(2) (a) I decided to use the following design flow:

Synthesize w/ Matlab in z-domain

$$F_{pass} = 2 \cdot \frac{1MHz}{10MHz} = 0.2$$

$$F_{stop} = 2 \cdot \frac{1.5MHz}{10MHz} = 0.3$$

$\Rightarrow$ ellipord(...) = $\boxed{6}$ (filter order)

↓

Break up into Biquads using "tf2sos"

→ estimate Biquad Qs, use inverse bilin. transform

$$s_p = \frac{2}{T}\left(\frac{z_p - 1}{z_p + 1}\right)$$

$$\Rightarrow \left.\begin{array}{l} Q_1 = 0.61 \\ Q_2 = 0.16 \end{array}\right\} \text{"Low Q"}$$

$Q_3 = 6.3$ "Hi Q" (>3)

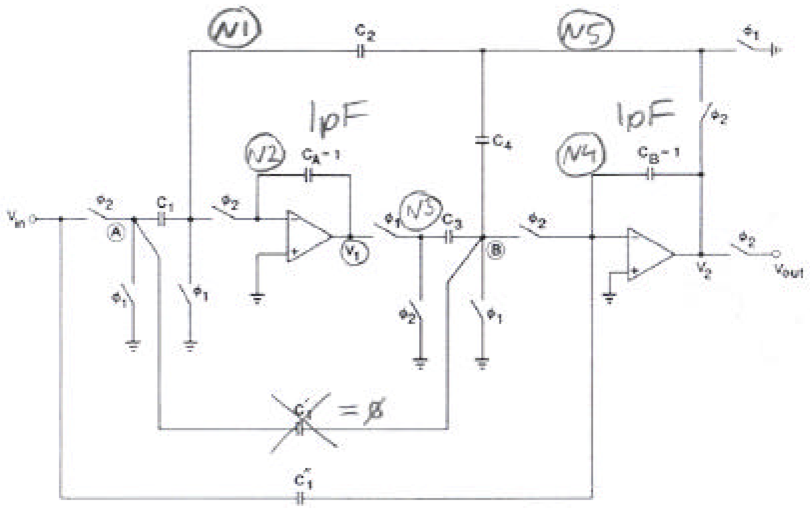
↓

Implement using Hi- and Lo Q circuits as given in Gregorian & Temes pp. 280.

→ see attached schematics with design equations ut

zation:

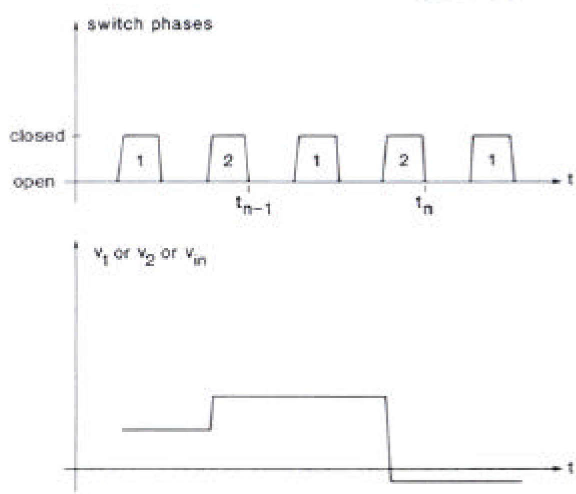
.10a.
le its
y an
on is
e the
input
rents
smit-
ed by
$V_{in} =$
value



(c)

LoQ

$$H(z) = \frac{a_2 z^2 + a_1 z + a_0}{b_2 z^2 + b_1 z + b_0}$$



(d)

FIGURE 5.10. continued.

$$C_1'' = a_0$$

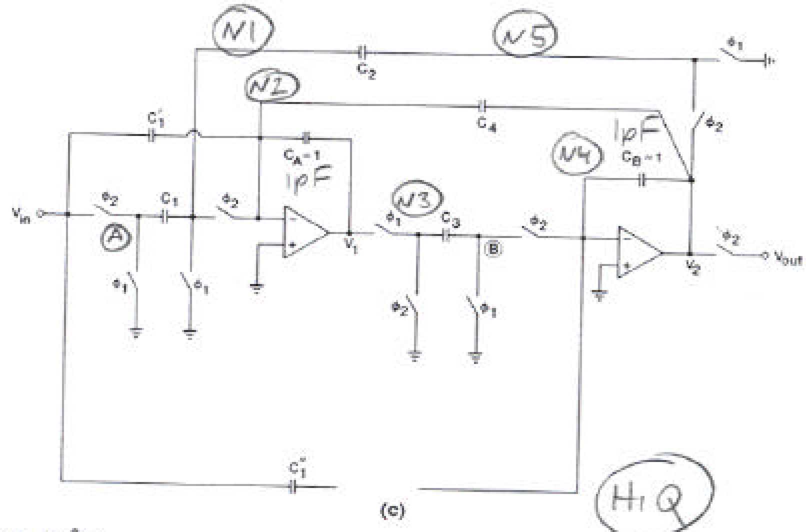
$$C_1' = a_2 - a_0$$

$$C_1 = \frac{a_0 + a_1 + a_2}{C_3}$$

$$C_4 = b_2 - 1$$

$$C_2 C_3 = b_1 + b_2 + 1$$

(Gregorian, Temes)



