EqualizIR Manual
v. 0.1
11 January, 2017

EQUALIZIR

INTRODUCTION

WHAT DOES IT DO?

The *EqualizIR* MATLAB application converts audio calibration data (usually from the *NICal* application) into a set of digital filter coefficients that can be used to equalize (a.k.a. pre-emphasis, compensation…) auditory stimuli so that the output more closely matches the original waveform.

The filter can be applied to stimuli stored in .WAV format from within the application ("Apply to WAV" menu item) or from the command line using the *ApplyFilterToWAV* function. Alternatively, the MATLAB *filt* or *filtfilt* functions can be used with a little more programming knowledge (*filtfilt* is used in the *ApplyFilterToWav* function).

Note that in its current implementation, the correction only operates on the **magnitude** of the signal and does not correct for **phase**. Future implementations might expand the correction to include phase. Or not…

WHERE CAN I FIND IT?

EqualizIR is part of the *TytoLogy* project housed at **github**:

https://github.com/TytoLogy/EqualizIR

Be sure to download/pull the 'installed' branch of code – other branches may (or may not) work!

WHAT ELSE IS NEEDED?

EqualizIR makes heavy use of other tools from the TytoLogy project.

The following toolboxes are needed:

https://github.com/TytoLogy/AudioToolbox

https://github.com/TytoLogy/UtilitiesToolbox

https://github.com/TytoLogy/PlotTools

as well as the MATLAB *Signal Processing Toolbox*

For acquisition of the calibration data, *NICal* (https://github.com/TytoLogy/NICal) can be used on systems with National Instruments DAQ cards. TDT hardware users can use the *TytoLogy* calibration programs in the *Calibrate* package (https://github.com/TytoLogy/Calibrate) or *SpeakerCal* (https://github.com/TytoLogy/SpeakerCal).

HOW DOES IT DO IT?

At its heart, EqualizIR uses the *invfreqz* MATLAB function to compute a minimum phase FIR filter. The filter so designed equalizes the magnitude of the signal while (hopefully) leaving the phase unaltered. The algorithm, as described by the author of invfreqz, Julius O. Smith at Stanford's CCRMA, is:

> "1. Interpolate the amplitude response samples from 0 to half the sampling rate, if necessary, and resample to a uniform "FFT frequency axis", if necessary. Denote the real, sampled amplitude response by S(k).
>
> 2. Perform an inverse FFT of log(S) to obtain the real cepstrum of s, denoted by c(n).
>
> 3. Fold the noncausal portion of c(n) onto its causal portion.
>
> 4. Perform a forward FFT, followed by exponentiation to obtain the minimum phase frequency response Sm(k), where now sm(n) is causal, and |Sm(k)|=S(k)."[1]

[1] https://ccrma.stanford.edu/~jos/pasp/Converting_Desired_Amplitude_Response.html

