DSP Correction of Impaired Musical Hearing

David McClain, Refined Audiometrics Laboratory

September 2023

Abstract

Musicians around the world are facing issues of hearing loss. Hearing aids attempt to recover speech perception, but perform poorly for recovering musical timbres. Musical hearing requires full-spectrum processing, and particular attention to the preservation of perceived harmonic overtones in order to recover the musical timbre of instruments. We describe a DSP implementation for very high fidelity perceptual recovery of musical sound for musicians with typical sensioneural hearing impairment.

Contents

1 Introduction

For many musicians around the world, the ravages of time and noise exposure, overloud performance venues, industrial noise from day jobs, riding overly loud motorcycles, participation in war zones, etc., has led to varying degrees of hearing loss. Their instruments no longer sound the way they used to. Former audiophiles have to give up in desperation. Most of the loss occurs in the high frequency range.

Maybe you have been cranking up the gain on that bright fuzz box to keep it sounding like it used to. But now your listeners are complaining about how crappy your sound is. Or else maybe you just go off into your memories while you play, re-living the old experiences - just like Beethoven had to do.

We seek to recover accurate musical perception of sound. To do so we need to pay particular attention to the recovery of musical harmonics, keeping them in proper proportion, so that instrument timbres are restored. We need full spectrum processing to get back as much of the audible spectrum as possible without added distortion artifacts.

Hearing aids attempt to recover speech perception. This is a different problem. One with narrower spectral demands, and often times intentional harmonic distortion actually helps. But they can do terrible things to music. They may turn an oboe solo during a Mozart Concerto into a Miles Davis muted jazz trumpet sound. Anachronistic! Noise cancellation erodes the bass drone beneath the music.

We start with the problem of wondering how things ought to sound? Our musical memory is famously short lived. So having some guidance would give us a target for correction. We know what we hear today, as our starting point. But what should it actually sound like? And secondly, how do you bend the sound so that we may hear that again?

We can't just use simple EQ. That might work at one loudness level, but then as soon as the music grows louder it knocks you out of your seat. Simple EQ can only ever be correct at just one loudness level. It becomes too little for fainter levels, and too much for louder levels.

Okay, so maybe let's try an audio compressor. That boosts low level signals but leaves the loud ones alone. This is better, but you need a different compression ratio, threshold, and makeup gain, on every frequency band. And when you listen carefully you find that there is no single compression ratio that works well at all loudness levels. It beats out simple EQ by being correct at two different loudness levels. But outside those levels it isn't enough, and inside it is still too much.

What we will show is that you need a ski-jump shaped nonlinear compression in every frequency band to make a proper restoration at all loudness levels and at every frequency.

2 Developing a Correction Target

Postulate the EarSpring Equation:

$$\left[\frac{d^2}{dt^2} + 2\beta\frac{d}{dt} + k\left(1 + \gamma\left\langle y^2\right\rangle\right)\right] y(t) = F(t)$$

EarSpring is a damped harmonic oscillator, modeling the whole of normal human hearing, whose spring stiffness grows with average power of vibration.

It is a nonlinear differential equation for the vibration amplitude due to a driving force varying with time. As the amplitude of vibration increases, its power increases as the square of the amplitude, and the spring grows stiffer. This shifts its natural resonant frequency higher. An auditory filter bank would sense the sound becoming flatter as lower frequency channels shift up toward the stimulus frequency.

It is important to emphasize that EarSpring is not a cochlear model. It is the simplest possible model for the whole of human hearing, which encompasses the cochlea, afferent 8th nerve, brain, and efferent 8th nerve systems. Our hearing is a huge nonlinear feedback control system. Cochlear models may someday become important. But just because you understand how a microphone works, doesn't mean you can know how the music will sound. That involves perception as well as physics.

The EarSpring model reproduces the phenomena of cube root loudness compression, subharmonic generation at loud levels, intermodulation distortion products, and pitch flattening with increasing presentation loudness, all while also satisfying measurable boundary conditions. If offers up a target for how things ought to sound.

The theory of nonlinear differential equations presents us with symmetry demands on the possible nonlinear behaviors. These allow only successive powers of $|y|^2$ in its series expansion. EarSpring includes only the lowest order nonlinear term, which suffices remarkably well for all hearing corrections. Higher order terms may well be present, but would only become important at extraordinary, non-musical, and possibly physiologically damaging, levels of sound intensity.

2.1 Frequency Independence

We should measure sound intensities, P, in phons. This expresses the sound intensity reaching the cochlear sensors in a frequency independent manner. Phons measure agrees with dBSPL measure (e.g., using loudspeakers and microphones) at 1 kHz, but differs in both zero point and scaling at all other frequencies.

