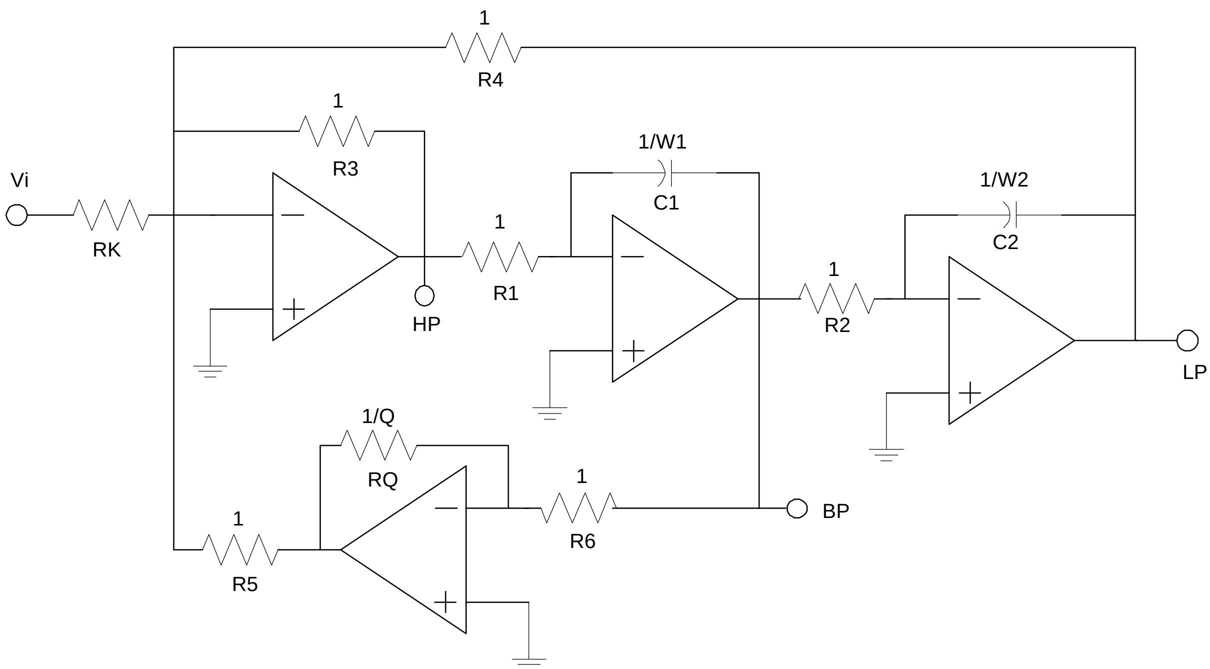
THIS HANDOUT WILL HELP WITH A GENERALIZED BIQUAD FOR ANY TYPE FILTER AND HOW TO GENERATE A SWITCH CAP FILE, SIMULATE IT AND PLOT THE RESULTS.



$$\frac{BP}{Vi} = \frac{Kw_1}{s^2 + \frac{w_1}{Q}s + w_1 w_2}$$

$$\frac{LP}{Vi} = \frac{-Kw_1 w_2}{s^2 + \frac{w_1}{Q}s + w_1 w_2}$$

$$\frac{HP}{Vi} = \frac{-Ks^2}{s^2 + \frac{w_1}{Q}s + w_1 w_2}$$

```
* HSPICE deck
* Tow Thomas - Bandpass and Lowpass
*
*
CW1   4 5 '1/(w1*r)'
R1    3 4 r
CW2   6 7 '1/(w2*r)'
R2    5 6 r
R3    2 3 r
R4    2 7 r
R5    9 2 r
R6    5 8 r
RQ    8 9 'r/q'
Rk    1 2 'r/k'
*
XOPAMP1     0 2 vdd vss 3 LM741/NS
XOPAMP2     0 4 vdd vss 5 LM741/NS
XOPAMP3     0 6 vdd vss 7 LM741/NS
XOPAMP4     0 8 vdd vss 9 LM741/NS
*
V0 1 0 ac 1.000000e+00
Vdd   vdd 0 DC 15
Vss   vss 0 DC -15
*
.param r=10.0k k=1.0 q=.707 w1=6.28k w2=6.28k
.op
.option post
.include "~/ee458/models/LM741.MOD"
.ac dec 25 1.000000e+01 1.000000e+05 sweep k dec 10 0.1 10
.print ac vdb(3) vdb(5) vdb(7) vp(3) vp(5) vp(7)
.end
```