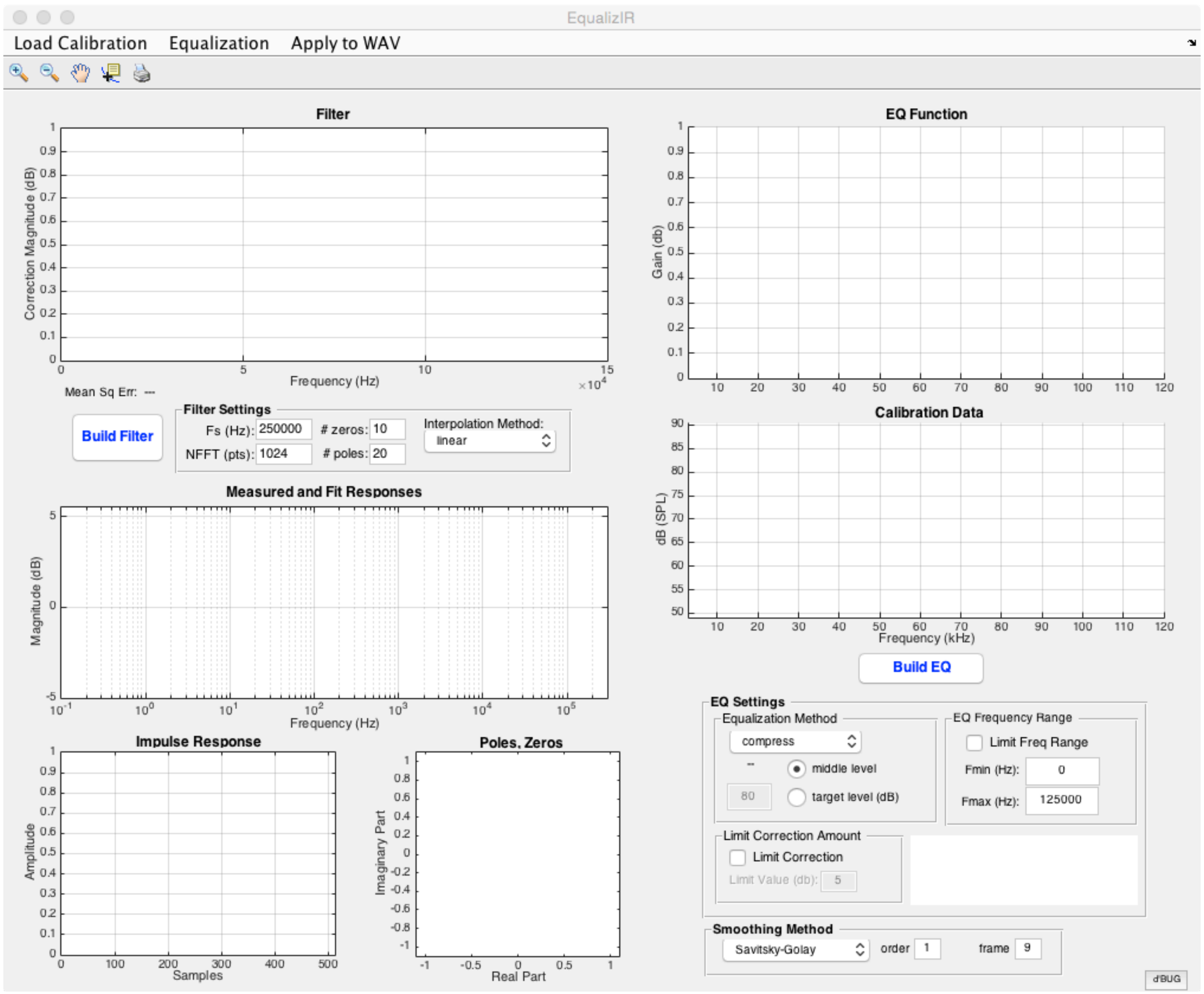
EQUALIZIR IN USE

START:

Launch the EqualizIR application by navigating to the EqualizIR directory and entering:

```
>> EqualizIR
```

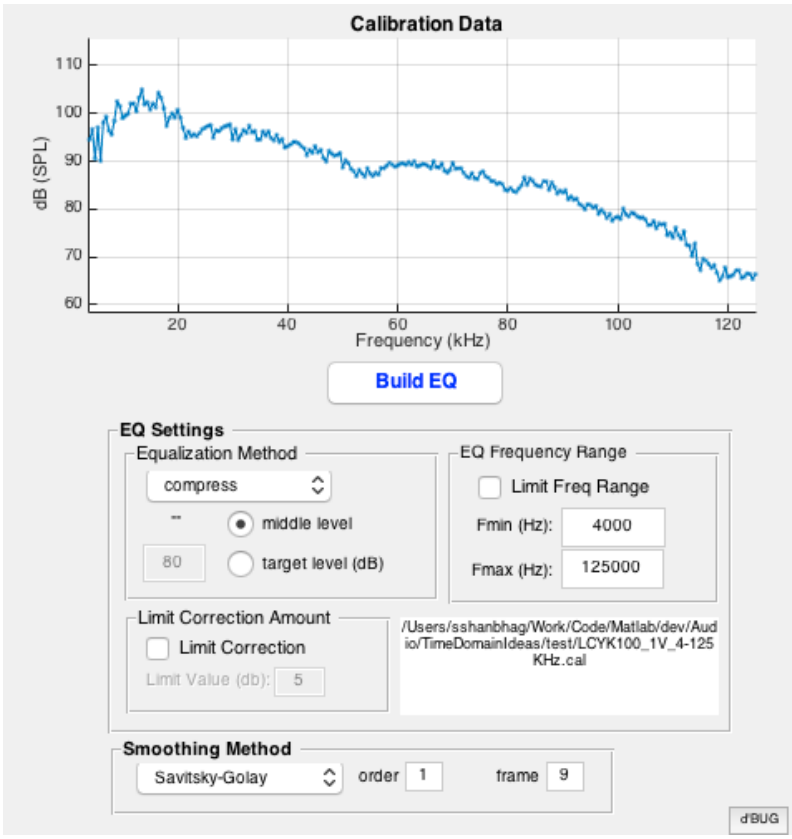
You should then see the EqualizIR application window:



LOAD CALIBRATION DATA:

The first thing to do is load your calibration data. Click on the "Load Calibration" menu and you'll be asked to select a .cal file. Locate your file and open it.

After selecting your calibration data file, the calibration magnitude data will be displayed in the (surprise!) "Calibration Data" graph:



BUILD THE EQUALIZATION FUNCTION:

Before creating the equalization filter, the calibration data need to be preprocessed.

This involves:

1. smoothing the calibration data to remove small dips/variations which can cause messy behavior by the filter
2. specifying the frequency range for the equalization filter to act upon
3. choosing an equalization method
4. (optional) limiting the correction amount

Smoothing Method

Two methods are available for smoothing the calibration data.

Savitsky-Golay filter (default)

Two parameters set the filter:

order	higher values = less smoothing values must be less than the frame size value	
frame	higher values = more smoothing values must be odd and greater than the order	

Moving Window

Only parameter is window size

EQ Frequency Range

When selected, allows the filter to be limited to a range of frequencies smaller than the full range of the calibration data

Equalization Method

Three methods are available:

- compress
  - finds midpoint between max and min SPL and attenuates or boosts magnitudes to this level –OR– uses target level*
- atten
  - finds lowest level within EQ frequency range and attenuates all magnitudes to this level
- boost
  - finds highest SPL and increases magnitudes to this level.

*for the compress methods, a target level may be specified (instead of the calculated middle range) by selecting the “target level” option and entering a target dB level in the edit box.

Limit Correction Amount

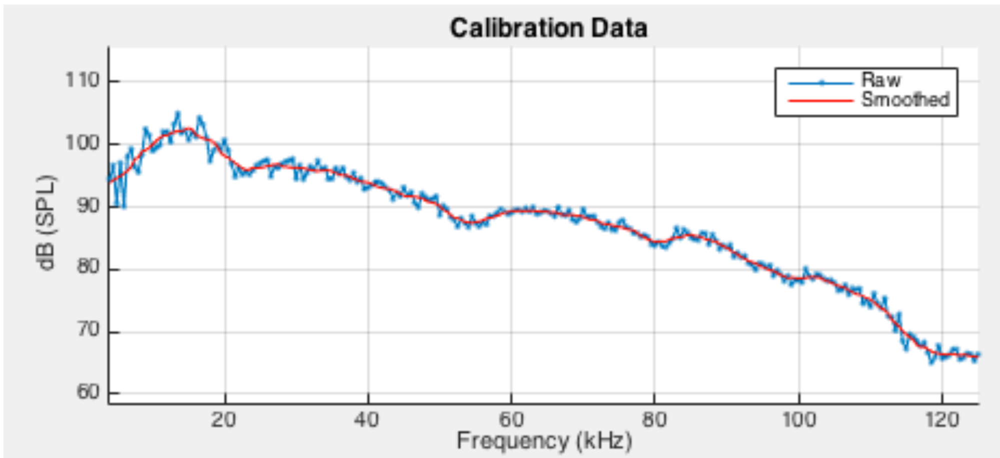
More than 15 dB of correction can run into problems. A hard limit can be set by selecting “Limit Correction” and entering a limit value