(c)

$$H(z) = \frac{a_2 z^2 + a_1 z + a_0}{b_2 z^2 + b_1 z + b_0}$$

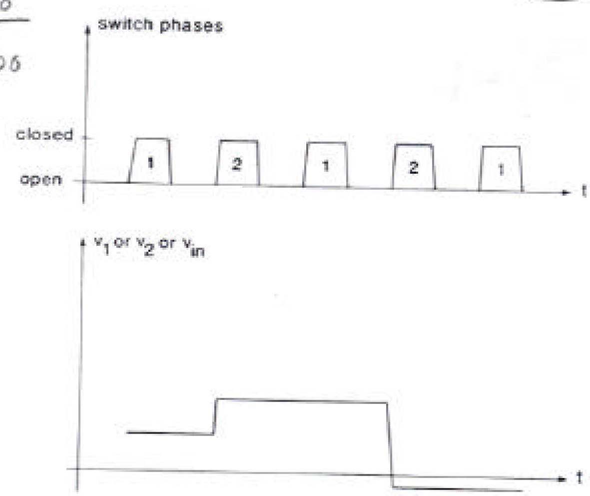
$$C_1'' = \frac{a_2}{b_2}$$

$$C_1' = \frac{a_2 - a_0}{b_2 C_3}$$

$$C_1 = \frac{a_0 + a_1 + a_2}{b_2 C_3}$$

$$C_4 = \frac{1 - 1/b_2}{C_3}$$

$$C_2 C_3 = \frac{1 + b_1 + b_2}{b_2}$$



(d)

FIGURE 5.13. continued.

(Gregorian, Temes)

2) b) Poles and Zeros

Biquad 1:

```
Z1 =
   -4.959483412658553e-001 +8.683520270003670e-001i
   -4.959483412658553e-001 -8.683520270003670e-001i
P1 =
    7.127856685302593e-001 +1.653876720637911e-001i
    7.127856685302593e-001 -1.653876720637911e-001i
```

Biquad 2:

```
Z2 =
    3.825418155817085e-001 +9.239381793883470e-001i
    3.825418155817085e-001 -9.239381793883470e-001i
P2 =
    7.246789558726074e-001 +4.320114746276476e-001i
    7.246789558726074e-001 -4.320114746276476e-001i
```

Biquad 3:

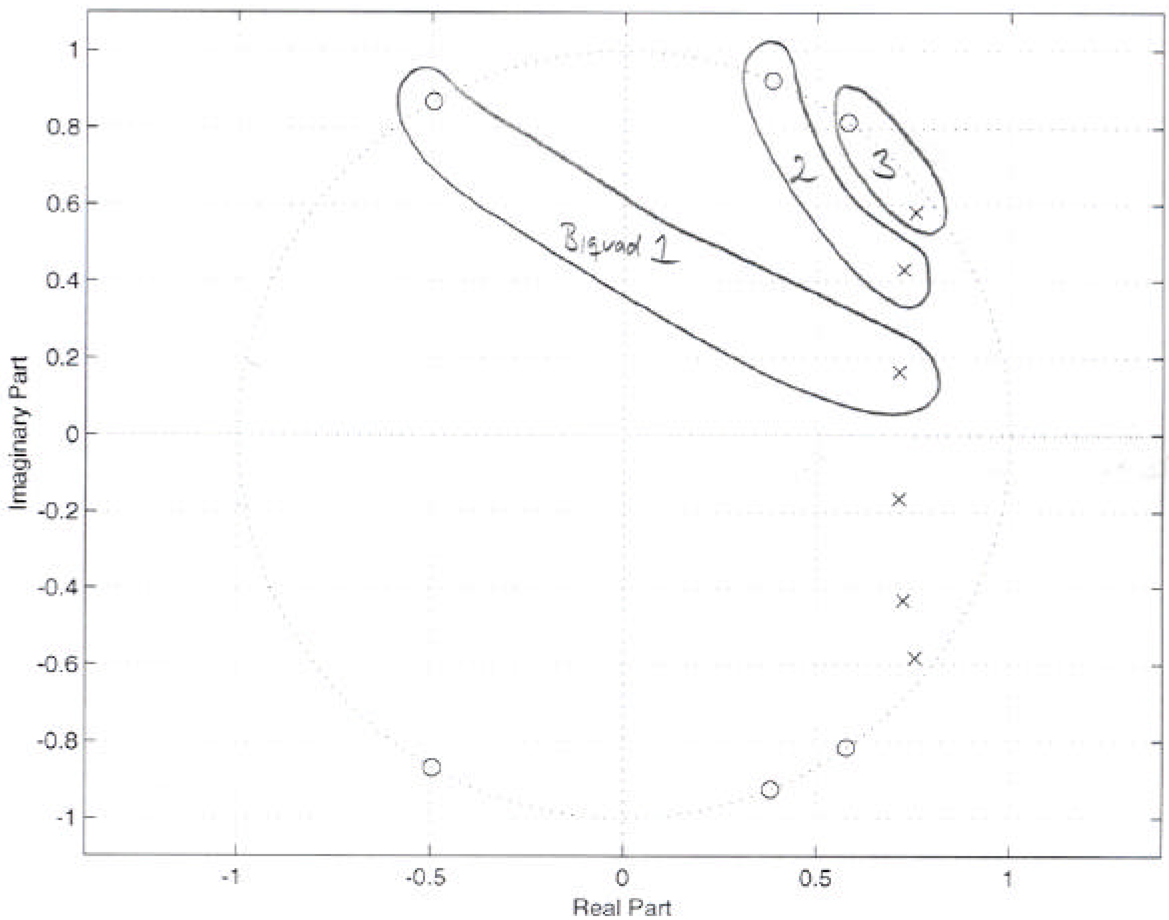
```
Z3 =
    5.785179590177820e-001 +8.156696458088302e-001i
    5.785179590177820e-001 -8.156696458088302e-001i
P3 =
    7.553705109318878e-001 +5.808504547226241e-001i
    7.553705109318878e-001 -5.808504547226241e-001i
```