In the *.param* statement, set the gain with *k* and set the quality factor with *q*. Also, set the two w_I respectively and *r* is the scaling factor. If you sum the high pass and low pass outputs will get a notch filter. Try this circuit it works pretty good. The sweep command in the *.ac* command allows one to vary any variable in the *.param* statement.

INPUT FILE DESCRIPTION FOR SWITCH CAP

There are three main parts to the file; timing, circuit, and analysis. The timing portion describes the clock or clocks being used to run the circuit. Next, the circuit section is very similar to hspice circuit description with all the switches, capacitors, and op amps connected together. Finally, the analysis part setups the type of analysis to be performed on the circuit, this also is very similar to hspice analysis setup. The following is a made up circuit (does not work) and each part will be discussed.

```
timing;
```

This defines the beginning of the timing block.

```
period 8.000000e-06;
```

This defines the master clock period to be 8 us, and it establishes that 1mcp, absolute unit of time, corresponds to 8 us.

```
clock clk 1 (0 4/8);
```

This statement defines a switching clock with the name “clk”. The “1” defines the period of the clock in 1mcp; therefore, the clock has period of 8 us. Inside the parenthesis, these numbers define the duration of the high interval of the clock with respect to 1mcp. In the example, the first half of the clock is high for 4 us.

```
end;
```

This signals the end of the of the block being described.

```
circuit;
```

This defines the beginning of the circuit description.

```
v1  ( 1  0);
```

Placement of a voltage source is done by starting the name of the source with “v” (in the example we have “v1” connected to node 1 and 0). Inside the parenthesis are the node connections. As in hspice “0” is globally defined as ground.

```
c1  ( 8  9) 2.215467e-08;
```

Placement of a capacitor element is done by starting the name of the capacitor with “c” (in the example we have “c1” connected to node 8 and 9). Inside the parenthesis are the node connections and value of the capacitor at the end as in hspice “0”.

```
c2  (12 13) 2.215467e-08;
c3  ( 5  6) 1.000000e-09;
c4    (16 17) 3.150003e-08;
```

```
cw2 (14 15) 2.843634e-09;
cw3 (10 11) 1.421823e-09;
ck  ( 2  3) 1.000000e-09;
cq  (18 19) 1.000000e-09;


s1  ( 1  2) clk;
```

Placement of a switch is done by starting the name of the switch with “s” (in the example we have “s1” connected to node 1 and 2). Inside the parenthesis are the node connections and outside at the end is the clock that controls the switch.

```
s2  ( 2  0) #clk;
```

This is the same as above except the clock controlling the switch is different. The negate sign (#) indicates that “s2” is controlled by the complementary value of clock “

```
s3  ( 3  0) #clk;
s4  ( 3  4) clk;

s5  ( 4  5) clk;
s6  ( 5  0) #clk;
s7  ( 6  0) #clk;
s8  ( 6  7) clk;

s9  ( 7  8) clk;
s10 ( 8  0) #clk;
s11 ( 9  0) #clk;
s12 ( 9 10) clk;

s13 (11 12) clk;
s14 (12  0) #clk;
s15 (13  0) #clk;
s16 (13 14) clk;

s17 (15 16) clk;
s18 (16  0) #clk;
s19 (17  0) #clk;
s20 (17  4) clk;

s21 (11 18) clk;
s22 (18  0) #clk;
s23 (19  0) #clk;
s24 (19  4) clk;

e1  (7 0 0 4) 2000000;
```

Placement of a voltage controlled voltage source is done by starting the name of the source with “e”. Inside the parenthesis are the node connections(in the example we have “e1” with the positive output node 7, negative output node 0, positive input node 0, and negative input node 4). The gain of the op amp is at the end (2000000).

```
e2  (11 0 0 10) 2000000;
```

```
e3  (15 0 0 14) 2000000;

end;
```