Build EQ

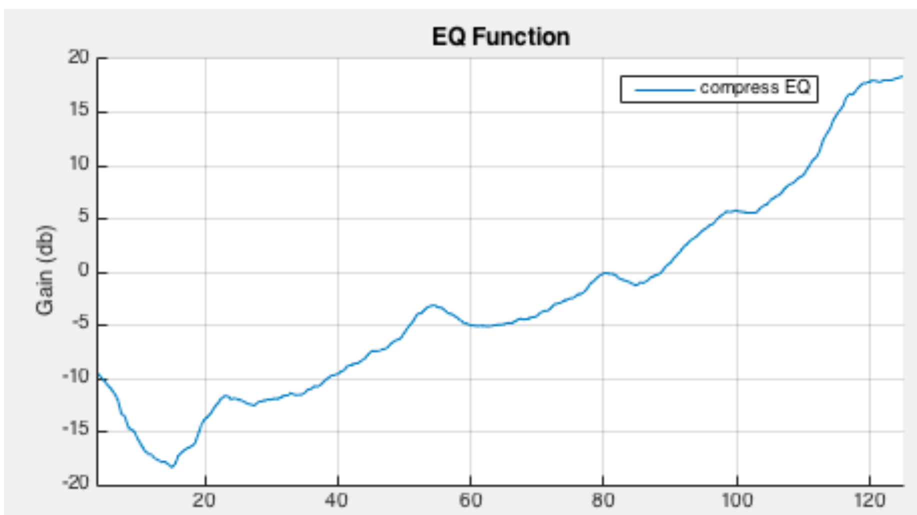
Finally, click the Build EQ button.

Some calculations will occur and the plots will be updated.

The Calibration Data plot will show an overlay of the smoothed calibration data in red:[2]



and the EQ Function graph will show the equalization curve:



[2] tip: if the legend on this or any other plots obscures the data, the legend can be repositioned by clicking and dragging the legend to a different location on the plot

BUILD THE FILTER:

After creating the EQ function, the equalization filter can be constructed. There are a few different settings available in the **Filter Settings** panel.

SAMPLE RATE (FS)

This must be set to the same sample rate as the stimuli that will be processed. Proper equalization will not be achieved if there is a mismatch!

IMPULSE RESPONSE AND FFT LENGTH (NFFT):

This sets the length of the filter impulse response. It must be a power of 2 (e.g., 512, 1024, 4096). Longer impulse responses are more computationally expensive. Short impulse responses might not decay to 0 by the end of the window, which will cause undesirable distortion in processed waveforms. The optimal length will be determined by: (1) the characteristics of the EQ Function, (2) the sampling rate, (3) the numbers of zeros and poles specified in the filter settings. Check the **Impulse Response** plot to see if it has decayed by the end to assess this.

ZEROS, # POLES

This specifies the characteristics of the filter. Empirical tests suggest that the default values (10 zeros, 20 poles) works for several calibration curves. Larger numbers will result in better fits to the desired EQ function at a cost of filter stability and increased round-off error.