Phons is what is presented to the EarSpring, after accounting for variations in mechanical coupling strengths at each frequency, and for the fact of auditory filtering due to the outer ear canal, and middle ear conduction efficiencies.

3 The Fundamental Equation for Hearing Corrections

$$\langle y^2 \rangle_{P+dP} - \langle y^2 \rangle_{P_{thr}} = \langle y^2 \rangle_P - \langle y^2 \rangle_0$$

At first glance, this may seem a surprising relation. The presented sound intensity needs to be increased to a level, $P + dP$, such that it produces a rise in vibration power above that of threshold level, said rise equal to the same rise above threshold for an unimpaired listener. Such a correction allows the impaired listener to hear the same experience.

And when properly applied in a frequency dependent manner, according to the degree of hearing impairment at each frequency, it preserves the unique timbres of instrumental sounds. Oboes continue to sound like oboes, and not become like muted jazz trumpets.

It seems surprising to us, as physicists and engineers. We are accustomed to using signal-to-noise ratios in detection theory and signal processing. But what, in our central nervous system could ever perform this kind of ratio calculation? Instead, we humans become inured to constant stimulation and notice only when it changes. We could envision a threshold mechanism among our neural networks that ignores stimulation unless it rises above that level, and then we get a sensation in proportion to the magnitude of the stimulus above that threshold.

Only this kind of power-additive behavior leads to the development of recruitment hearing (cf., Figure 1). Assuming a signal-to-noise ratio behavior leads to absurd conclusions about the amount of correction gain needed. The EarSpring equation describes the physics of vibration. But the fundamental hearing correction equation describes how our perception works, after the physics has produced its effects.

We have already talked about a unit of perception called Phons. We can't measure such things in the laboratory. We can only determine the Phons level by asking other humans about what they hear. The real physical world is stuck using microphones and loudspeakers and voltage meters. The lab uses dBSPL to measure sound intensity, and that is surely related to Phons, the perceptual measure. But signal-to-noise ratios belong to physics, not to perception. Sound power differences appear to belong to perception.

So here we are mixing the two universes - physical and perceptual, and using equations to model their relation to each other.

Make the identification of the vibration power ratio with Sones measure:

$$S(P) \equiv \frac{\langle y^2 \rangle_P}{\langle y^2 \rangle_{40}}$$

for sound intensity, P, phons. Then the fundamental equation can be expressed as:

$$S(P + dP) = S(P) + (S(P_{thr}) - S(0)) \approx S(P) + S(P_{thr})$$

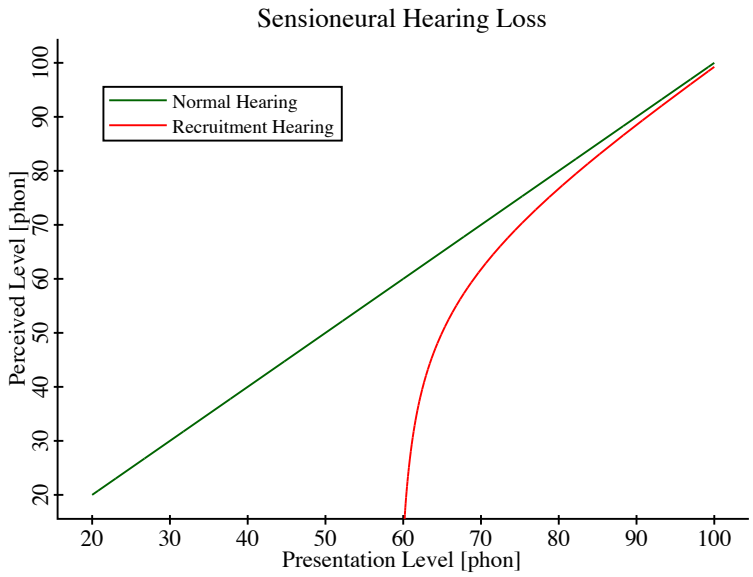

Figure 1: *Normal v.s. Impaired (aka Recruitment) Hearing. Normal hearing perceives that which you present. But damaged hearing quickly loses perception as presentation declines in loudness. This happens by varying degrees at each presentation frequency. The damage is described in terms of an elevated threshold of hearing - in this case 60 phon.*

since, typically, $S(P_{thr}) \gg S(0)$.

