First, enter in the discrete data for the desired transfer function. To do this, we will create two vectors: f, Hd. f contains the discrete frequency points, and Hd contains the ideal transfer function. To create the vector f, first type "f" then ":". The colon will give := for an assignment. Then choose "Matrix" from the "insert" menu. Enter in the # of rows that you need, and 1 column to form the vector. Enter in the value for each, and hit the tab key to move to the next block.

$nf := 12$

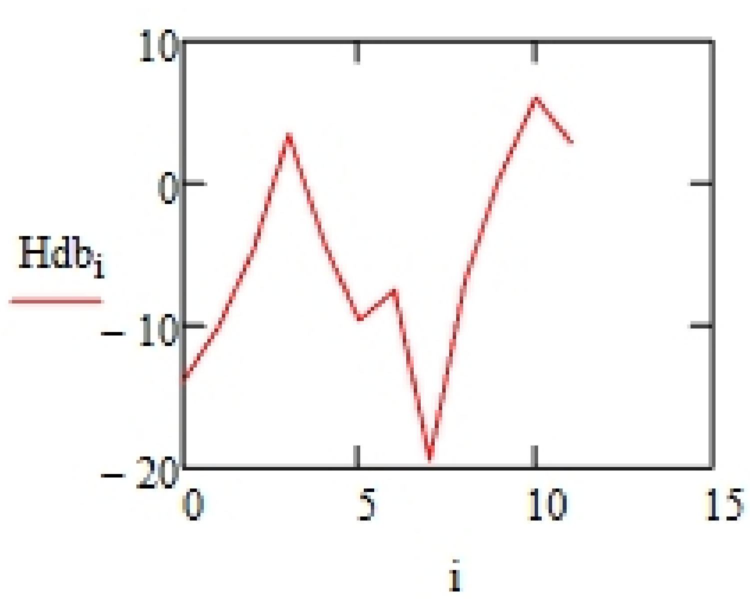
$$fd := \begin{pmatrix} 40 \\ 60 \\ 100 \\ 200 \\ 400 \\ 600 \\ 1000 \\ 2000 \\ 4000 \\ 6000 \\ 8000 \\ 10000 \end{pmatrix} \qquad Hd := \begin{pmatrix} 0.2 \\ 0.31 \\ 0.579 \\ 1.487 \\ 0.618 \\ 0.329 \\ 0.421 \\ 0.107 \\ 0.455 \\ 1.069 \\ 1.995 \\ 1.387 \end{pmatrix}$$

Next, calculate Hd in dB. To create the looping index, first type "i" ":" (for the assignment), then "0 ; 13" Note that the semicolon give the range (symbol of ..). Also note that the vector is indexed from 0 to 13, not 1 to 14. Also, to get the subscript for Hdb or Hd, type "["

$i := 0 .. nf - 1$

$Hdb_i := 20 \cdot \log(Hd_i)$

Plot Hdb as a check: Choose the graph toolbar from the right-hand-side of the MathCad window. Then, select the X-Y plot icon. On the vertical axis, type "Hdb[i". On the horizontal axis, type "i", then enter:



Define a function representing an ideal bandpass filter. The variables of the function are p (complex freq), Av (voltage gain), fo (resonant freq), and Q. Note, when entering equations use the spacebar to control the "underscore" location. This determines the active area. Secondly, to raise something to a power, use ^. i.e., p-square would be entered as "p^2", then hit the spacebar to complete the task and continue entering in data. When entering a quotient, use the spacebar to underscore all the variables that will be in the numerator. Then, hit "/" to establish the quotient. When done, use the space bar to move out of the denominator and back up to the main line.

$$Hbp(p,Av,fo,Q) := Av\cdot\frac{2\cdot\pi\cdot fo}{Q}\cdot\frac{p}{p^2+\frac{2\cdot\pi\cdot fo}{Q}\cdot p+(2\cdot\pi\cdot fo)^2}$$

Define the top-level design as a superposition of three parallel bandpass filters combined through a summing amp. Note that the gains are controlled via the filter as well as the summing amp itself.

$$Htop(p,A1,f1,Q1,A2,f2,Q2,A3,f3,Q3) := Hbp(p,A1,f1,Q1) + Hbp(p,A2,f2,Q2) + Hbp(p,A3,f3,Q3)$$

From the data set, our group decides to use a summing amp of two band pass filters, with resonant frequencies at 200 Hz and 8 kHz. A simple loop is created to alter the gains and the Q's of the filters, and the mean is computed. To do this, initially determine the bounds of the gains and the Q's. Here, we will vary the gain of filter 1 from 1.1 to 1.6 (over 6 steps), the gain of filter 2 from 1.8 to 2.3, the Q of filter 1 from 1 to 3.5 (over 6 steps), and the Q of filter 2 from 1 to 3.5 (again over 6 steps).

