 R3 R5 R7 + C6 C8 R2 R3 R5 R7) s )

Khp = Factor[Limit[Hha , s->Infinity]]

R2 (R3 + R4)
------------
(R1 + R2) R3
```

```
Hb = V3/V1 /. CircuitResponse //Together //Simplify;
Hba = Limit[Hb, A->Infinity];
Print["Hb(s) = ",
 Collect[Numerator[Hba],s]/Collect[Denominator[Hba],s]]

Hb(s) = -((C8 R2 (R3 + R4) R7 s) /
    (R1 R4 + R2 R4 + (C8 R1 R3 R7 + C8 R1 R4 R7) s +
                                                    2
      (C6 C8 R1 R3 R5 R7 + C6 C8 R2 R3 R5 R7) s ))

num = Numerator[Hba];
den = Denominator[Hba];
numlist = CoefficientList[num,s];
denlist = CoefficientList[den,s];
Hbp = Factor[numlist[[2]]/denlist[[2]]]

  R2
-(--)
  R1
```

A.6.6 Definitions and Procedures

```
PoleQpole[H_,s_] := Module[{den,fp,Qp},
  den = Denominator[H];
  fp = Sqrt[Coefficient[den,s,0]/Coefficient[den,s,2]]/(2*Pi);
  Qp = (Coefficient[den,s,2]/Coefficient[den,s,1])*(2*Pi*fp);
  Simplify[{fp, Qp}]];
ZeroQzero[H_,s_] := Module[{fz,num,Qz0},
  num = Numerator[H];
  Qz0 = (Coefficient[num,s,2]/Coefficient[num,s,1]);
  fz = Sqrt[Coefficient[num,s,0]/Coefficient[num,s,2]]/(2*Pi);
  Simplify[{fz, Qz0*fz}]];
Sensitivity[F_,x_] := (x/F)*D[F,x];
GSP[F_,A_] := Limit[A*Sensitivity[F,A],A->Infinity]//Simplify;
GSPepsA[F_,epsA_] := -(1/epsA)*Sensitivity[F,epsA]//Together;
PrintLabeledList[expressions_List,labels_List] := Map[
 Print[#[[1]]," = ",#[[2]]]&
,Transpose[{labels,expressions}]
];
```

A.6.7 Poles, Zeros, Q-Factors

```
{fp,Qp} = Simplify[PoleQpole[H,s]];
fp0 = Together[Limit[fp,A-> Infinity]];
Qp0 = Limit[Qp,A-> Infinity];
PrintLabeledList[{fp0,Qp0},{"fp","Qp"}];
```

```
                      R4
          Sqrt[-------------]
               C6 C8 R3 R5 R7
fp = --------------------
               2 Pi
                                        R4
        C6 (R1 + R2) R3 R5 Sqrt[-------------]
                                   C6 C8 R3 R5 R7
Qp = ---------------------------------------
                    R1 (R3 + R4)
```

A.6.8 Gain-Sensitivity Product (GSP)

```
fpepsA = Together[fp /. A->1/e];
QpepsA = Together[Qp /. A->1/e];
GSPQp = Simplify[GSPepsA[QpepsA,e] /. e->0];
GSPfp = Simplify[GSPepsA[fpepsA,e] /. e->0];
PrintLabeledList[{GSPfp,GSPQp},{"GSPfp","GSPQp"}];
```

```
                  2                                2        2
         -(R1 R3 ) + 2 R1 R3 R4 + 3 R2 R3 R4 + R1 R4  + R2 R4
GSPfp = ---------------------------------------------------
                       2 R3 (R1 R4 + R2 R4)
                     2   2                     2                2   2                2   3
GSPQp = (2 C6 R1  R3  R4 R5 + 4 C6 R1 R2 R3  R4 R5 + 2 C6 R2  R3  R4 R5 - C8 R1  R3  R7 -

          2   2                  2               2   2                 2       2
    C8 R1  R3  R4 R7 + 3 C8 R1 R2 R3  R4 R7 + 2 C8 R2  R3  R4 R7 - 3 C8 R1  R3 R4  R7 -

                    2          2   3                  3
    2 C8 R1 R2 R3 R4  R7 - C8 R1  R4  R7 - C8 R1 R2 R4  R7) /
  (2 R3 (R1 R4 + R2 R4) (C8 R1 R3 R7 + C8 R1 R4 R7))
```

A.6.9 Design

Find Element Values

```
DesignGPHQ[Qp_,wp_,Cx_,Rx_] :=
 Module[{C6,C8,R1,R2,R3,R4,R5,R7,Ro},
 C6 = Cx;
 C8 = Cx;
 Ro = 1/(wp*Cx);
 R1 = Rx;
 R3 = Rx;
 R5 = Rx;
 R7 = Rx;
 R4 = Rx*(Rx/Ro)^2;
 R2 = Rx*(Qp*(1+R4/Rx)/Sqrt[R4/Rx]-1);
 {R1,R2,R3,R4,R5,C6,R7,C8}];
 {R1,R2,R3,R4,R5,C6,R7,C8} = DesignGPHQ[Q,Wp,Co,Rd];
```

Design Examples

```
values = {Q -> 6., Wp -> 2*Pi*1500.
, Co -> 68.*10^(-9), Rd -> 1800.} //N;
gspQpfp = Together[Abs[GSPQp/2]+Abs[Q*GSPfp] /. values];
```

```
PrintLabeledList[N[{Q,Wp/(2*Pi),gspQpfp} /. values]
,{"Qp","fp (Hz)","GSP"} ];
Print["-----------------"]
PrintLabeledList[Together[{R1,R2,R3,R4,R5,C6*10^9,R7,C8*10^9} /. values]
,{"R1 (ohm)","R2 (ohm)","R3 (ohm)","R4 (ohm)","R5 (ohm)","C6 (nF)"
,"R7 (ohm)","C8 (nF)"}];
```

