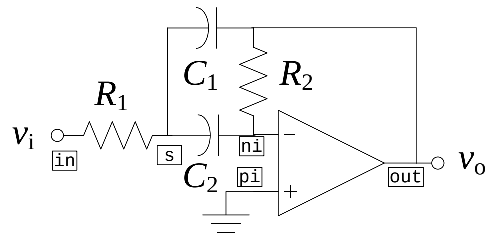
Dr. J. E. Rayas-Sánchez

OPTIMIZING A MULTIPLE-FEEDBACK 2ND ORDER BAND-PASS FILTER: PROBLEM STATEMENT

Consider the following multiple-feedback 2nd order band-pass active filter. At a resonant frequency f_0 = 15 KHz, the magnitude of the voltage gain should be $|A_v|$ = 800, with a bandwidth BW = 750 Hz.



The following Matlab script implements classical formulas to design this type of filters using ideal Op-Amps.

```
%         Design Procedure for an Ideal Multiple-Feedback Band-Pass Filter
%
% This program calculates the circuit element values of an ideal multiple-feedback
% band-pass active filter.
% Reference:
% S. Franco, Design with Operational Amplifiers and Analog Integrated
% Circuits. New York, NY: McGraw-Hill, 1988, pp. 130-131.

fo = 15e3;   % Resonant frequency (Hz).
BW = 750;    % Bandwidth (Hz).
C = 1e-9;    % Capacitors C1=C2 (F).
Q = fo/BW;   % Quality factor.
R2 = Q/(pi*fo*C);
R1 = R2/(4*Q^2);
Resonant_gain = -2*Q^2;
```

If $C_1 = C_2$ = 1nF, and f_0 = 15 KHz with BW = 750 Hz, then

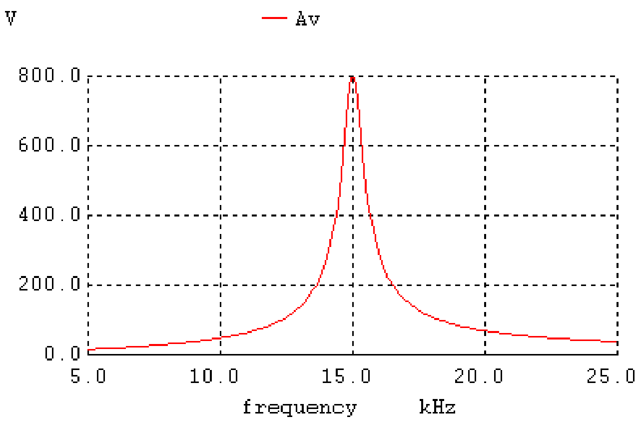
Q = 20, $|A_v|$ = 800 at f_0, yielding R_1 = 265.26 Ω and R_2 = 424.41 KΩ.

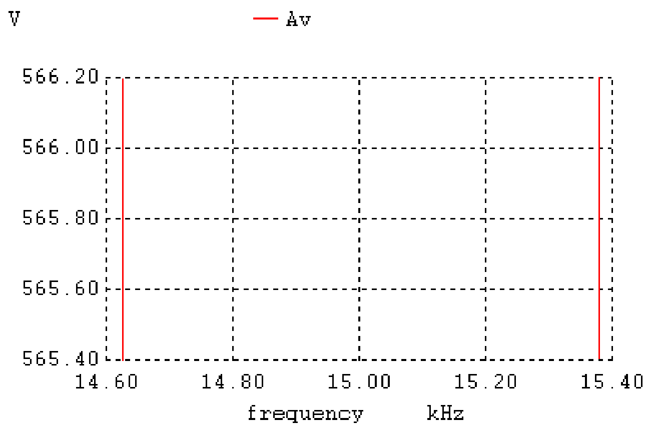
Simulating the corresponding filter with SPICE using a quasi-ideal Op-Amp:

```
Ideal Multiple-Feedback 2nd Order Band-Pass Filter

Vi   in   0    DC 0V AC 1V
R1   in   s    265.26
R2   out  ni   424.41K
C1   s    out  1nF
C2   s    ni   1nF
XOA1 ni 0 out QuasiIdealOpAmp

.lib C:\command_line_ WinSpice\MisModelosSpice.txt

.control
delete all
AC LIN 300 5KHz 25KHz
Av = mag(v(out)/v(in))
plot Av
.endc
.end
```

Notice that the Op-Amp model `QuasiIdealOpAmp` is defined within the library file `MisModelosSpice.txt`.

SPICE results are shown below, fulfilling the design specifications almost exactly.