(Grouping done by Matlab "tf2sos"
→ see next page)

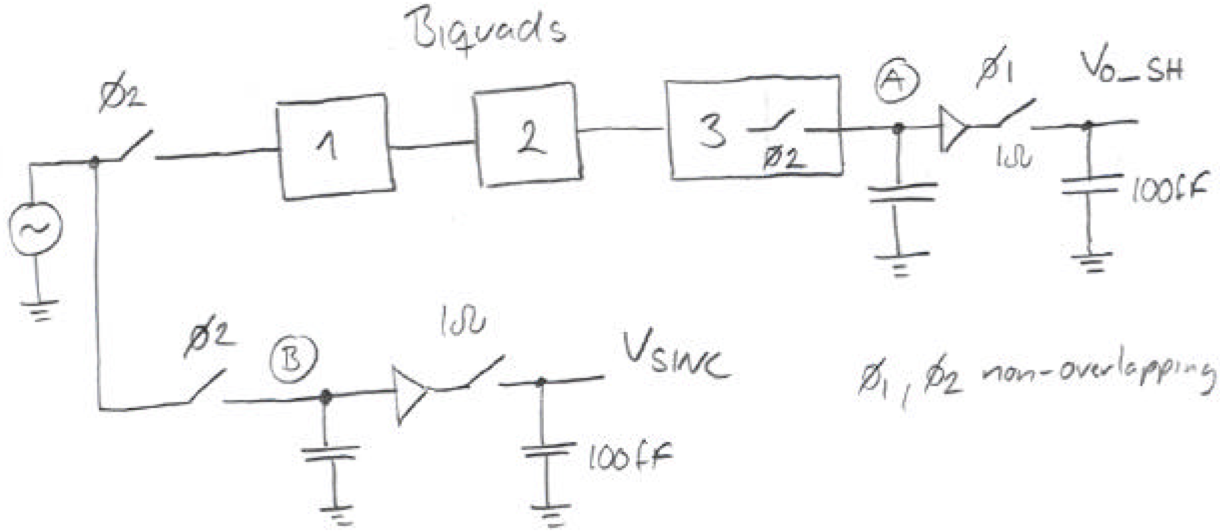
② b)

p/z grouping by Matlab:

("Minimum sensitivity to coefficient tolerances, see e.g. Oppenheim & Schafer)



(2) (c) Top Schematic:



→ (A), (B) are "track and hold nodes", the held value is transferred with ϕ_1 to form an "ideal" staircase voltage ≡ Dirac sampling with falling edge of ϕ_2

→ V_{SINC} will be used in the post processor to eliminate sinc distortion.

$$V_{O\_SH}[dB] - V_{SINC}[dB] \Rightarrow \text{actual discrete time response}$$

→ Biquads scaled in Matlab using:

$$\frac{\sum a}{\sum b} \overset{!}{=} 1 \quad \text{in} \quad \frac{a_2 z^2 + a_1 z + a_0}{b_2 z^2 + b_1 z + b_0} \qquad z \to 1 \text{ @ DC}$$

(see Matlab script)

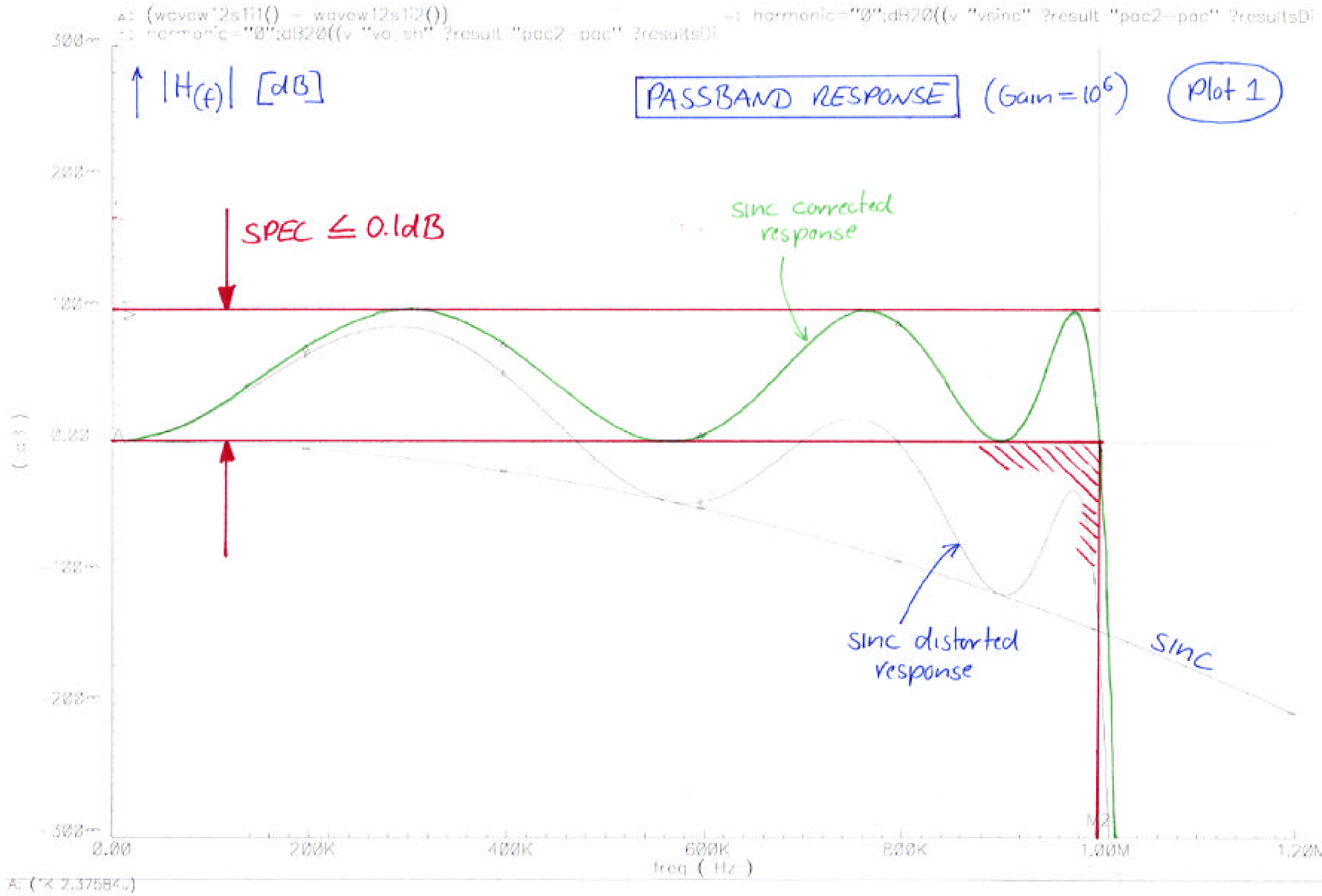
→ Good agreement w/ Spectre simulation, see Plots 1, 2