```
Qp = 6.
fp (Hz) = 1500.
GSP = 17.1412
-----------------
R1 (ohm) = 1800.
R2 (ohm) = 20020.9
R3 (ohm) = 1800.
R4 (ohm) = 2395.4
R5 (ohm) = 1800.
C6 (nF) = 68.
R7 (ohm) = 1800.
C8 (nF) = 68.
```

```
values = {Q -> 25., Wp -> 2*Pi*1500.
, Co -> 22.*10^(-9), Rd -> 4700.} //N;
gspQpfp = Together[Abs[GSPQp/2]+Abs[Q*GSPfp] /. values];
PrintLabeledList[N[{Q,Wp/(2*Pi),gspQpfp} /. values]
,{"Qp","fp (Hz)","GSP"} ];
Print["-----------------"]
PrintLabeledList[Together[{R1,R2,R3,R4,R5,C6*10^9,R7,C8*10^9} /. values]
,{"R1 (ohm)","R2 (ohm)","R3 (ohm)","R4 (ohm)","R5 (ohm)","C6 (nF)"
,"R7 (ohm)","C8 (nF)"}];
```

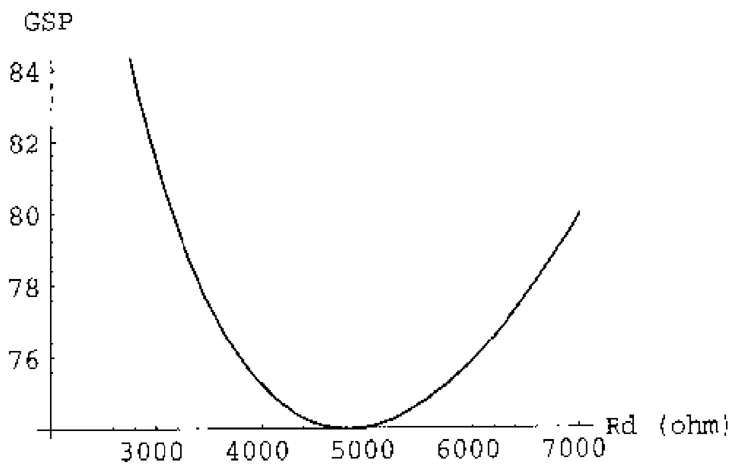
```
Qp = 25.
fp (Hz) = 1500.
GSP = 74.014
-----------------
R1 (ohm) = 4700.
R2 (ohm) = 230378.
R3 (ohm) = 4700.
R4 (ohm) = 4463.56
R5 (ohm) = 4700.
C6 (nF) = 22.
R7 (ohm) = 4700.
C8 (nF) = 22.
```

■ A.6.10 Optimization

■ Find Rd for Low Gain-Sensitivity Product

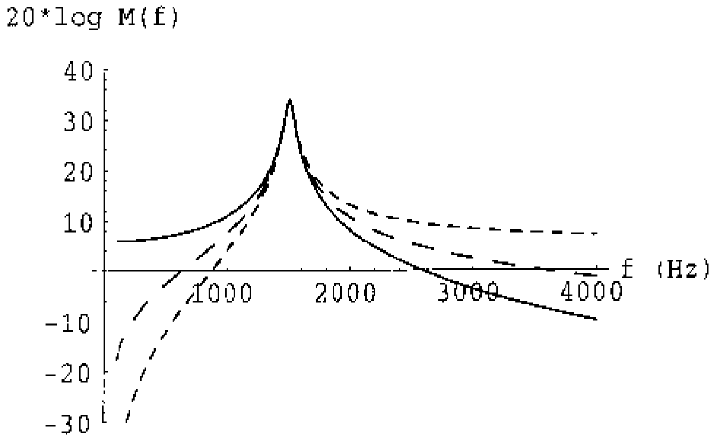
```
values = {Q -> 25., Wp -> 2*Pi*1500.
, Co -> 22.*10^(-9)} //N;
gspQpfp1 = Together[Abs[GSPQp/2.] + Abs[Q*GSPfp] /. values] //Simplify;
```

```
r1 = 2000;
r2 = 7000;
Plot[{gspQpfpl}
, {Rd, r1, r2}
, AxesLabel -> {"Rd (ohm)","GSP"}
];
```




```
{Gspmin,Rset} = FindMinimum[gspQpfpl, {Rd, r1, r2}]

{73.99, {Rd -> 4822.22}}

values = {Q -> 25., Wp -> 2*Pi*1500.
, Co -> 22.*10^(-9)} //N;
gspQpfp = Together[Abs[GSPQp/2]+Abs[Q*GSPfp] /. values /. Rset];
PrintLabeledList[N[{Q,Wp/(2*Pi),gspQpfp} /. values/. Rset]
,{"Qp","fp (Hz)","GSP"} ];
Print["------------------"]
PrintLabeledList[Together[{R1,R2,R3,R4,R5,C6*10^9,R7,C8*10^9} /. values/.Rset]
,{"R1 (ohm)","R2 (ohm)","R3 (ohm)","R4 (ohm)","R5 (ohm)","C6 (nF)"
,"R7 (ohm)","C8 (nF)"}];

Qp = 25.
fp (Hz) = 1500.
GSP = 73.99
------------------
R1 (ohm) = 4822.22
R2 (ohm) = 236289.
R3 (ohm) = 4822.22
R4 (ohm) = 4820.9
R5 (ohm) = 4822.22
C6 (nF) = 22.
R7 (ohm) = 4822.22
C8 (nF) = 22.
```

```
Hlp = Ha /. values /. Rset//Together;
num = Numerator[Hlp];
den = Denominator[Hlp];
numlist = CoefficientList[num,s];
denlist = CoefficientList[den,s];
Hlp = 1/denlist[[3]] * num/(den/denlist[[3]])//Factor

                     8
         1.74124 10
------------------------------
           7                  2
8.88264 10   + 376.991 s + 1. s