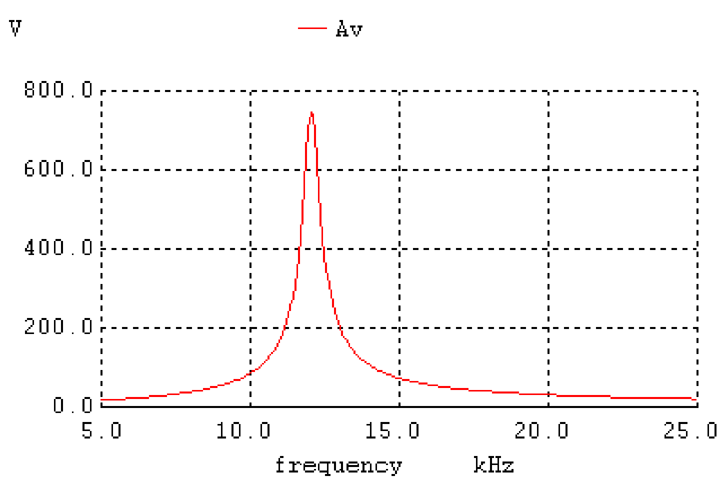


Since $800/\sqrt{2} = 565.68$, notice that the obtained bandwidth is almost 750 Hz (15.38KHz – 14.63KHz).

Now the filter is simulated using a more realistic Op-Amp model:

```
Multiple-Feedback 2nd Order Band-Pass Filter

Vcc  vp   0    DC 12V
Vee  vn   0    DC -12V
Vi   in   0    DC 0V AC 1V
R1   in   s    265.26
R2   out  ni   424.41K
C1   s    out  1nF
C2   s    ni   1nF
XOA1 0 ni vp vn out uA741
.lib C:\command_line_ WinSpice\MisModelosSpice.txt

.control
delete all
AC LIN 300 5KHz 25KHz
Av = mag(v(out)/v(in))
plot Av
.endc
.end
```

The new SPICE results are shown below, which clearly violates the design specifications.



The Op-Amp models used in the above SPICE netlists are:

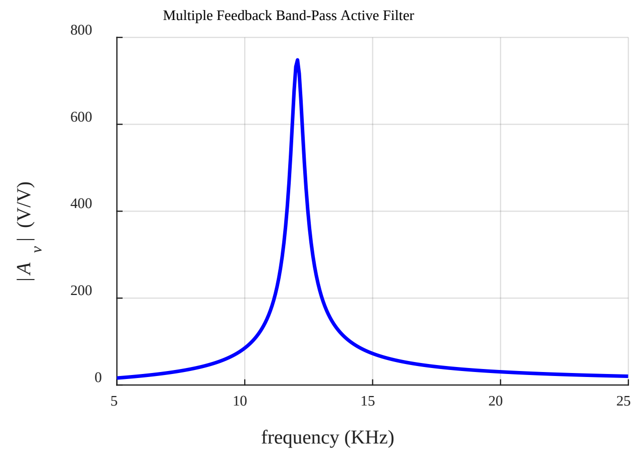
```
* Subcircuit for a Quasi Ideal OpAmp
.SUBCKT QuasiIdealOpAmp N P OUT
Ri N P 1000MEGAohms
Ro I OUT 0.01ohm
Eo I 0 P N 100E6
.ENDS


* UA741 OPERATIONAL AMPLIFIER "MACROMODEL" SUBCIRCUIT
* Texas Instruments
* CREATED USING PARTS RELEASE 4.01 ON 07/05/89 AT 09:09
* (REV N/A)      SUPPLY VOLTAGE: +/-15V
* CONNECTIONS:   NON-INVERTING INPUT
*                | INVERTING INPUT
*                | | POSITIVE POWER SUPPLY
*                | | | NEGATIVE POWER SUPPLY
*                | | | | OUTPUT
*                | | | | |
.SUBCKT UA741    1 2 3 4 5
*
  C1   11 12 4.664E-12
  C2    6  7 20.00E-12
  DC    5 53 DX
  DE   54  5 DX
  DLP  90 91 DX
  DLN  92 90 DX
  DP    4  3 DX
  EGND 99  0 POLY(2) (3,0) (4,0) 0 .5 .5
  FB    7 99 POLY(5) VB VC VE VLP VLN 0 10.61E6 -10E6 10E6 10E6 -10E6
  GA 6  0 11 12 137.7E-6
  GCM 0  6 10 99 2.574E-9
  IEE  10  4 DC 10.16E-6
  HLIM 90  0 VLIM 1K
  Q1   11  2 13 QX
  Q2   12  1 14 QX
  R2    6  9 100.0E3
  RC1   3 11 7.957E3
  RC2   3 12 7.957E3
  RE1  13 10 2.740E3
  RE2  14 10 2.740E3
  REE  10 99 19.69E6
  RO1   8  5 150
  RO2   7 99 150
  RP    3  4 18.11E3
  VB    9  0 DC 0
  VC 3 53 DC 2.600
  VE   54  4 DC 2.600
  VLIM  7  8 DC 0
  VLP  91  0 DC 25
  VLN   0 92 DC 25
.MODEL DX D(IS=800.0E-18)
.MODEL QX NPN(IS=800.0E-18 BF=62.50)
.ENDS
```