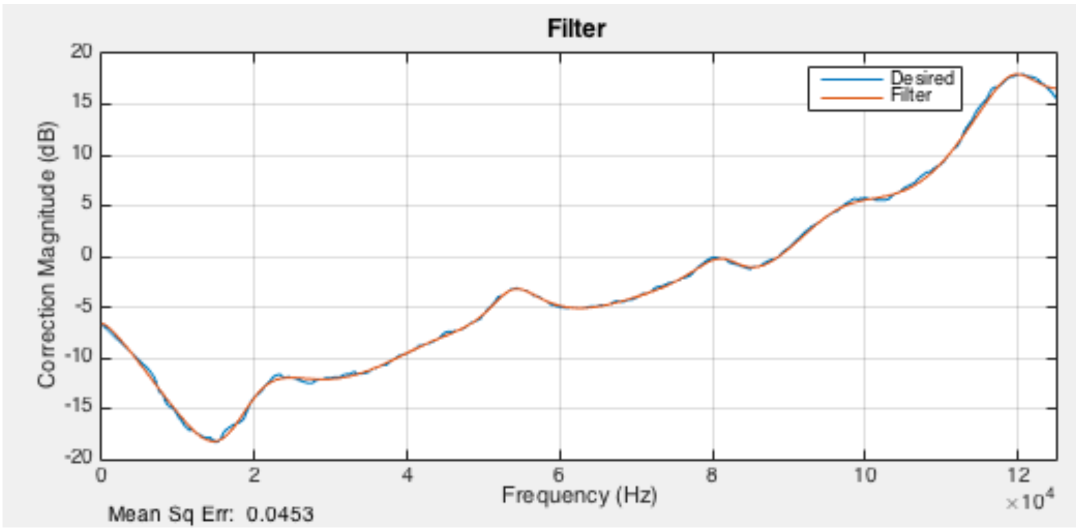
INTERPOLATION METHOD

The EQ function (from the calibration data) is, in general, not sampled at the same frequencies as the filter function (which will be at the stimulus sample rate, **Fs**). Also, the filter sampling frequencies go from 0 (DC) to the stimulus Nyquist frequency (Fs/2). As a result, the filter needs to be interpolated and, possibly, extrapolated at the high and low frequency regions. This sets the interpolation method used. In general, a linear interpolation works best, but other options are available for your convenience.

Build Filter:

After adjusting the filter settings, press the **Build Filter** button. After some calculations, the Filter and other functions will be plotted:

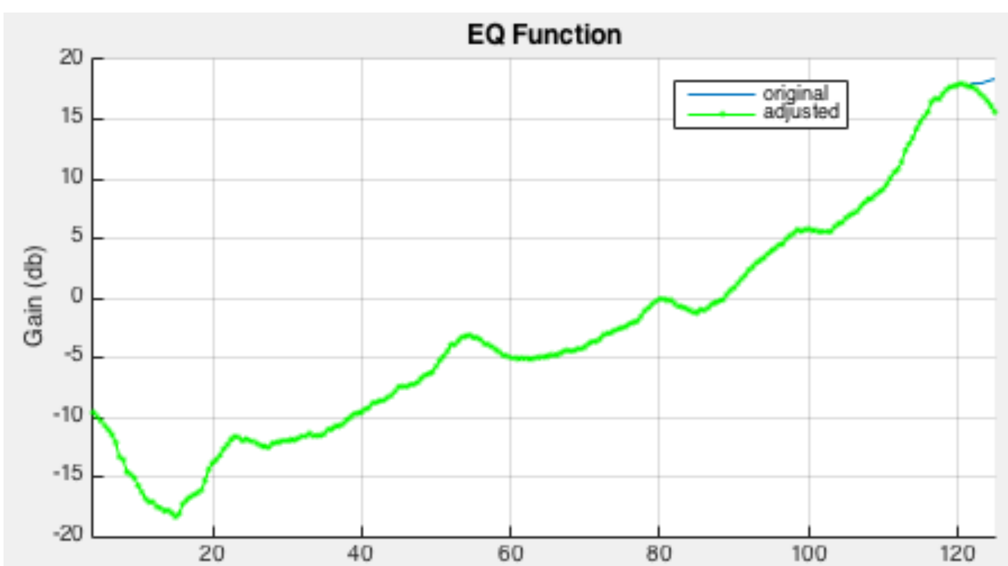
Filter



The **Filter** plot shows the desired EQ Function in blue and the calculated EQ Function in red. The goodness-of-fit is calculated as the mean-squared-error (MSE) between the two curves and is shown in the lower left corner ("Mean Sq Err"). Lower MSE is better – if the MSE is unacceptably high, try (1) increasing the number of zeros and/or poles, (2) increase the filter length (NFFT).