Hhp = Hha /. values /. Rset//Together;
num = Numerator[Hhp];
den = Denominator[Hhp];
numlist = CoefficientList[num,s];
denlist = CoefficientList[den,s];
Hhp = 1/denlist[[3]] * num/(den/denlist[[3]])//Factor

                  2
         1.95973 s
------------------------------
           7                  2
8.88264 10   + 376.991 s + 1. s

Hbp = Hba /. values /. Rset//Together;
num = Numerator[Hbp];
den = Denominator[Hbp];
numlist = CoefficientList[num,s];
denlist = CoefficientList[den,s];
Hbp = 1/denlist[[3]] * num/(den/denlist[[3]])//Factor

         -18472.6 s
------------------------------
           7                  2
8.88264 10   + 376.991 s + 1. s

AHlp = 20*Log[10,Abs[Hlp] /. s -> N[I*2*Pi*f]];
AHhp = 20*Log[10,Abs[Hhp] /. s -> N[I*2*Pi*f]];
AHbp = 20*Log[10,Abs[Hbp] /. s -> N[I*2*Pi*f]];
Plot[{AHlp, AHhp, AHbp}
, {f, 100, 4000}
,PlotRange -> {-30,40}
,PlotStyle -> {Dashing[{}],
               Dashing[{.02}],
               Dashing[{.04}]}
, AxesLabel -> {"f (Hz)","20*log M(f)"}
];
```


20*log M(f)
40
30
20
10
-10
-20
-30
1000
2000
3000
4000
f (Hz)

A.7 Advanced Analog Filter Design Case Studies

Miroslav D. Lutovac, Dejan V. Tosic and Brian L. Evans
lutovac@iritel.bg.ac.yu tosic@galeb.etf.bg.ac.yu bevans@ece.utexas.edu

A.7.1 References

1. D.V. Tosic, M.D.Lutovac, B.L.Evans,
"Advanced filter design,"
Proc. IEEE Asilomar Conf. Signal, Systems, Computer,
Nov. 1997, pp.710–715.

2. M. D. Lutovac, D. V. Tosic and B. L. Evans,
"Advanced Filter Design for Signal Processing using MATLAB and Mathematica,"
http://iritel.iritel.bg.ac.yu/~lutovac/www/afdhome.htm
http://galeb.etf.bg.ac.yu/~tosic/afdhome.htm
http://www.ece.utexas.edu/~bevans/

A.7.2 Initialization

```
SetDirectory[HomeDirectory[]];
<<afd\math\m\clearall.m
<<graphics`graphics`
```

A.7.3 Notation

a – selectivity factor
ap – maximum passband loss, dB, of realized filter
Ap – maximum passband loss, dB, in specification
as – minimum stopband loss, dB, of realized filter
As – minimum stopband loss, dB, in specification
A2a(n,Ap,As) – minimum selectivity factor from attenuation spec
A2K(A) – characteristic function in terms of attenuation in dB
e – ripple factor
fp – passband edge [Hz] of realized filter
Fp – passband edge [Hz] in specification
fs – stopband edge [Hz] of realized filter
Fs – stopband edge [Hz] in specification
Ke(n,a,e,x) – characteristic function
L(n,a) – discrimination factor
n – order
nbut(Fp,Fs,Ap,As) minimum Butterworth order from specification
ncheb(Fp,Fs,Ap,As) minimum Chebyshev order from specification
nellip(Fp,Fs,Ap,As) minimum order from specification
q(k) – modular constant
R(n,a,x) – elliptic rational function
S(n,a,e) – list of transfer function poles
S(n,a,e,i) – transfer function pole

SA – attenuation specification
SK – characterictic function specification
X(n,a) – list of zeros of elliptic rational function
X(n,a,i) – zero of elliptic rational function
Z(n,a,e) – zeta function

A.7.4 Definitions and Procedures

```
X[n_Integer,a_,i_Integer] :=  -JacobiCD[
 (2*i-1)*EllipticK[1/a^2]/n, 1/a^2
];
X[n_Integer,a_,i_Integer] := 0 /; And[i==(n+1)/2,OddQ[n]];
X[n_Integer, a_] :=  X[n,a,#]& /@ Range[n];
L[n_Integer,a_] := Block[{i,r},
 If[EvenQ[n],
  r = (1/a^n)*Product[(a^2 - X[n,a,i]^2)^2, {i,n/2}]/
             Product[(1   - X[n,a,i]^2)^2, {i,n/2}];,
  r = (1/a^(n-2))*Product[(a^2 - X[n,a,i]^2)^2, {i,(n-1)/2}]/
                 Product[(1   - X[n,a,i]^2)^2, {i,(n-1)/2}];
 ];
 r
];
R[n_Integer,a_,x_] := Block[{i,r,r0},
 If[EvenQ[n],
  r = Product[x^2 -      X[n,a,i]^2, {i,n/2}]/
      Product[x^2 - a^2/X[n,a,i]^2, {i,n/2}];
  r0 = Product[1 -      X[n,a,i]^2, {i,n/2}]/
       Product[1 - a^2/X[n,a,i]^2, {i,n/2}];,
  r = x*Product[x^2 -      X[n,a,i]^2, {i,(n-1)/2}]/
        Product[x^2 - a^2/X[n,a,i]^2, {i,(n-1)/2}];
  r0 = Product[1 -      X[n,a,i]^2, {i,(n-1)/2}]/
       Product[1 - a^2/X[n,a,i]^2, {i,(n-1)/2}];
 ];
 r/r0
];
Z[n_Integer,a_,e_] :=  JacobiSN[
 InverseJacobiSN[1/Sqrt[1+e^2],1-1/(L[n,a])^2]*
 EllipticK[1-1/a^2]/EllipticK[1-1/(L[n,a])^2],1-1/a^2
];
S[n_Integer, a_, e_, i_Integer] := Block[
{den,num,numim,numre,x,z},
 x = X[n,a,i];
 z = Z[n,a,e];
 numre = -z*Sqrt[1 - z^2]*Sqrt[1 - x^2]*Sqrt[1 - x^2/a^2];
 numim = x*Sqrt[1 - (1-1/a^2)*z^2];
 num = numre + I*numim;
```