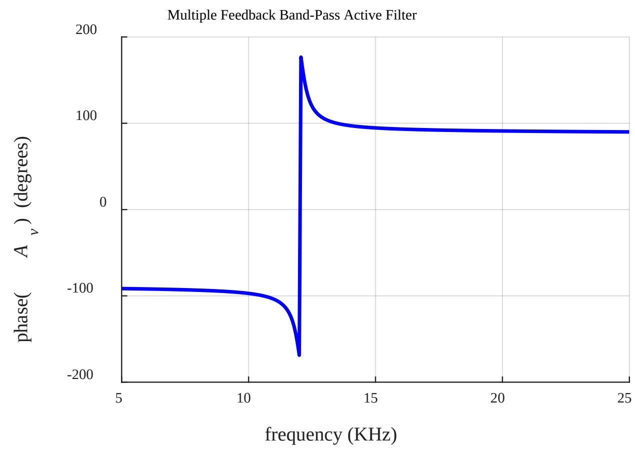
Creating a Matlab driver to simulate the filter in WinSpice:

```matlab
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%         Matlab Driver for a Multiple-Feedback 2nd Order Band-Pass
%                       Filter Simulated in WinSpice
%
% This function drives WinSpice to simulate a multiple-feedback 2nd order
% band-pass filter in frequency domain using a realistic Op-Amp model.
% Usage: [f,mAv,pAv] = MFBP2_SPICE(x,IF,FF,FP)
%         x: vector of selected design variables.
%         IF: Initial frequency (Hz).
%         FF: Final frequency (Hz).
%         FP: Number of frequency points per frequency sweep.
%         f: column vector containing the FP simulated frequency points (Hz).
%         mAv: column vector containing the magnitude of the voltage gain (V/V).
%         pAv: column vector containing the phase of the voltage gain (degrees).
% Functions required: None.

function [f,mAv,pAv] = MFBP2_SPICE(x,IF,FF,FP)

% WinSpice Executable File in Command Line Mode
WinSpiceExe = 'C:\command_line_WinSpice\wspice3';

% WinSpice Project File Name
WinSpiceProjectFileName = 'MFBP2_m.cir';

% Design Variables, x
R1 = x(1);
R2 = x(2)*1e3;  % Convert R2 from Kohms to ohms.
C1 = x(3)*1e-9; % Convert C from nF to F.
C2 = C1;

% Define WinSpice Script, ws
ws{1} = 'Multiple-Feedback 2nd Order Band-Pass Filter';
ws{2} = 'Vcc  vp   0    DC 12V';
ws{3} = 'Vee  vn   0    DC -12V';
ws{4} = 'Vi   in   0    DC 0V AC 1V';
ws{5} = ['R1 in s ' mat2str(R1)];
ws{6} = ['R2 out ni ' mat2str(R2)];
ws{7} = ['C1 s out ' mat2str(C1)];
ws{8} = ['C2 s ni ' mat2str(C2)];
ws{9} = 'XOA1  0 ni vp vn out uA741';
ws{10} = '.lib C:\command_line_WinSpice\MisModelosSpice.txt';
ws{11} = '.control';
ws{12} = 'delete all';
ws{13} = ['AC LIN ' mat2str(FP) ' ' mat2str(IF) ' ' mat2str(FF)];
ws{14} = 'mAv = mag(v(out)/v(in))';
ws{15} = 'pAv = phase(v(out)/v(in))';
ws{16} = 'write AC_response.csv mAv pAv';
ws{17} = 'quit';
ws{18} = '.endc';
ws{19} = '.end';

% Save WinSpice Script as a Circuit File in Matlab Working Directory
ckt_file = char(ws);
[rows,~] = size(ckt_file);
fid = fopen(WinSpiceProjectFileName,'w+');  % File identifier opened.
for i = 1:rows
   fprintf(fid, '%s', ckt_file(i,:)); % Save each row of ckt_file.
   fprintf(fid, '%s\r\n', '');
end
fclose(fid);  % File identifier closed.

% Run WinSpice
system([WinSpiceExe ' ' WinSpiceProjectFileName]);
```

```
% Read WinSpice Output Files
RespAC = csvread('AC_response.csv',1,0); % Read AC responses.
f = RespAC(:,1);
mAv = RespAC(:,3);
pAv = RespAC(:,5)*180/pi;

% Erase Data Files
delete AC_response.csv;
```

Running the previous Matlab function and plotting the corresponding responses:





Based on the previous Matlab function, write a suitable objective function to optimize the filter such that it fulfills the desired performance for $|A_v|$. For the optimization method use: a) the Nelder-Mead method available in Matlab (`fminsearch`); b) your algorithm for Conjugate Gradient optimization; c) your algorithm for Quasi-Newton optimization.