This signals the end of the of the block being described.

```
analyze sss;
```

This defines the beginning of the analysis description.

```
infreq 1.000000e+01 1.000000e+06 log 100;
```

This statements sweeps the frequency logarithmically, “log” and performs and analysis of 100 points from 10Hz to 1MHz.

```
set v1 ac 1.0 0.0;
```

This sets the voltage source, “v1”, to an ac source with amplitude “1” and phase of “0”.

```
print vdb(11) vp(11);
```

The print statement is like in hspice. Vdb(11) specifies the voltage magnitude in decibels at node 11, and vp(11) specifies the phae of the voltage at node 11. Other possible options are as follows. Vr - real part. Vi – imaginary part. Vm – magnitude part. Vgd – group delay.

```
end;
```

This signals the end of the of the block being described.

```
end;
```

This signals the end of the file.

SIMULATING SWITCH CAP FILE AND PLOTTING THE RESULTS

To simulate a switch cap file, type "sw2" at command prompt as shown below.

```
eesun3:/home13/kanji/junk>sw2
SWITCAP2 Archive:v1.1        As of:Nov-01-90
  Copyright (c) 1990 by Columbia University
Enter input file name (. to quit):
```

Now type the file name.

```
Enter input file name (. to quit): swcapbp.swi
```

Then it will ask for an output file name, type an output file name, and it will return to the command prompt.

```
Enter output file name: swcapbp.dat
eesun3:/home13/kanji/junk>
```

First you must edit the output file before plotting it with *gnuplot*. Open the file with a text editor and check to see if the switch cap file simulated. At the start of the file it will always have the input file. If the simulation failed due to an error, the file will have these errors after the input file portion. If the simulation was successful then a new section called "Sinusoidal Steady State Analysis" will appear. In this section it will have the frequency range requested plus all the items requested by the print statement in the input file. You need to delete all the text above the first row of numbers. In other words make the file look like it is filled with rows and columns of numbers. Now be sure there are no blank lines between the top of the page and first row of numbers then save the file.

Now you are ready to plot the results with *gnuplot*. At the command prompt type *gnuplot* and the following should occur.

```
 eesun3:/home13/kanji/ee458/lab/lab6>gnuplot

        G N U P L O T
        unix version 3.5
        patchlevel 3.50.1.17, 27 Aug 93
        last modified Fri Aug 27 05:21:33 GMT 1993

        Copyright(C) 1986 - 1993   Thomas Williams, Colin
Kelley

        Send comments and requests for help to info-
gnuplot@dartmouth.edu
```

```
        Send bugs, suggestions and mods to bug-
gnuplot@dartmouth.edu

Terminal type set to 'x11'
gnuplot>
```

Now, type the following to set up the graph and the statements are self-explanatory.

```
gnuplot> set logscale x
gnuplot> set grid
gnuplot> set xlabel "Frequency(Hz)"
gnuplot> set ylabel "Magnitude(dB)"
gnuplot> set title "2nd Order Butterworth Band Pass Filter
- Magnitude vs Frequency”
```

Now you are ready to plot the output file so type the following statement.

```
gnuplot> plot "swcapbp.dat" using 1:2 "%f%f%f" w lines
```

`plot` – means plot
`“swcapbp.dat”` – is the data file
`using 1:2` – use column one as x axis and column two as y axis
`“%f%f%f”` – tells gnuplot the number of columns and formatt they are in. In the example we have three columns in the data file in scientific notation.
`w lines` – connect data points with a continuous line

In the example column one is the frequency, column two is the magnitude, and column three is the phase. The statement above plotted the magnitude vs frequency. The statements below plot the phase vs frequency and renames the y axis.

```
gnuplot> set ylabel "Phase(degrees)"
gnuplot> plot "swcapbp.dat" using 1:3 "%f%f%f" w lines
```

The statements below can used to print the graph to a printer. The first one prints the graph to the default printer in the eesun lab. The next one allows the user to determine the printer. (hpserc_1 is the printer in WERC 318).

```
gnuplot> set output "prn"
gnuplot> set output "|lp -d hpserc_1"
```