```
  den = 1 - (1 - x^2/a^2)*z^2;
  num/den
 ];
 S[n_Integer, a_, e_] := S[n,a,e,#]& /@ Range[n];
 A2K[A_] := Sqrt[1 - 10^(-A/10)]/10^(-A/20);
 Ke[n_Integer, a_, e_, x_] := e*Abs[R[n,a,x]];
```

■ A.7.5 Specification

```
SA = {3000., 3225., 0.2, 40.};
{Fp, Fs, Ap, As} = SA;
Kp = A2K[Ap];
Ks = A2K[As];
SK = {Fp, Fs, Kp, Ks}

{3000., 3225., 0.217091, 99.995}
```

■ A.7.6 Minimum Order

```
nellip[Fp_,Fs_,Ap_,As_] := Block[
 {num, den,
  k = Fp/Fs,
  L = Sqrt[(-1 + 10^(As/10))/(-1 + 10^(Ap/10))]},
 num = EllipticK[1-1/L^2]/EllipticK[1/L^2];
 den = EllipticK[1-k^2]/EllipticK[k^2];
 Ceiling[num/den//N]
];
ncheb[Fp_,Fs_,Ap_,As_] := Block[
 {L = Sqrt[(-1 + 10^(As/10))/(-1 + 10^(Ap/10))],
  aspec = Fs/Fp},
 Ceiling[ArcCosh[L]/ArcCosh[aspec]//N]
];
nbut[Fp_,Fs_,Ap_,As_] := Block[
 {L = Sqrt[(-1 + 10^(As/10))/(-1 + 10^(Ap/10))],
  aspec = Fs/Fp},
 Ceiling[Log[10,L]/Log[10,aspec]//N]
];
{nellip[Fp,Fs,Ap,As], ncheb[Fp,Fs,Ap,As], nbut[Fp,Fs,Ap,As]}
nmin = nellip[Fp,Fs,Ap,As];
nmax = 2*nmin;
nlist = Range[nmin,nmax]

{8, 18, 85}
{8, 9, 10, 11, 12, 13, 14, 15, 16}
```

A.7.7 Range of Selectivity Factor and Ripple Factor

```
q[k_] := Block[{c,e,r,s,t},
 If[k<=1/Sqrt[2.0],
   t = (1/2)*(1 - (1-k^2)^(1/4))/(1 + (1-k^2)^(1/4));,
   t = (1/2)*(1 - Sqrt[k])/(1 + Sqrt[k]);
 ];
 e = {1,5, 9, 13,  17,   21,    25,     29,      33,       37};
 c = {1,2,15,150,1707,20910,268616,3567400,48555069,673458874};
 s = Sum[c[[i]]*(t^e[[i]]),{i,Length[e]}];
 If[k<=1/Sqrt[2.0],
   r = s;,
   r = Exp[Pi^2/Log[s]];
 ];
 N[r]
];
A2a[n_,Ap_,As_] := Block[
 {m, num, den, terms=9, L, qL},
 L = Sqrt[(-1 + 10^(As/10))/(-1 + 10^(Ap/10))];
 qL = q[1/L]^(1/n);
 num = 1 + 2*Sum[(-1)^m*(qL)^(m^2), {m,1,terms}];
 den = 1 + 2*Sum[(qL)^(m^2), {m,1,terms}];
 1/Sqrt[1 - (num/den)^4]
];
amin8  = A2a[nmin,Ap,As];
amax8  = a/. FindRoot[R[nmin,a,Fs/Fp]==Ks/Kp, {a,Fs/Fp,1.1}];
amin9  = A2a[9,Ap,As];
amax9  = a/. FindRoot[R[9,a,Fs/Fp]==Ks/Kp, {a,amax8,1.1}];
amin10 = A2a[10,Ap,As];
amax10 = a/. FindRoot[R[10,a,Fs/Fp]==Ks/Kp, {a,amax9,1.2}];
amin11 = A2a[11,Ap,As];
amax11 = a/. FindRoot[R[11,a,Fs/Fp]==Ks/Kp, {a,amax10,1.2}];
amin12 = A2a[12,Ap,As];
amax12 = a/. FindRoot[R[12,a,Fs/Fp]==Ks/Kp, {a,amax11,1.2}];
amin13 = A2a[13,Ap,As];
amax13 = a/. FindRoot[R[13,a,Fs/Fp]==Ks/Kp, {a,amax12,1.3}];
amin14 = A2a[14,Ap,As];
amax14 = a/. FindRoot[R[14,a,Fs/Fp]==Ks/Kp, {a,amax13,1.4}];
amin15 = A2a[15,Ap,As];
amax15 = a/. FindRoot[R[15,a,Fs/Fp]==Ks/Kp, {a,amax14,1.6}];
amin16 = A2a[16,Ap,As];
amax16 = a/. FindRoot[R[16,a,Fs/Fp]==Ks/Kp, {a,amax15,1.9}];
aminlist = {amin8,amin9,amin10,amin11,amin12,amin13,amin14,amin15,amin16};
amaxlist = {amax8,amax9,amax10,amax11,amax12,amax13,amax14,amax15,amax16};
eminlist = Table[Ks/L[n,amaxlist[[n-8+1]]], {n,nmin,nmax}];
{emin8,emin9,emin10,emin11,emin12,emin13,emin14,emin15,emin16} = eminlist;
emax = Kp;
emaxlist = Table[Kp,{nmax-nmin+1}];
```