Armed with the fundamental equation for hearing correction, we can compute correction gains for each presented sound intensity, so that we can rectify the recruitment hearing due to hearing loss. It forms a nonlinear compressor, specific to each frequency and the degree of impairment at that frequency. (cf., Figure 2)

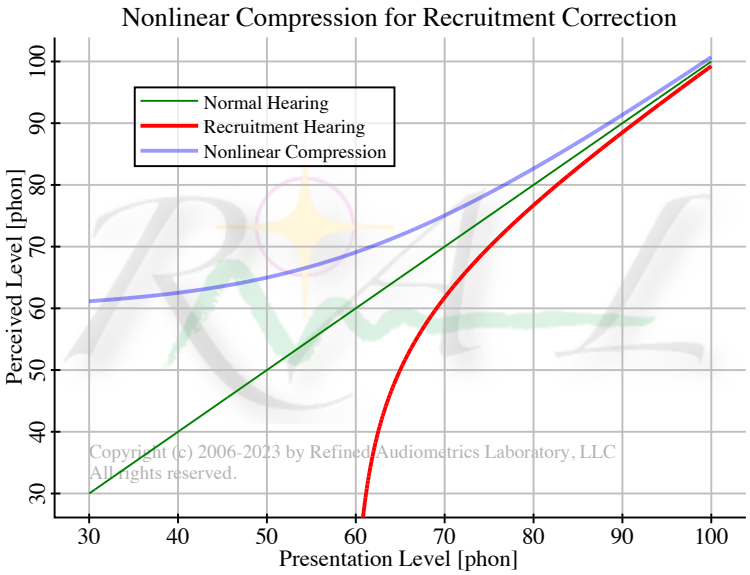

Figure 2: *Nonlinear compression can be used to rectify recruitment hearing. It is the mirror image of the recruitment curve, reflected across the diagonal representing normal hearing.*

4 Solving the EarSpring Target Equation

For hearing corrections we need to solve the fundamental correction equation for the necessary signal gain, dP, in Phon space, which can be related back to the its frequency dependent gain in SPL space. All corrections must ultimately be performed in SPL space.

But to do that for the fundamental correction equation, we need to know what Sones level is present. You can't measure it with a meter. But EarSpring furnishes us with a way to derive it from measurable dBSPL sound levels.

4.1 Before We Dive Into That...

Getting to the solution could be arduous. How about we try something quick?

Simple approximations to within 1.5% relative error above 36 phon:

$$S(P) \approx 10^{(P-40)/30}$$

$$P(S) \approx 40 + 30 \log_{10}(S)$$

These approximations assume cube-root compression at all loudness levels. Not true, but pretty good until you reach significantly below 40 phon. Everyday sounds are mostly above 40 dBSPL. You can only reach deep below in a sound isolation booth. You probably won't be able to hear a whisper at 20 dBSPL while riding on the subway, unless it is very close to your ear and also somewhat louder than usual.

A simple correction gain approximation, which performs within 3 dB of correct behavior down to whisper-soft presentation levels of 20 phon, for even the most extreme correctable threshold elevations, is:

$$dP \approx 30 \log_{10}\left(1 + 10^{-(P-P_{thr})/30}\right)$$

High levels of threshold elevation produce recruitment hearing which has very steep slopes near their threshold levels, (cf., Figure 1). Any small errors or variations in the correction gain produce exaggerated effects for small signals in after-correction perception. For making hearing corrections we are condemned to attempting open-loop control of a high-gain system. So high accuracy in correction gains is desired.

But it should be apparent now, after looking at the shape of that recruitment hearing curve, that a rectifying compression curve will end up looking very much like a ski-jump, not a straight line. Just reflect that recruitment curve across the diagonal which represents normal hearing.

4.2 Rolling Up Our Sleeves

Along with the use of Phons for frequency independence, we also define a frequency normalized damping constant, $\hat{\beta} \equiv \beta/\sqrt{k}$. Doing this will make our equations apply equally well for all signal frequencies. This is also equivalent to assuming constant-Q auditory filters, which is mostly true in the treble region above 500 Hz, where most sensioneural hearing damage occurs.

Resonant frequencies are:

$$\omega_P = \pm\sqrt{\left(1 + \gamma\ \langle y \rangle_P^2\right) - \hat{\beta}^2}$$

The pitch flattening ratio can be written as:

$$F_{90}^2 = \left(\frac{\omega_{90}}{\omega_{40}}\right)^2 = \frac{1 + \Gamma_{40}\, S_{90} - \hat{\beta}^2}{1 + \Gamma_{40} - \hat{\beta}^2}$$

where factor Γ_{40} represents the contribution to detuning at 40 phon, due to the γ stiffness term in EarSpring:

$$\Gamma_{40} = \gamma\ \langle y \rangle_{40}^2$$

Factor S_{90} represents the power ratio between vibrations resulting from sounds at 90 phon, compared to those resulting from sounds at 40 phon:

$$S_{90} = \frac{\langle y \rangle_{90}^2}{\langle y \rangle_{40}^2}$$

With P as Phons, we make the identification of the power ratio, S_P, with Sones. In general:

$$S_P = 10^{(P-40)/10} \cdot \frac{4\hat{\beta}^2 + \Gamma_{40}^2}{4\hat{\beta}^2 + \Gamma_{40}^2\, S_P^2}$$