(2) d) found: $A_{min} \cong \underline{\underline{1200}}$ for $R_p = 0.12\,dB$, assuming f_{corner} is allowed to shift

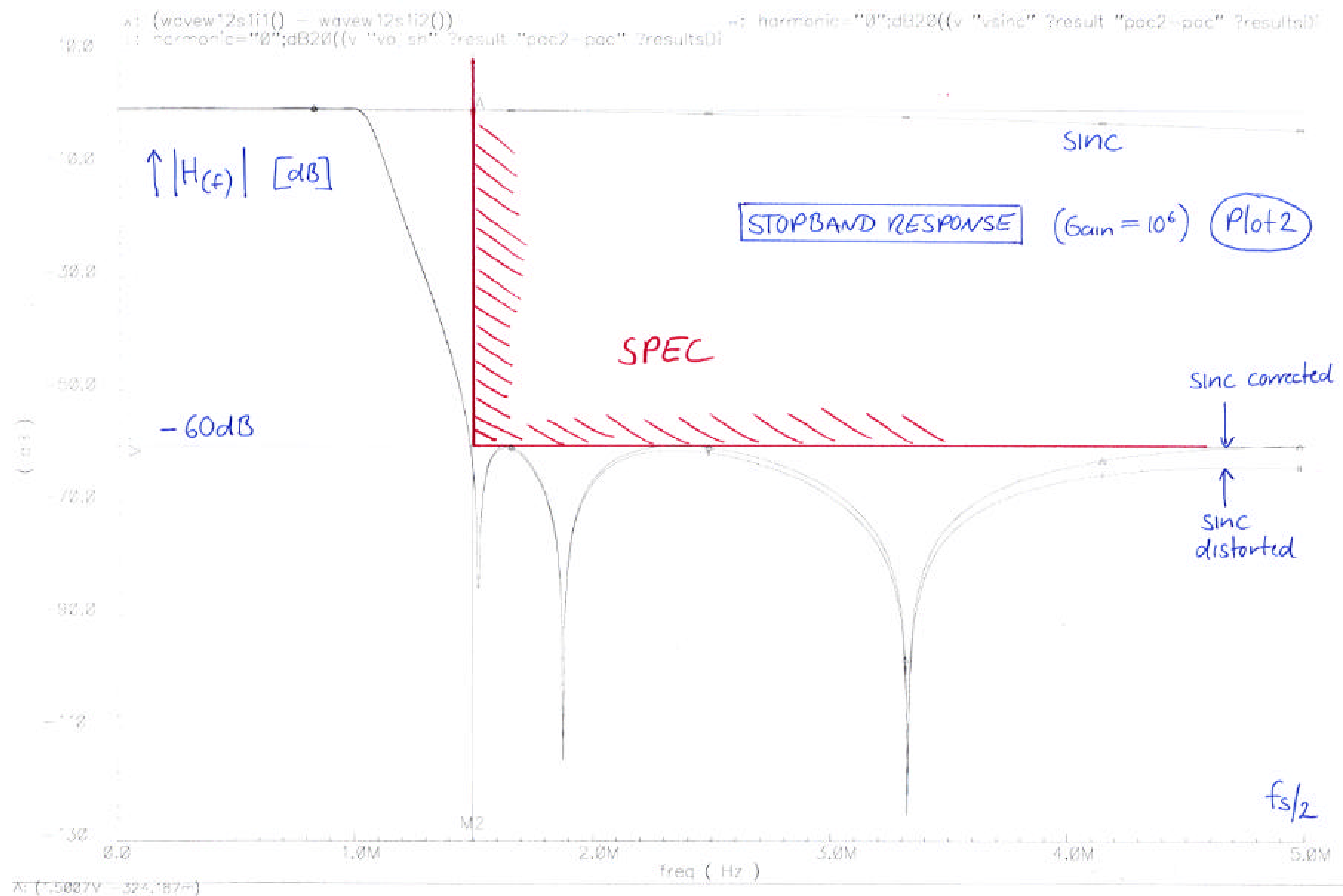
$$\cong \frac{8\,kHz}{1\,MHz} = 0.8\,\%$$

→ Stopband attenuation appears to be insensitive to low gain / gain variations.