```
TableForm[Transpose[{nlist,aminlist,amaxlist,eminlist,emaxlist}]
, TableHeadings->{{},{"n","amin","amax","emin","emax"}}]
```

n	amin	amax	emin	emax
8	1.04285	1.08323	0.0757872	0.217091
9	1.022	1.09807	0.0184689	0.217091
10	1.01135	1.12013	0.00368669	0.217091
11	1.00587	1.15172	0.000578651	0.217091
12	1.00304	1.19663	0.0000676012	0.217091
13	1.00158	1.26158	5.39161 10^{-6}	0.217091
14	1.00082	1.35951	2.53319 10^{-7}	0.217091
15	1.00042	1.51914	5.28479 10^{-9}	0.217091
16	1.00022	1.82219	2.51893 10^{-11}	0.217091

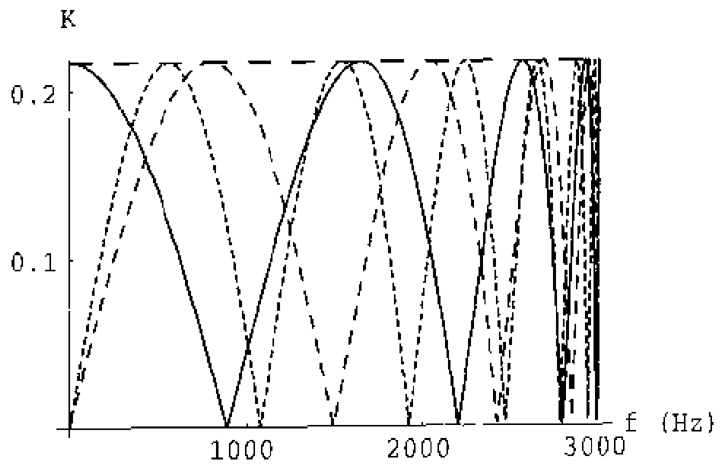
■ A.7.8 Range of Edge Frequencies

```
fpminlist = Fs / amaxlist;
{fpmin8,fpmin9,fpmin10,fpmin11,fpmin12,fpmin13,
 fpmin14,fpmin15,fpmin16} = fpminlist;
fpmaxlist = Fs / aminlist;
{fpmax8,fpmax9,fpmax10,fpmax11,fpmax12,fpmax13,
 fpmax14,fpmax15,fpmax16} = fpmaxlist;
fsminlist = Fp * aminlist;
{fsmin8,fsmin9,fsmin10,fsmin11,fsmin12,fsmin13,
 fsmin14,fsmin15,fsmin16} = fsminlist;
fsmaxlist = Fp * amaxlist;
{fsmax8,fsmax9,fsmax10,fsmax11,fsmax12,fsmax13,
 fsmax14,fsmax15,fsmax16} = fsmaxlist;
TableForm[Transpose[{nlist,fpminlist,fpmaxlist,fsminlist,fsmaxlist}]
, TableHeadings->{{},{"n","fpmin","fpmax","fsmin","fsmax (Hz)"}}]
```

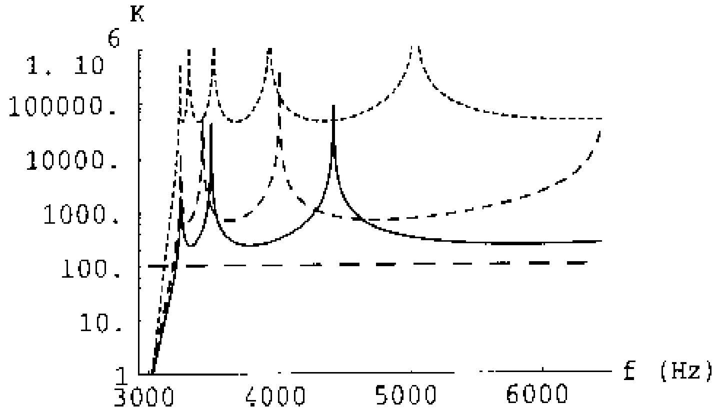
n	fpmin	fpmax	fsmin	fsmax (Hz)
8	2977.2	3092.49	3128.55	3249.7
9	2936.97	3155.57	3066.01	3294.21
10	2879.14	3188.79	3034.06	3360.38
11	2800.16	3206.17	3017.62	3455.16
12	2695.06	3215.22	3009.13	3589.89
13	2556.32	3219.92	3004.73	3784.74
14	2372.18	3222.36	3002.45	4078.53
15	2122.91	3223.63	3001.27	4557.42
16	1769.85	3224.29	3000.66	5466.56

A.7.9 Design D1

```
Plot[Evaluate[{Kp
              , Ke[ 8, Fs/Fp, emax, f/Fp]
              , Ke[ 9, Fs/Fp, emax, f/Fp]
              , Ke[13, Fs/Fp, emax, f/Fp]
             }]
, {f,0,Fp}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{1000, 2000, 3000}, {0, 0.1, 0.2}}
, PlotRange -> All];
```

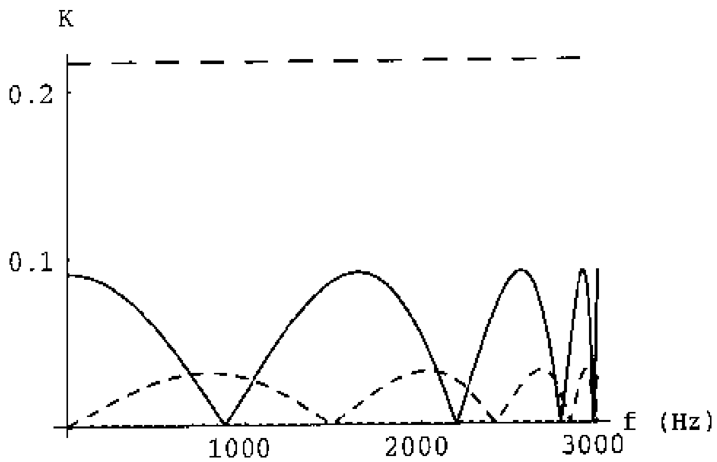