Pitch flattening happens principally from the γ stiffness term in EarSpring. There is a second order downward shift resulting from the damping in the system, $\hat{\beta}$. The half-power bandwidth of the system is $2\,\hat{\beta}$. The system Q-factor is $1/(2\,\hat{\beta})$ if we refer to the undamped natural frequency, $\sqrt{1 + \gamma\ \langle y \rangle_P^2}$, in its definition.

We have boundary conditions: the laboratory measured degree of tone pitch flattening as loudness increases from 40 to 90 phon, and the absolute threshold of hearing expressed in sones:

$$\begin{aligned} F_{90} &\approx 75\,\text{cents} \rightarrow 1.044 \\ S_0 &= (1/22)^2 \approx 0.002066\ \text{sones} \end{aligned}$$

The pitch flattening ratio, expressed as the ratio of EarSpring resonant frequencies, corresponds to our measured flattening of 75 cents. As will be seen, the exact measure of flattening will be relatively unimportant. It exists, but its effects are second order except at very low loudness levels. It primarily affects the value found for the stiffness term, Γ_{40}, and the damping constant, $\hat{\beta}$.

Solve this system of equations for $\hat{\beta}$, Γ_{40}, and S_{90}:

$$\Gamma_{40} = \frac{F_{90}^2 - 1}{S_{90} - F_{90}^2} \cdot \left(1 - \hat{\beta}^2\right)$$

$$S_0 = 10^{-4} \cdot \frac{4\hat{\beta}^2 + \Gamma_{40}^2}{4\hat{\beta}^2 + \Gamma_{40}^2\, S_0^2}$$

$$S_{90} = 10^{5} \cdot \frac{4\hat{\beta}^2 + \Gamma_{40}^2}{4\hat{\beta}^2 + \Gamma_{40}^2\, S_{90}^2}$$

Derived Values:

$$\begin{array}{rcl} \hat{\beta} & \approx & 0.0002214 \\ \Gamma_{40} & \approx & 0.001963 \\ S_{90} & \approx & 47.19 \text{ sones} \end{array}$$

The effect of varying F_{90} over a broad range, from 25 to 125 cents, leaves the value found for S_{90} unchanged to 4 significant figures, while moving the obtained $\hat{\beta}$ from 0.00007 to 0.00038. However, in terms of correction gains developed at these F_{90} extremes, they show a $< 10\,\mu$phon variation for a situation with $P = 30\,$phon and $P_{thr} > 60\,$phon. An immeasurable difference.

a cubic equation in S_P. The solution looks like you see in Figure 3 This can be solved for closed-form solutions for either Sones in terms of phons, or Phons equivalent to some level in sones:

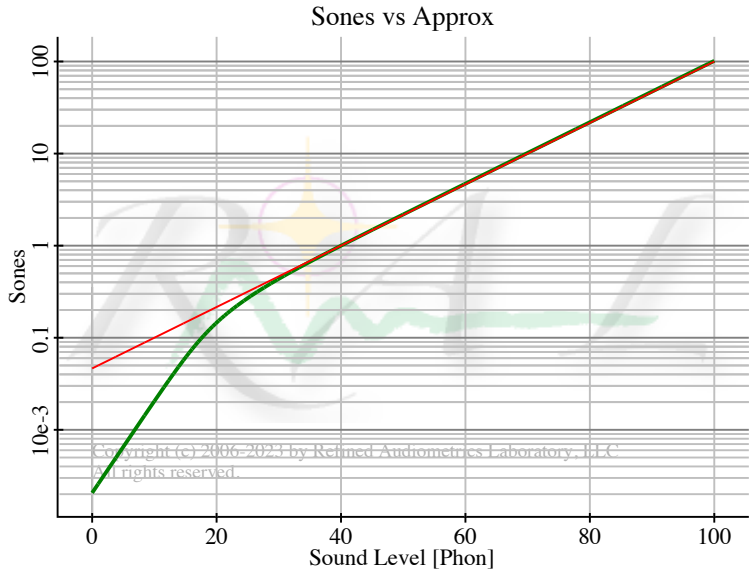

Figure 3: *The solution of the EarSpring equation, $S(P)$, and its simple approximation. Linear behavior near threshold levels, and cube-root compression above 40 phon.*