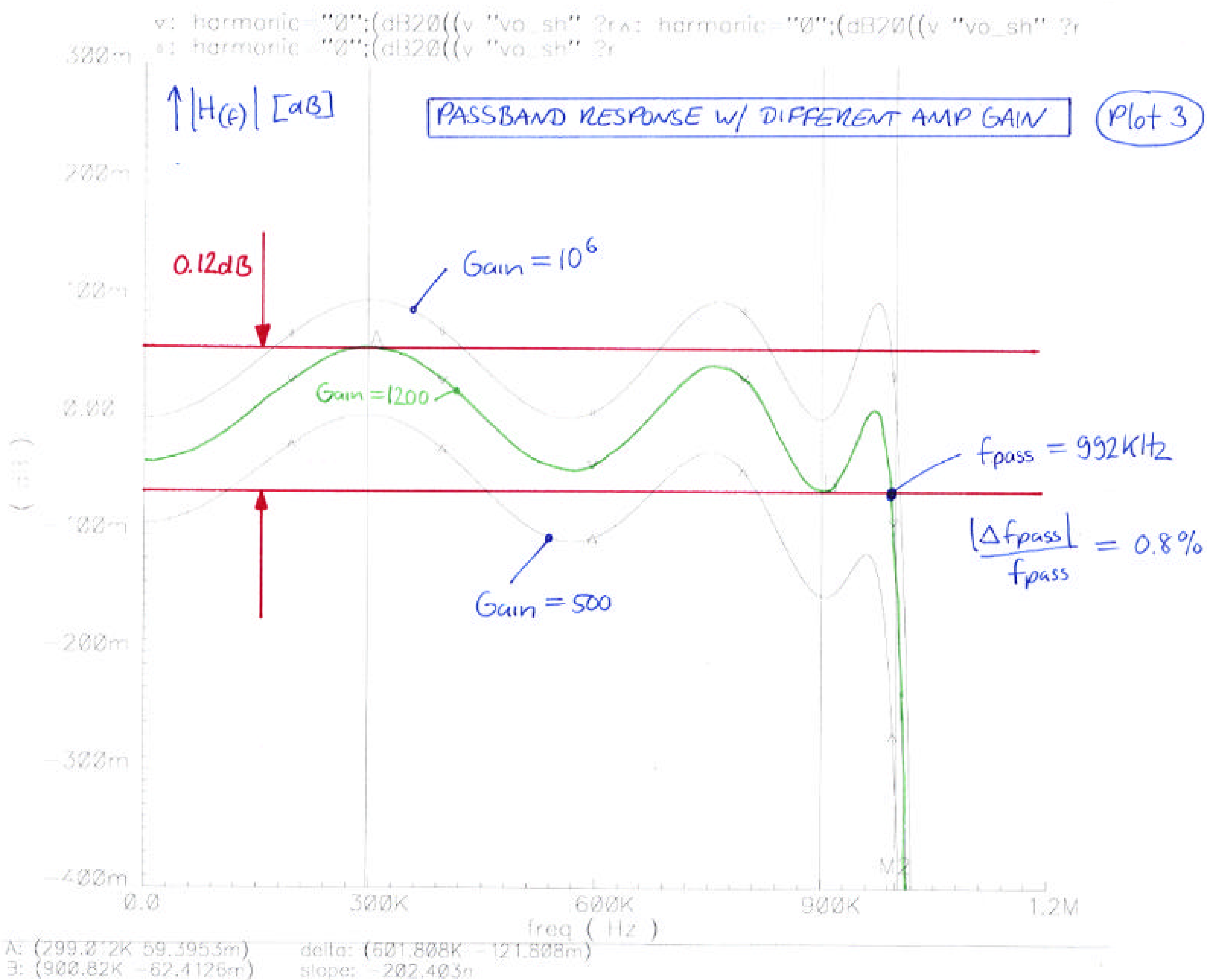
→ See plots 3, 4



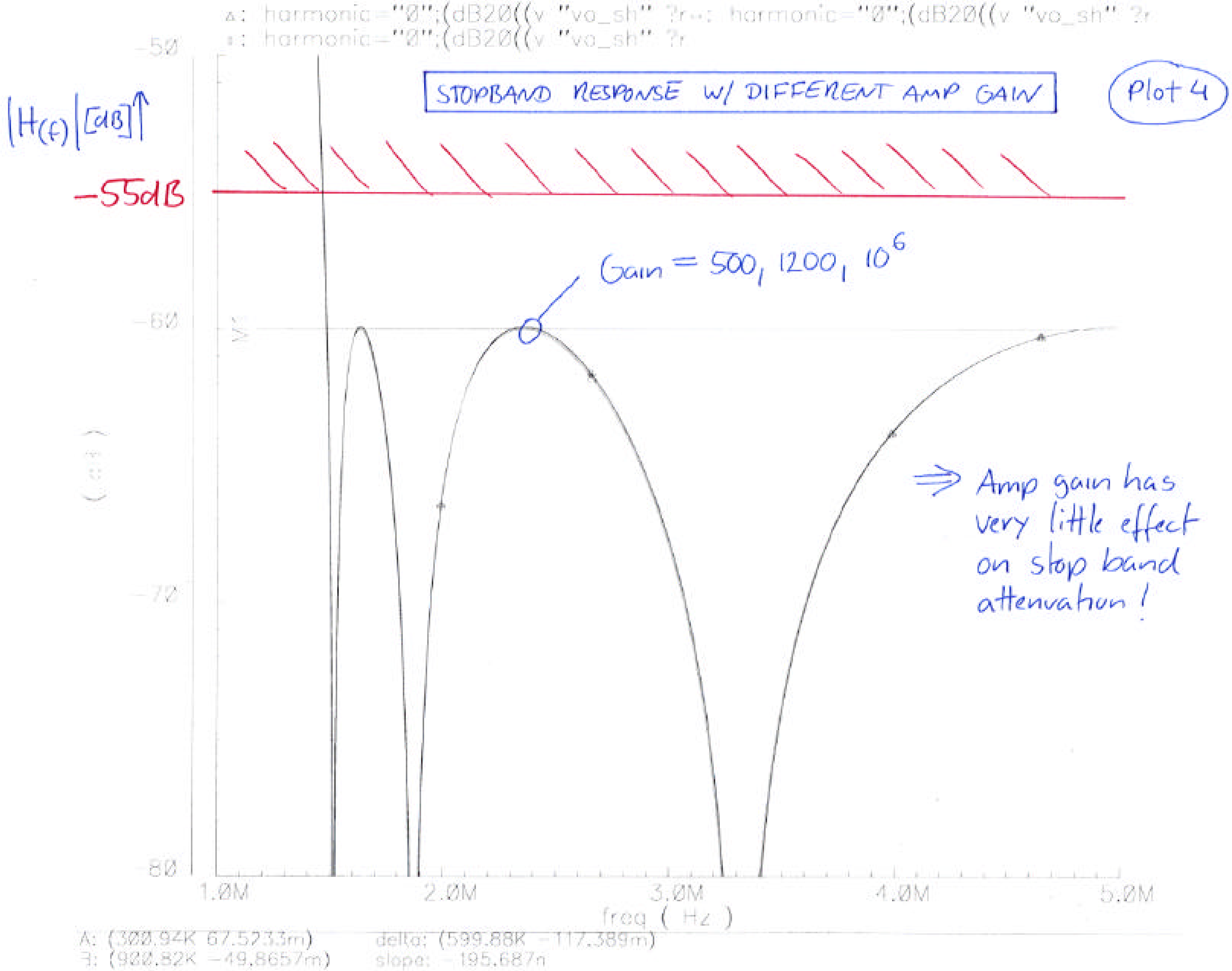
A: (wavew12s1i1() - wavew12s1i2())
harmonic="0";dB20((v "vsinc" ?result "pac2-pac" ?resultsDi
harmonic="0";dB20((v "vo_sh" ?result "pac2-pac" ?resultsDi
↑ |H(f)| [dB]
PASSBAND RESPONSE
(Gain = 10^6)
Plot 1
SPEC ≤ 0.1dB
sinc corrected response
sinc distorted response
sinc
300m
200m
100m
0.00
-100m
-200m
-300m
(dB)
0.00
200K
400K
600K
800K
1.00M
1.20M
freq (Hz)
A: (-X 2.37584)



(wavew12s1i1() - wavew12s1i2())
harmonic="0";dB20((v "vout_sn" ?result "pac2-pac" ?results()
harmonic="0";dB20((v "vsinc" ?result "pac2-pac" ?results()
↑|H(f)| [dB]
sinc
STOPBAND RESPONSE
(Gain = 10^6)
Plot2
SPEC
-60dB
sinc corrected
sinc distorted
fs/2
M2
0.0
1.0M
2.0M
3.0M
4.0M
5.0M
freq (Hz)



