QUICK START GUIDE
Audio EQ Cookbook HP15c CE software pac

Pepin Torres, P.E. pepin[dot]torres[at]gmail[dot]com

Introduction:

This software pac calculates all the filters coefficients and related parameters as defined in the classic document Cookbook formulae for audio equalizer biquad filter coefficients.

Who is this for?

This software pac is for any audio enthusiants, audio engineers, or software engineers that need to **generate the coefficients** for any of the 9 different filters in the cookbook (when there is no computer in sight). Furthermore, this pac also **calculates the complex frequency response** of any of the 9 filters at any arbitrary frequency f_x. One use-case would be to generate the coefficients of a filter with center frequency f_0 and calculate the complex response for a set of frequencies f_x (think of being able to plot the response by sampling at different f_x frequencies). Another use-case would be to run this program for a set of filters with different center frequencies f_0 and evaluate each filter at a fixed f_x to understand the gain/phase contributions of each filter at f_x (think of an EQ with multiple bands that overlap f_x).

Image file: **`audio_eq_cookbook.mem`**

Calculate All Filter Coefficients if you know f_0, F_s, Q, and A

`STO` F_s in `R9`
`STO` f_0 in `R.0`
`STO` Q in `R.1` (Q>0)
`STO` A in `R.2` (A>0)

Press **`GSB C`**

Recall a_n and b_n coeffiecients for a given filter via `RCL B` or `RCL A` by setting `R0` and `R1` to the desired row/column index (remember that in user mode index increments are automatic and most helpful). See table below:

Filter	Row Matrix B	Row Matrix A
Low Pass	1	1
High Pass	2	1
Band Pass Q peak gain	3	1
Band Pass 0dB peak gain	4	1
Notch Filter	5	1
All-Pass Filter	6	1
Peaking	7	2
Low Shelf	8	3
High Shelf	9	4

Note that the a_n coefficients are always identical for the first 6 filters, as such they are calculated once to save space.

If you need to calculate parameter A from filter gain in dB and shelf slope S (Peaking, Low-, and High-Self filters)

Place S in y-register
Place Gain in dB x-register

Press **GSB A**

This program will populate R.2 with A

If you need to calculate Q from the filter bandwidth

Place bandwidth in y-register. (The bandwidth must be given in fractions or multiples of an octave.)

Press **GSB B**

This program will populate R.1 with Q as a function of bandwidth. The bandwidth will be stored in R.9 for reference.

If you want to obtain the frequency response at an arbitrary frequency f_x

Place the B matrix row index of the filter you want to use in the y-register. (See table on first page.)
Place the frequency f_x in the x-register

Press **GSB D**

Register x will hold the complex response $H(f_x)$ in rectangular form. Press g →P to convert to polar. To get $|H(f_x)|$ from its rectangular form, press GSB 0. (value = $20\log_{10}$(abs(x-register)))
Note: Bonus Program 1 gets the linear value of a quantity in dB (value = $10^{dB/20}$). So, 3.0 GSB 1 returns 0.707

Memory usage

This program is HUGE (459 program bytes, 528 when placed in memory) and must be run on 15.2 mode only. The 5 matrices take up 71 registers (568 bytes) for a total mamory usage of 1,096 bytes. This leaves little room anything else, but given the magic of updatable images, this tool can be loaded when needed and replaced for something else less demanding.