$$
\begin{aligned}
\text{Sones}(\mathit{phons}) \quad &= \\
\hat{p}^2 \quad &\leftarrow \quad 10^{(\mathit{phons}-40)/10}, \\
\Phi \quad &\leftarrow \quad 3\sqrt{3}\left(\Gamma_{40}^2\right)^2\left(4\hat{\beta}^2+\Gamma_{40}^2\right)\hat{p}^2, \\
\Lambda \quad &\leftarrow \quad \left(\Phi+\sqrt{\Phi^2+4\left(4\hat{\beta}^2\Gamma_{40}^2\right)^3}\right)^{1/3}, \\
& \qquad \frac{\Lambda}{\sqrt[3]{2}\sqrt{3}\,\Gamma_{40}^2}-\frac{\sqrt[3]{2}\left(4\hat{\beta}^2\right)}{\sqrt{3}\,\Lambda}
\end{aligned}
$$

$$
\begin{aligned}
\text{Phons}(\mathit{sones}) \quad &= \\
\hat{p}^2 \quad &\leftarrow \quad \mathit{sones}\cdot\left(\frac{4\hat{\beta}^2+\Gamma_{40}^2\,\mathit{sones}^2}{4\hat{\beta}^2+\Gamma_{40}^2}\right), \\
& \qquad 10\log_{10}(\hat{p}^2)+40
\end{aligned}
$$

Phons is readily seen as a mixture of linear and cubic behavior in sones. Above 40 phon (= 1 sone), the cubic behavior dominates. At absolute threshold levels, the linear behavior dominates. But the mixed behavior range, from 10 to 40 phon, is important for proper hearing corrections when damage threshold levels are high.

The simple approximations shown earlier, which assume purely cube root compression

at all loudness levels, become too simple in the face of steep recruitment hearing. They overestimate the Sones level by a significant amount across the mixed behavior zone. When used as the basis for developing corrective gains, such gains are too large for weak signals, making sonic details in decaying reverb tails become harsh and crunchy sounding. High thresholds lead to very steep recruitment curves, which demand precision better than 0.1% in the offered hearing correction gains.

Substituting numerical values for the constants:

$$\begin{aligned}
\text{Sones(phons)} &= \\
\hat{p}^2 &\leftarrow 10^{(phons-40)/10}, \\
\Phi &\leftarrow 3.127\text{e-}16\,\hat{p}^2, \\
\Lambda &\leftarrow \left(\Phi + \sqrt{\Phi^2 + 1.726\text{e-}36}\right)^{1/3}, \\
& \quad 118900\,\Lambda - \tfrac{1.426\text{e-}7}{\Lambda} \\
\text{Phons}(sones) &= \\
\hat{p}^2 &\leftarrow sones \cdot \left(0.04840 + 0.9516\,sones^2\right), \\
& \quad 10\log_{10}(\hat{p}^2) + 40
\end{aligned}$$

From these, see that 40 phon $\rightarrow$ 1 sone, and 1 sone $\rightarrow$ 40 phon, as required.

With these equations, we can now find the required hearing compensation gain, dP, for any sound level, P, and threshold elevation, P_{thr}, as:

$$dP = \text{Phons}\left(\text{Sones}\left(P\right) + \left(\text{Sones}\left(P_{thr}\right) - \text{Sones}\left(0\right)\right)\right) - P$$

5 Degrees of Damage

It might be tempting to conclude that if you have profound hearing loss, with threshold elevations above 90 dBHL, then you must have lost all your hearing sensors at that frequency. But that conclusion is unwarranted. People with profound hearing loss can still hear very loud sounds.

Consider how our nervous system responds. Touch your finger on a hotplate and the reaction is to immediately withdraw in searing pain. That pain dominates your imagination for some while, but it gradually dissipates to tolerable levels. The brain begins to ignore the constancy of the pain.

Now imagine that hearing loss is caused by some damaged hair cells. Those hair cells could be imagined to be putting out a screaming loud signal for themselves, equivalent to some sound level, P_{dam}. But they are surrounded by other hair cells for the same frequency, which are not damaged.

The brain sets its threshold to ignore the constancy of those damaged screams. What fraction, f, of hair cells need to be damaged in order to have an elevated threshold, P_{thr}?

$$fS(P_{dam}) = (1-f)S(P_{thr})$$

so

$$f = \frac{1}{1 + S(P_{dam})/S(P_{thr})}$$

Any reasonable value could be assumed for P_{dam}. How about using the threshold of pain, 120 phon?