v: harmonic="0";(dB20((v "vo_sh" ?r ∧: harmonic="0";(dB20((v "vo_sh" ?r
◦: harmonic="0";(dB20((v "vo_sh" ?r
↑ |H(f)| [dB]
PASSBAND RESPONSE W/ DIFFERENT AMP GAIN
Plot 3
0.12dB
Gain = 10^6
Gain = 1200
Gain = 500
fpass = 992KHz
|Δfpass| / fpass = 0.8%
300m
200m
100m
0.00
-100m
-200m
-300m
-400m
(dB)
0.0
300K
600K
900K
1.2M
freq (Hz)
A: (299.072K 59.3953m)
B: (900.82K -62.4126m)
delta: (601.808K -121.808m)
slope: -202.403n



harmonic="0";(dB20((v "vo_sh" ?r--: harmonic="0";(dB20((v "vo_sh" ?r
harmonic="0";(dB20((v "vo_sh" ?r
STOPBAND RESPONSE W/ DIFFERENT AMP GAIN
Plot 4
|H(f)| [dB]
-55dB
Gain = 500, 1200, 10^6
-50
-60
-70
-80
1.0M
2.0M
3.0M
4.0M
5.0M
freq (Hz)
⇒ Amp gain has very little effect on stop band attenuation!
A: (300.94K 67.5233m)
B: (900.82K -49.8657m)
delta: (599.88K -117.389m)
slope: -195.687n