```
LogPlot[Evaluate[{Ks
              , Ke[ 8, Fs/Fp, emax, f/Fp]
              , Ke[ 9, Fs/Fp, emax, f/Fp]
              , Ke[13, Fs/Fp, emax, f/Fp]
             }]
, {f,Fp,2*Fs}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 4000, 5000, 6000}, Automatic}
, PlotRange -> {1,10^6}];
```



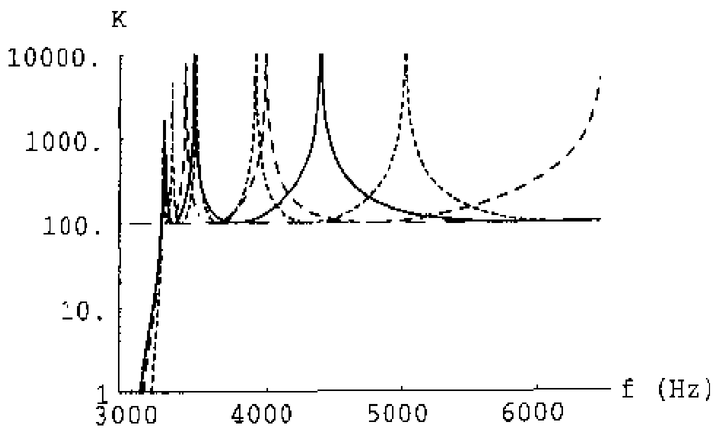
■ A.7.10 Design D2

```
Plot[Evaluate[{Kp
                , Ke[ 8, Fs/Fp, Ks/L[ 8,Fs/Fp], f/Fp]
                , Ke[ 9, Fs/Fp, Ks/L[ 9,Fs/Fp], f/Fp]
                , Ke[13, Fs/Fp, Ks/L[13,Fs/Fp], f/Fp]
               }]
, {f,0,Fp}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{1000, 2000, 3000}, {0, 0.1, 0.2}}
, PlotRange -> All];
```



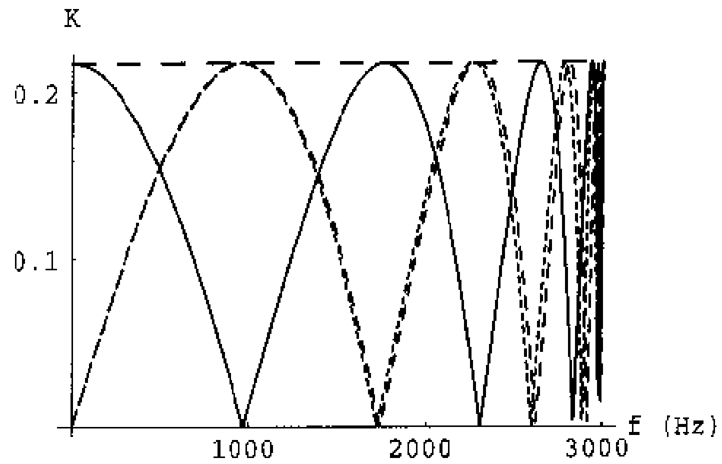

```
LogPlot[Evaluate[{Ks
                , Ke[ 8, Fs/Fp, Ks/L[ 8,Fs/Fp], f/Fp]
                , Ke[ 9, Fs/Fp, Ks/L[ 9,Fs/Fp], f/Fp]
                , Ke[13, Fs/Fp, Ks/L[13,Fs/Fp], f/Fp]
               }]
, {f,Fp,2*Fs}
```

```
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 4000, 5000, 6000}, Automatic}
, PlotRange -> {1,10^4}];
```

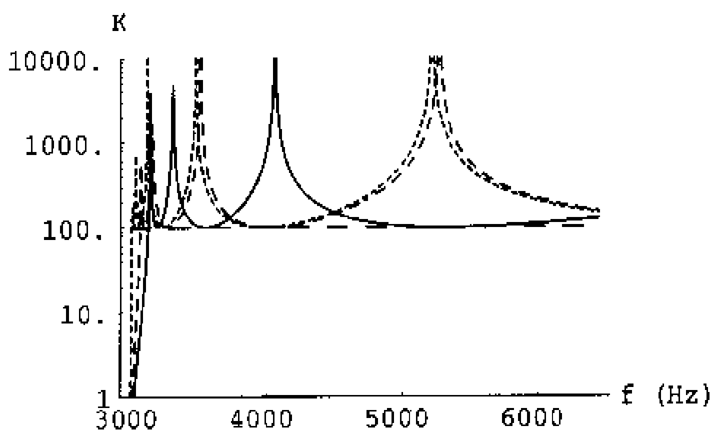


■ A.7.11 Design D3a

```
Plot[Evaluate[{Kp
               , Ke[ 8, amin8,  emax, f/Fp]
               , Ke[ 9, amin9,  emax, f/Fp]
               , Ke[13, amin13, emax, f/Fp]
              }]
, {f,0,Fp}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{1000, 2000, 3000}, {0, 0.1, 0.2}}
, PlotRange -> All];
```

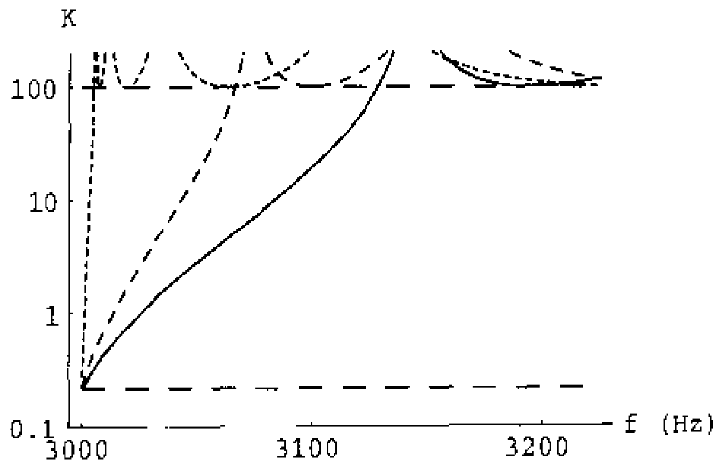