What fraction of hair cells need to be damaged in order to produce profound hearing loss with a threshold of 90 phon?

$$f = \frac{1}{1 + S(120)/S(90)} \approx 9\%$$

So, far from complete annihilation of hair cells at that frequency. 91% of hair cells can still respond. It's just that in order to help those hair cells recover some sense of normal hearing in the midst of all the screams, it takes so much correction gain that we risk damaging them.

But for any other threshold elevations, we should modify the fundamental equation of hearing correction to read:

$$(1-f)\left(S(P+dP) - S(P_{thr})\right) = S(P) - S(0)$$

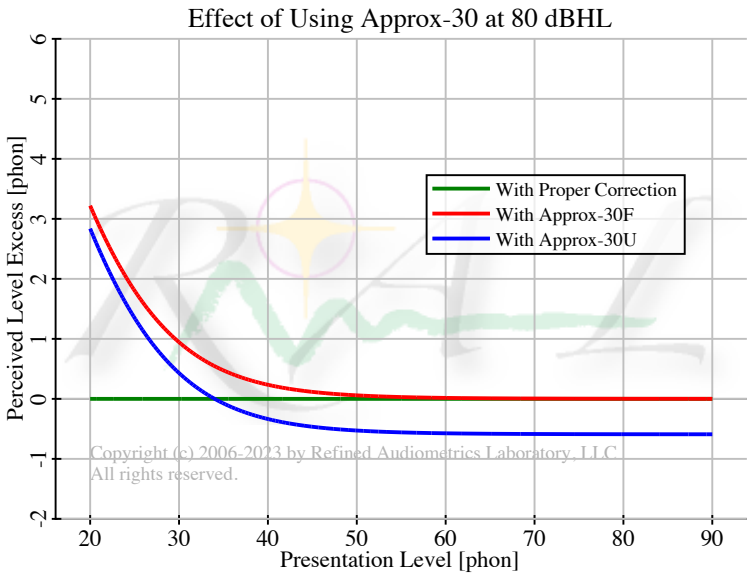

Figure 4: *Post-correction error in using approximation vs. proper corrections for an extreme threshold elevation. Here, Approx-30F is the approximation including effects of live hair cells remaining. Approx-30U is the approximation presented originally, which did not consider hair cell damage.*

5.1 An Even Better Approximation

For use in our correction computations we prefer to use the fraction of live hair cells instead of the fraction of dead ones. For that, we have:

$$f_{live} = \frac{1}{1 + S(P_{thr})/S(P_{dam})}$$

And we make our correction equation read:

$$S(P + dP) = \left(S(P) - S(0)\right)/f_{live} + S(P_{thr})$$

In the approximation, this becomes:

$$dP \approx 30 \log_{10}(1/f_{live} + 10^{-(P-P_{thr})/30})$$

A comparison between using the approximation and proper corrections is shown in Figure 4. The post-correction errors are not too serious. As will be shown, we have more to fear from inadequate threshold estimation than what errors we see here. Still, using the correct solution is not that cumbersome in comparison to the approximation.

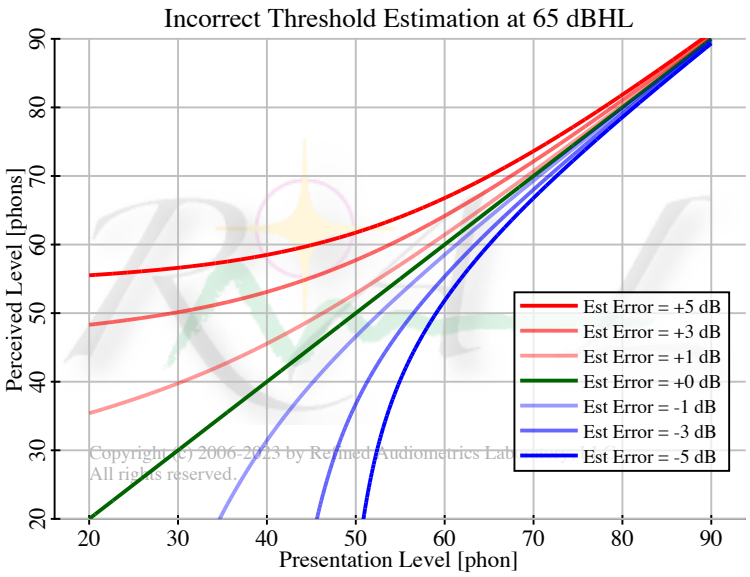

Figure 5: *Post-correction effects from using incorrectly estimated threshold elevations for a moderately severe impairment. The top curve shows what happens if you measured 70 dBHL but actually only had 65 dBHL. Blue curves are for the converse situations.*

5.2 Errors in Practice

Figure 5 shows what happens if we mis-measure the threshold elevation. Since audiology claims accuracies of ±5 dBHL, you can see how inadequate that would be for our purposes. And as threshold elevation increases it gets more extreme.