$f1 := 200$	$A10 := 1.4$	$dA1 := 0.1$	$Q10 := 1.5$	$dQ1 := 0.5$
$f2 := 1000$	$A20 := 0.2$	$dA2 := 0.05$	$Q20 := 12$	$dQ2 := 0.5$
$f3 := 8000$	$A30 := 1.8$	$dA3 := 0.1$	$Q30 := 1$	$dQ3 := 0.5$

Next, define a program block to scan through the values of the voltage gains and Q's. The program block will scan through all the values, then find the minimum variance error (note this can be modified to find the minimum mean error). For the minimum variance error, store the values of A1, A2, Q1, and Q2 that resulted in that error.

To enter in this program block, first type "vmin:" for the assignment.
Next, select the "Programming Toolbar". On the programming toolbar, select "Add line".
A program block will come up. With the cursor in the first placeholder, select "Add line" a few more times, to add more lines to the program block (Note that with MathCad, always create more lines than you need. You can come back and delete them later).
Now, move the cursor to the first placeholder of the program block and then enter A1m <-- 0 (where the arrow is from the programming toolbar) to intialize A1m.
Continue on to initialize the other variables. Note here vm will be the minimum variance error. However, it has been assigned a very large value intially. Once this is entered in, the loops need to be entered. To this end, select "for" from the programming toolbar.
Then, type "0;5" in the right-hand place holder, and the variable

i1 in the left-hand place holder. Next, move the cursor to the placeholder in the indented line of the for loop. Add a few lines here by clicking "Add Line" from the programming toolbar. Then, enter A1 <-- A10+dA1*i1. This assigns a value to A1 that is incremented by i1*dA1. Note that again, the arrow is selected from the toolbar.
Continue with this process to enter in all the loops, scanning successively through A1, A2, Q1, and Q2. In the inner most loop. In the inner most loop, we will compute the transfer function of the sum of our two band pass filters. Also, the mean and then the variance are calculated. Finally, conditional statements are added in. Note that the "if" here must be selected from the programming toolbar. Effectively, these statements assign a new value to the variables A1m, A2m, Q1m, Q2m if the current variance error is less than the current minimum error. The value of vm (the minimum variance error) is also updated.
Finally, a vector is created containing the values of A1m, A2m, Q1m, and Q2m that resulted in the minimum variance error. Also, the variance error and the mean error are also included in this vector. This vector is assigned to vmin.

Note that this only finds the minimum within your specified bounds. Thus, we have to be careful in properly specifying the bounds of the computation and the stepsize.

```
vmin := | A1m ← 0
        | A2m ← 0
        | Q1m ← 0
        | Q2m ← 0
        | vm ← 9999
        | for i1 ∈ 0 .. 5
        |   | A1 ← A10 + dA1·i1
        |   | for i2 ∈ 0 .. 5
        |   |   | A2 ← A20 + dA2·i2
        |   |   | for j1 ∈ 0 .. 5
        |   |   |   | Q1 ← Q10 + dQ1·j1
        |   |   |   | for j2 ∈ 0 .. 5
        |   |   |   |   | Q2 ← Q20 + dQ2·j2
        |   |   |   |   | for i3 ∈ 0 .. 5
        |   |   |   |   |   | A3 ← A30 + dA3·i3
        |   |   |   |   |   | for j3 ∈ 0 .. 5
        |   |   |   |   |   |   | Q3 ← Q30 + dQ3·j3
        |   |   |   |   |   |   | μ ← 0
        |   |   |   |   |   |   | for k1 ∈ 0 .. nf − 1
        |   |   |   |   |   |   |   | ht ← Htop(2j·π·fd_k1, A1, f1, Q1, A2, f2, Q2, A3, f3, Q3)
        |   |   |   |   |   |   |   | htm ← |ht|
        |   |   |   |   |   |   |   | htdb ← 20·log(htm)
        |   |   |   |   |   |   |   | μ ← (μ + htdb) − Hdb_k1
        |   |   |   |   |   |   |      ..
```