Matlab Script ①

```
% EE247 Homework #4 Boris Murmann

% filter specification
clear;
Wpass = 2*pi*1e6;        % passband edge [rad]
Wstop = 2*pi*1.5e6;      % start of stopband [rad]
Rp    = 0.1;             % passband ripple (dB)
Rs    = 60;              % stopband attenuation (dB)

% sampling frequency and normalized band specifications
Wsamp = 2*pi*10e6;
Wp    = 2*Wpass/Wsamp;
Ws    = 2*Wstop/Wsamp;

% discrete time filter synthesis
[N,Wn] = ellipord(Wp, Ws, Rp, Rs);
[num,den] = ellip(N, Rp, Rs, Wn);
freqz(num,den);

% break up into biquads - matlab provides optimized pole zero grouping
[SOS,G] = tf2sos(num,den)
num1 = SOS(1,1:3);
num2 = SOS(2,1:3);
num3 = SOS(3,1:3);
den1 = SOS(1,4:6);
den2 = SOS(2,4:6);
den3 = SOS(3,4:6);

% calculate Q of continous time equivalent sections
[Z1, P1, K1] = tf2zp( num1, den1 );
[Z2, P2, K2] = tf2zp( num2, den2 );
[Z3, P3, K3] = tf2zp( num3, den3 );
s1 = (Wsamp/pi) * (P1(1)-1) / (P1(1)+1);
s2 = (Wsamp/pi) * (P2(1)-1) / (P2(1)+1);
s3 = (Wsamp/pi) * (P3(1)-1) / (P3(1)+1);
Q1 = 0.5*sqrt(1+(imag(s1)/real(s1))^2 )
Q2 = 0.5*sqrt(1+(imag(s2)/real(s2))^2 )
Q3 = 0.5*sqrt(1+(imag(s3)/real(s3))^2 )

% normalize to positive powers of z and divide by "a2" to match (5.27) Gregorian
NUM1 = num1 / den1(3);
NUM2 = num2 / den2(3);
NUM3 = num3 / den3(3);
DEN1 = den1 / den1(3);
DEN2 = den2 / den2(3);
DEN3 = den3 / den3(3);

% scale for unity gain at DC (z=1) for each biquad
k1 = sum(DEN1) / sum(NUM1);
k2 = sum(DEN2) / sum(NUM2);
k3 = sum(DEN3) / sum(NUM3);
NUM1 = k1 * NUM1;
NUM2 = k2 * NUM2;
NUM3 = k3 * NUM3;

% calculate component values, for Q>3 use hi Q circuit
C0=1e-12;
if Q1>3
   Caps1 = HiQ(NUM1, DEN1, C0)
else
   Caps1 = LoQ(NUM1, DEN1, C0)
end
if Q2>3
   Caps2 = HiQ(NUM2, DEN2, C0)
else
```

```
   Caps2 = LoQ(NUM2, DEN2, C0)
end
if Q3>3
   Caps3 = HiQ(NUM3, DEN3, C0)
else
   Caps3 = LoQ(NUM3, DEN3, C0)
end
```

(2)

3

```
% Lo Q (Q<3) Gregorian p.286
function  Caps = LoQ(NUM, DEN, C0)
a2 = NUM(1);
a1 = NUM(2);
a0 = NUM(3);
b2 = DEN(1);
b1 = DEN(2);
b0 = DEN(3);
C2 = sqrt(b1+b2+1);
C3 = sqrt(b1+b2+1);
C1x = a2-a0;
C1xx = a0;
C1 = (a0+a1+a2)/C3;
C4 = b2-1;
Caps = C0*[C1,C1x,C1xx,C2,C3,C4];


% Hi Q (Q>3) Gregorian p.291
function Caps = HiQ(NUM, DEN, C0)
a2 = NUM(1);
a1 = NUM(2);
a0 = NUM(3);
b2 = DEN(1);
b1 = DEN(2);
b0 = DEN(3);
C2 = sqrt( (1+b1+b2)/b2 );
C3 = sqrt( (1+b1+b2)/b2 );
C1x = (a2-a0) / (b2*C3);
C1xx = a2/b2;
C1 = (a0+a1+a2)/ (b2*C3);
C4 = (1 - (1/b2)) / C3;
Caps = C0*[C1,C1x,C1xx,C2,C3,C4];
```

Spectre Netlist ①

```
*** EE247 HW #4 Boris Murmann

simulator lang=spectre
parameters fs=10Meg  c0=1p Ron=1 AmpGain=1e6
//
*** first low Q section
+c11=     4.529440715720658e-013
+c1x1=    4.163336342344337e-029   // =zero
+c1xx1=   6.857132907366235e-014
+c21=     4.529440715720658e-013
+c31=     4.529440715720658e-013
+c41=     8.677048917763188e-013
//
*** second low Q section
+c12=     6.072036269121990e-013
+c1x2=    -8.881784197001252e-028  //=zero
+c1xx2=   2.985596869872886e-013
+c22=     6.072036269121991e-013
+c32=     6.072036269121991e-013
+c42=     4.049018365178336e-013
//
*** hi Q section
+c13=     6.302625148882051e-013
+c1x3=    -2.079239384327061e-027  //=zero
+c1xx3=   4.712310360217251e-013
+c23=     6.302625148882049e-013
+c33=     6.302625148882049e-013
+c43=     1.460155701618134e-013

*** stimulus
vin1    (vin      0)  vsource pacmag=1
//vin1    (vin     0)  vsource type=sine ampl=1 freq=fs/10
vphi1   (phi1     0)  vsource type=pulse val0=0 val1=1 period=1/fs
width=0.45/fs rise=.1n fall=.1n delay=0.05/fs
vphi2   (phi2     0)  vsource type=pulse val0=0 val1=1 period=1/fs
width=0.45/fs rise=.1n fall=.1n delay=0.55/fs

*** circuit
b1    (vin  vo1   phi1  phi2)  loq  pc0=c0 pc1=c11 pc1xx=c1xx1 pc2=c21
pc3=c31 pc4=c41
b2    (vo1  vo2   phi1  phi2)  loq  pc0=c0 pc1=c12 pc1xx=c1xx2 pc2=c22
pc3=c32 pc4=c42
b3    (vo2  vout  phi1  phi2)  hiq  pc0=c0 pc1=c13 pc1xx=c1xx3 pc2=c23
pc3=c33 pc4=c43

*** sample and hold output
c1    (vout  0)              capacitor c=100f
buf1  (voutbuf 0       vout  0)  vcvs gain=1
s2     (voutbuf vo_sh  phi1  0)  relay vt1=0 vt2=1 rclosed=1
ropen=100e9
c2    (vo_sh       0)              capacitor c=100f

*** sample and hold input (-> sinc response for correction)
s11     (vin      vth   phi2  0)  relay vt1=0 vt2=1 rclosed=1
ropen=100e9
c11   (vth  0)               capacitor c=100f
buf11 (vthbuf 0       vth   0)  vcvs gain=1
s21     (vthbuf vsinc   phi1  0)  relay vt1=0 vt2=1 rclosed=1
```