But the good news, for moderately severe and greater impairment, is that you should be able to vary the threshold slider in increments of 1 dB and petty clearly discern where the correct elevation is, during your own listening tests.

And as you can see, at quiet levels, even a 1 dB mis-measure of threshold elevation swamps the modest overcorrection coming from using either approximation.

If audiometry were precise, then you would probably never underestimate a threshold elevation because you simply wouldn't hear the test tone until it crosses above threshold. Testing typically proceeds in increments of 5 dB. Over-estimation would be more likely to happen as the tone suddenly pops into audibility just above threshold, using such coarse probing. But if the audiometer has a calibration error, then either outcome is possible.

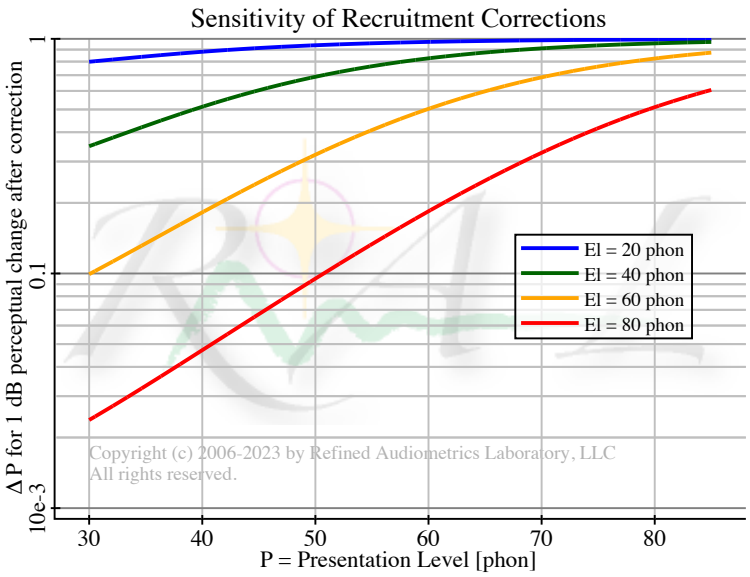

Figure 6: *Sensitivity of extreme recruitment hearing to small changes in correction gains.*

6 Hearing With Large Threshold Elevations

Hearing with threshold elevations below 20 dBHL is considered normal hearing. dBHL is an SPL-space measurement, with the same scaling as dBSPL, but uses a frequency dependent zero point which follows along the absolute threshold of hearing (ATH). It measures the elevation above ATH in laboratory measurable dB units.

Damaged hearing begins at threshold elevations above 20 dBHL. It can range from mild impairment all the way up to profound impairment at levels above 90 dBHL. It becomes impossible to offer corrections for profound impairment, as the necessary correction gains exceed maximum safe power levels, and could induce further physiological damage.

As mentioned, large threshold elevations present challenges to gain corrections for low signal levels. Such high threshold elevations have very steep recruitment curves near their threshold. The graph in Figure 6 shows, in absolute measure, how much change in correction gain is required to produce a post-correction change of 1 dB in perceived loudness, for presentation levels from 30 phon to 85 phon.

For example, at 30 phon, an 80 phon threshold elevation requires only 0.025 phon change in correction gain to produce what sounds like a 1 dB change in corrected loudness. The overall correction gain needed here would be 50.27 phon. So this is less than 0.05% change