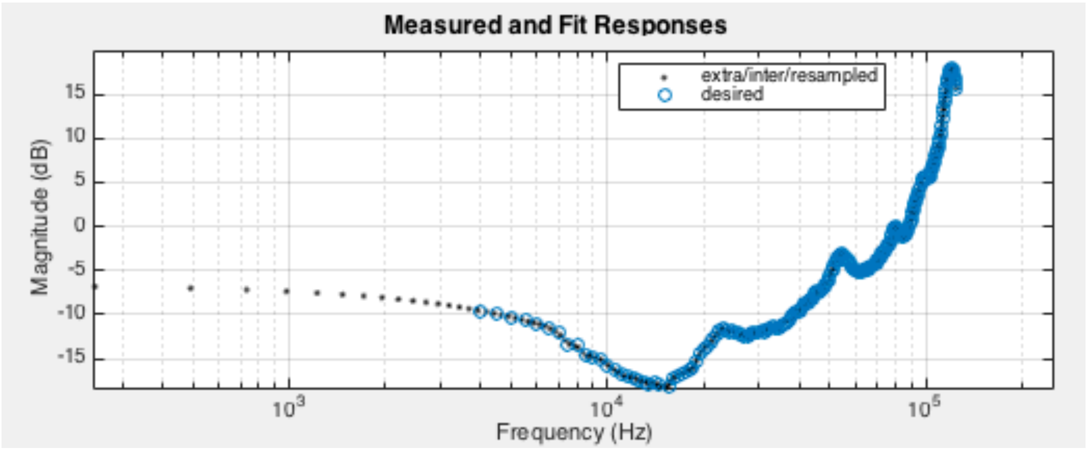
EQ Function

In order to avoid undesirable effects from extrapolation of the EQ Function at the low and high frequency ranges, the EQ Function is slightly adjusted. The adjusted value is overlaid in green on the EQ Function plot:



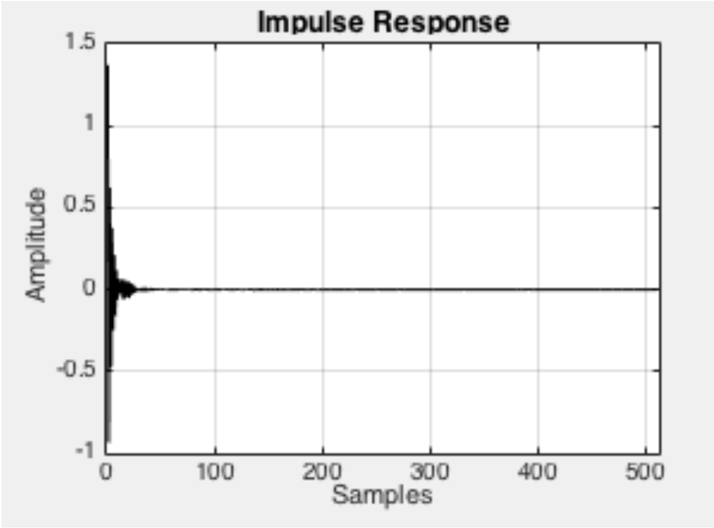
Measured and Fit Responses

To better see the effects of interpolation and extrapolation, the desired and fit responses are shown here on a semilog x-axis plot:



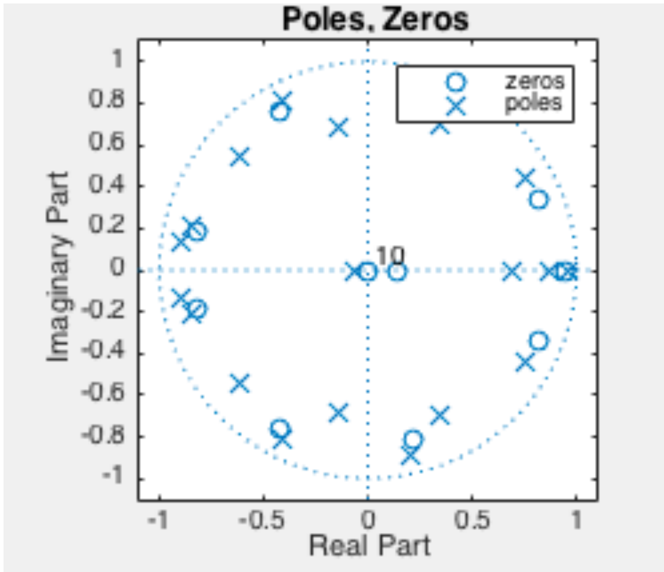
Impulse Response

The filter impulse response is plotted as a function of samples:



Poles, Zeros

Filter zeros (o) and poles (x) are shown here in a standard Z-plane plot. The unit circle denotes the region of stability (inside the circle) for the filter



SAVE THE FILTER:

Once the filter is to your liking, you can save it using the **Equalization** -> **Save EQ** menu.

EQUALIZING STIMULI

Select the Apply to WAV menu item. The program will ask you to select a .WAV file.

After selection of your file, the program will ask where to save the equalized file (and append '_eq' to the filename by default).

TESTING EQUALIZATION

For long duration stimuli, this is best done using another program such as Avisoft. Play your stimulus through the system, record the response, and examine the spectrogram to see if anything inappropriate has been done to the stimulus.

TO DO:

A program similar to FlatWav will allow measurement of output levels using a calibration microphone