2

```
ropen=100e9
c21    (vsinc      0)                capacitor c=100f

********************************************************************
* Lo Q biquad
subckt loq (vi vos phi1 phi2)
  parameters pc0 pc1 pc1xx pc2 pc3 pc4
  rsim1      (vi    n2     phi2  phi1  a      n1)    rsim_series
Csamp=pc1
  rsim2      (v1    n4     phi1  phi2  n3     b )    rsim_series_inv
Csamp=pc3
  c1xx             (vi    n4)    capacitor   c=pc1xx
  c2         (n1    n5)    capacitor   c=pc2
  c4         (n5    b)     capacitor   c=pc4
  switch1          (n5    0      phi1   0) relay vt1=0 vt2=1 rclosed=Ron
ropen=100e9
  switch2          (n5    vo     phi2   0) relay vt1=0 vt2=1 rclosed=Ron
ropen=100e9
  switch3          (vos          vo     phi2   0) relay vt1=0 vt2=1
rclosed=Ron ropen=100e9
  amp1       (v1    0      0      n2) vcvs gain=AmpGain
  amp2       (vo    0      0      n4) vcvs gain=AmpGain
  ca         (n2    v1)    capacitor   c=pc0
  cb         (n4    vo)    capacitor   c=pc0
ends loq

* Hi Q biquad
subckt hiq (vi vos phi1 phi2)
  parameters pc0 pc1 pc1xx pc2 pc3 pc4
  rsim1      (vi    n2     phi2  phi1  a      n1)    rsim_series
Csamp=pc1
  rsim2      (v1    n4     phi1  phi2  n3     b )    rsim_series_inv
Csamp=pc3
  c1xx             (vi    n4)    capacitor   c=pc1xx
  c2         (n1    n5)    capacitor   c=pc2
  c4         (n2    vo)    capacitor   c=pc4
  switch1          (n5    0      phi1   0) relay vt1=0 vt2=1 rclosed=Ron
ropen=100e9
  switch2          (n5    vo     phi2   0) relay vt1=0 vt2=1 rclosed=Ron
ropen=100e9
  switch3          (vos          vo     phi2   0) relay vt1=0 vt2=1
rclosed=Ron ropen=100e9
  amp1       (v1    0      0      n2) vcvs gain=AmpGain
  amp2       (vo    0      0      n4) vcvs gain=AmpGain
  ca         (n2    v1)    capacitor   c=pc0
  cb         (n4    vo)    capacitor   c=pc0
ends hiq

* simulated resistor with capacitor in series
subckt rsim_series (vl vr phi1 phi2 n1 n2)
  parameters Csamp
  cs      (n1   n2)           capacitor c=Csamp
  s1      (vl   n1   phi1 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  s2      (n1   0    phi2 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  s3      (n2   0    phi2 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  s4      (n2   vr   phi1 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  cp1   (n1   0)            capacitor   c=1f
  cp2   (n2   0)            capacitor   c=1f
```

3

```
ends rsim_series

* simulated resistor with capacitor in series and charge inversion
subckt rsim_series_inv (vl vr phi1 phi2 n1 n2)
  parameters Csamp
  cs      (n1  n2)            capacitor c=Csamp
  s1      (vl  n1  phi1 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  s2      (n1  0   phi2 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  s3      (n2  0   phi1 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  s4      (n2  vr  phi2 0 ) relay vt1=0 vt2=1 rclosed=Ron ropen=100e9
  cp1  (n1  0)            capacitor  c=1f
  cp2  (n2  0)            capacitor  c=1f
ends rsim_series_inv
*********************************************************************

pss2  pss   fund=fs   maxacfreq=fs/2 tstab=1/fs/4
pac2  pac   start=1k  stop=fs/2 lin=1000
*tran1 tran  stop=10/fs method=gear2only

simOptions options
+         rawfmt=psfbin
+           reltol=1e-5
+           vabstol=1n
+           iabstol=1p

alter1 alter param=AmpGain value=1.2e3
pss3  pss   fund=fs   maxacfreq=fs/2 tstab=1/fs/4
pac3  pac   start=1k  stop=fs/2 lin=1000

alter2 alter param=AmpGain value=0.5e3
pss4  pss   fund=fs   maxacfreq=fs/2 tstab=1/fs/4
pac4  pac   start=1k  stop=fs/2 lin=1000
```