```
LogPlot[Evaluate[{Ks
                , Ke[ 8, amin8,  emax, f/Fp]
                , Ke[ 9, amin9,  emax, f/Fp]
                , Ke[13, amin13, emax, f/Fp]
               }]
, {f,Fp,2*Fs}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 4000, 5000, 6000}, Automatic}
, PlotRange -> {1,10^4}];
```



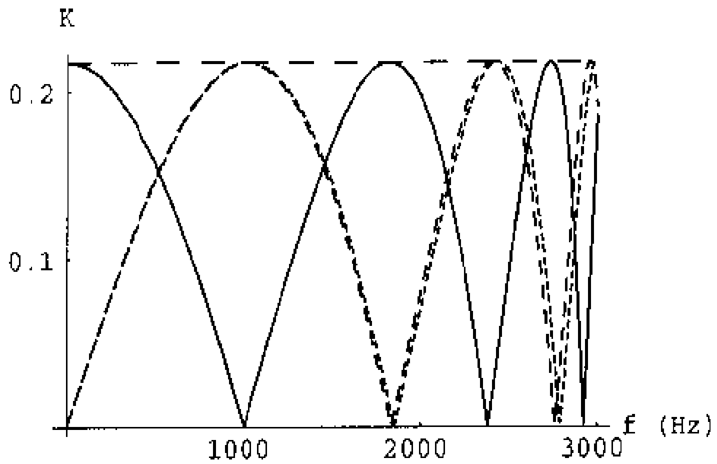

```
LogPlot[Evaluate[{Kp,Ks
                , Ke[ 8, amin8,  emax, f/Fp]
                , Ke[ 9, amin9,  emax, f/Fp]
                , Ke[13, amin13, emax, f/Fp]
               }]
, {f,Fp,Fs}
, AxesLabel -> {"f (Hz)", "K"}
```

```
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 3100, 3200}, {0.1, 1, 10, 100}}
, PlotRange -> {0.1,200}];
```

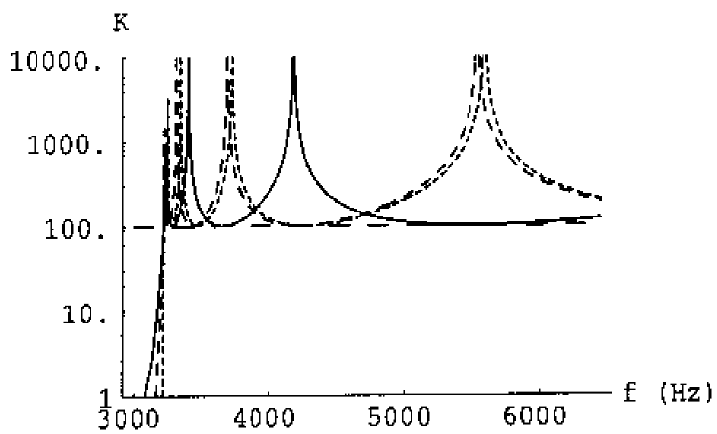


■ A.7.12 Design D3b

```
Plot[Evaluate[{Kp
              , Ke[ 8, amin8,  emax, f/fpmax8]
              , Ke[ 9, amin9,  emax, f/fpmax9]
              , Ke[13, amin13, emax, f/fpmax13]
             }]
, {f,0,Fp}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{1000, 2000, 3000}, {0, 0.1, 0.2}}
, PlotRange -> All];
```

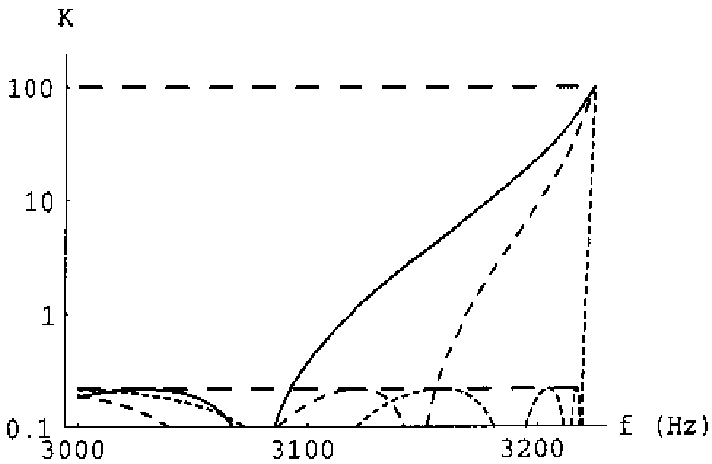



```
LogPlot[Evaluate[{Ks
                , Ke[ 8, amin8,  emax, f/fpmax8]
                , Ke[ 9, amin9,  emax, f/fpmax9]
                , Ke[13, amin13, emax, f/fpmax13]
               }]
, {f,Fp,2*Fs}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 4000, 5000, 6000}, Automatic}
, PlotRange -> {1,10^4}];
```



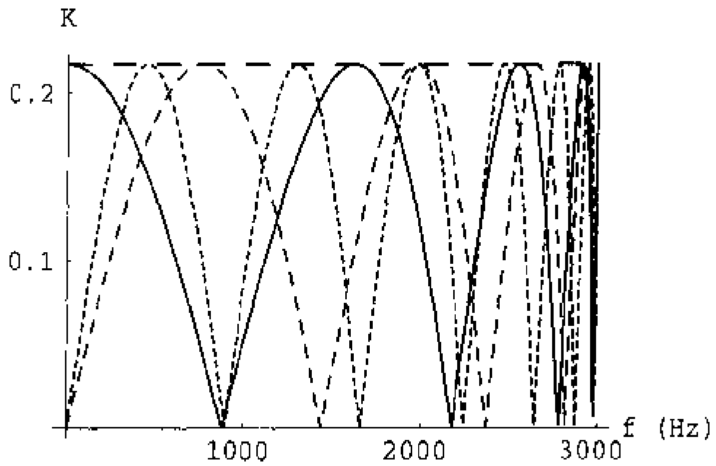

```
LogPlot[Evaluate[{Kp,Ks
                , Ke[ 8, amin8,  emax, f/fpmax8]
                , Ke[ 9, amin9,  emax, f/fpmax9]
                , Ke[13, amin13, emax, f/fpmax13]
               }]
, {f,Fp,Fs}
, AxesLabel -> {"f (Hz)", "K"}
```

```
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 3100, 3200}, {0.1, 1, 10, 100}}
, PlotRange -> {0.1,200}];
```

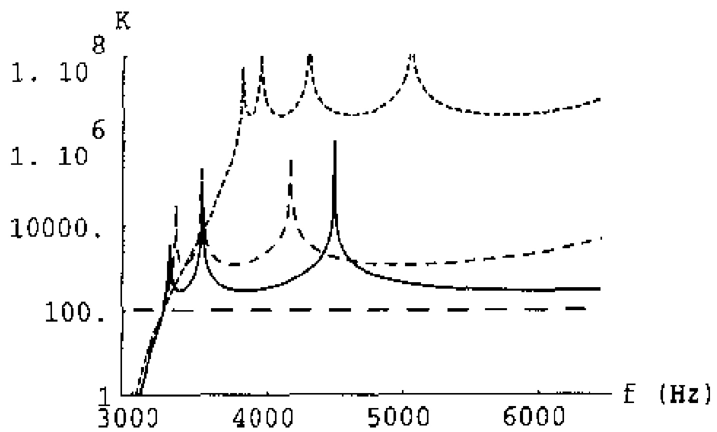


■ A.7.13 Design D4a

```
Plot[Evaluate[{Kp
              , Ke[ 8, amax8,  emax, f/Fp]
              , Ke[ 9, amax9,  emax, f/Fp]
              , Ke[13, amax13, emax, f/Fp]
             }]
, {f,0,Fp}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{1000, 2000, 3000}, {0, 0.1, 0.2}}
, PlotRange -> All];
```



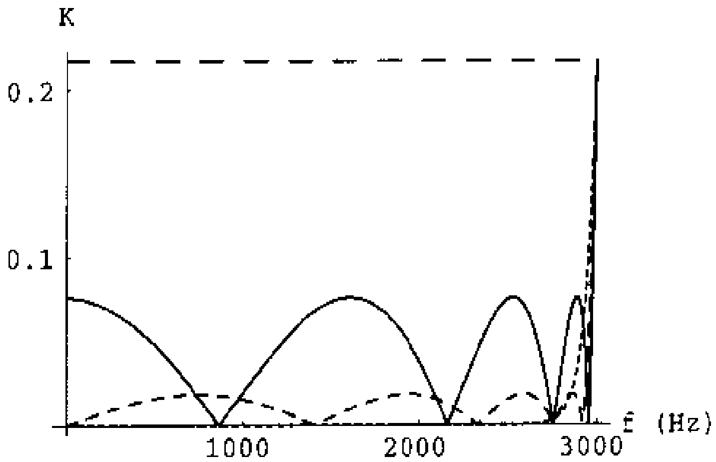

```
LogPlot[Evaluate[{Ks
                , Ke[ 8, amax8,  emax, f/Fp]
                , Ke[ 9, amax9,  emax, f/Fp]
                , Ke[13, amax13, emax, f/Fp]
               }]
, {f,Fp,2*Fs}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 4000, 5000, 6000}, Automatic}
, PlotRange -> {1,10^8}];
```



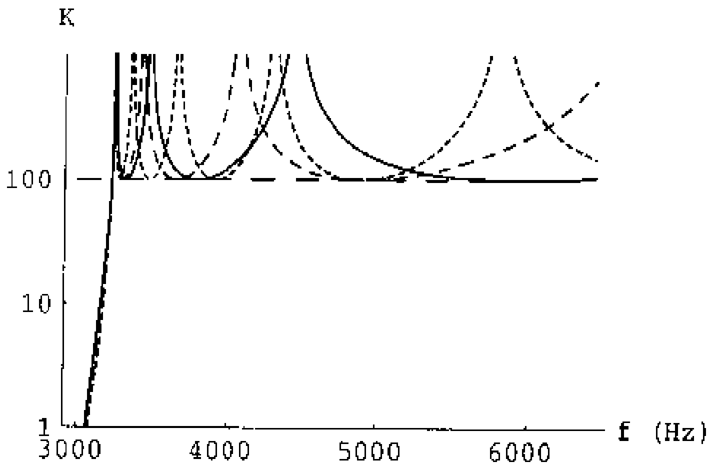
■ A.7.14 Design D4b

```
Plot[Evaluate[{Kp
              , Ke[ 8, amax8,  emin8,  f/(Fs/amax8)]
              , Ke[ 9, amax9,  emin9,  f/(Fs/amax9)]
              , Ke[13, amax13, emin13, f/(Fs/amax13)]
             }]
, {f,0,Fp}
```

```
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{1000, 2000, 3000}, {0, 0.1, 0.2}}
, PlotRange -> All];
```




```
LogPlot[Evaluate[{Ks
                , Ke[ 8, amax8,  emin8,  f/(Fs/amax8)]
                , Ke[ 9, amax9,  emin9,  f/(Fs/amax9)]
                , Ke[13, amax13, emin13, f/(Fs/amax13)]
               }]
, {f,Fp,2*Fs}
, AxesLabel -> {"f (Hz)", "K"}
, PlotStyle -> {Dashing[{0.04}],
                Dashing[{}],
                Dashing[{0.02}],
                Dashing[{0.01}]
               }
, Ticks -> {{3000, 4000, 5000, 6000}, {1,10,100}}
, PlotRange -> {1,10^3}];
```



K
100
10
1
3000
4000
5000
6000
f (Hz)