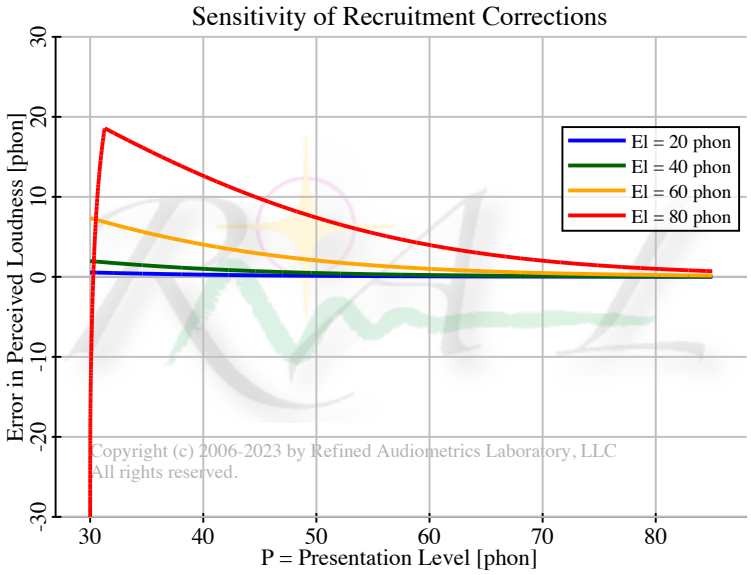

Figure 7: *Overestimated threshold elevations by 1 phon. Supplied corrective gains become too large. The sharp decline at 30 phon for the case of 80 phon elevation is due to imposed gain limits in the interest of ear safety.*

in gain.

But let's assume we can both measure dynamic sound levels, and do the math, accurately. What happens if we incorrectly estimate the threshold elevation by as little as 1 phon?

Pure tone audiometry has self-reported accuracies of ± 5 dB. At frequencies with extreme threshold elevations this becomes a relatively easier binary decision by the listener - either you hear the test tone or you don't, with little doubt either way.

If we overestimate thresholds by just 1 phon then those channels with extreme elevations will be getting too much corrective gain, as shown in Figure 7. A signal dithering around the musical noise floor will appear to go from absent to significantly louder than it ought to seem after correction, inducing a kind of stuttering or crunchy chatter for the noise floor.

Conversely, if we underestimate the threshold elevation by this much then we have the situation in Figure 8. In this case it is doubtful you would ever be able to hear the musical noise floor in the corrected channels with elevations much above 50 phon. And audiology testing should never be able to underestimate.

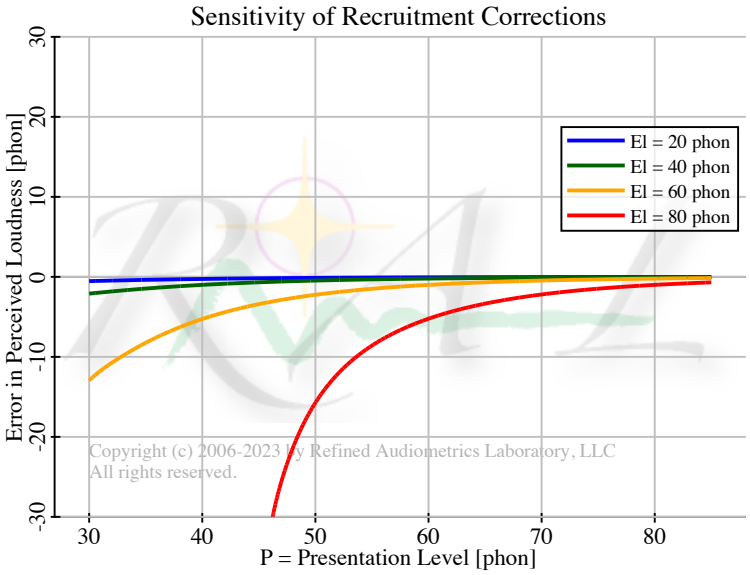

Figure 8: *Underestimated threshold elevations by 1 phon. Supplied corrective gains become too small.*

These figures illustrate the reason for my skepticism about accurately determining a high threshold elevation with pure tone audiometry. Stepping the tone by increments of 5 dB leaves much room for overestimation.

During corrective fitting there will be a sharp distinction between excessive background noise chatter and none at all, as you make small changes in estimated threshold elevations, in the worst afflicted frequency bands. So this indicates a possible method for more accurately finding the correct threshold elevation: simply decrease the elevation in steps of 1 dBHL until the noise chatter suddenly quiets. At that point you are within a fraction of 1 dBHL of the correct elevation.

7 Bark Frequency Channels

All of the above equations apply to only one frequency channel at a time. They all depend on knowing the sound power present in the frequency channel.

Our hearing is not channelized with fixed-frequency channels. Rather, our hearing establishes self-organized frequency bands surrounding the loudest frequency components in the sound. Bands of critical bandwidth can be defined as channels wherein: strong signals within one critical band mask out the weaker components in that same band, and partially mask sound in other nearby critical bands in a diminishing manner.

Critical bandwidths correspond reasonably well with bands of Bark frequency, denoted as z_{Bark}. (cf., Figure 9) These are nearly constant-Q bands in the higher frequencies, and exhibit increasing absolute bandwidth as frequency grows. Below 500 Hz, they seem to exhibit fixed bandwidths of about 100 Hz. The entire spectrum of audible sounds, ranging from 20 Hz to 20 kHz, is covered by 25 Bark bands. This is shown in Figure 10.

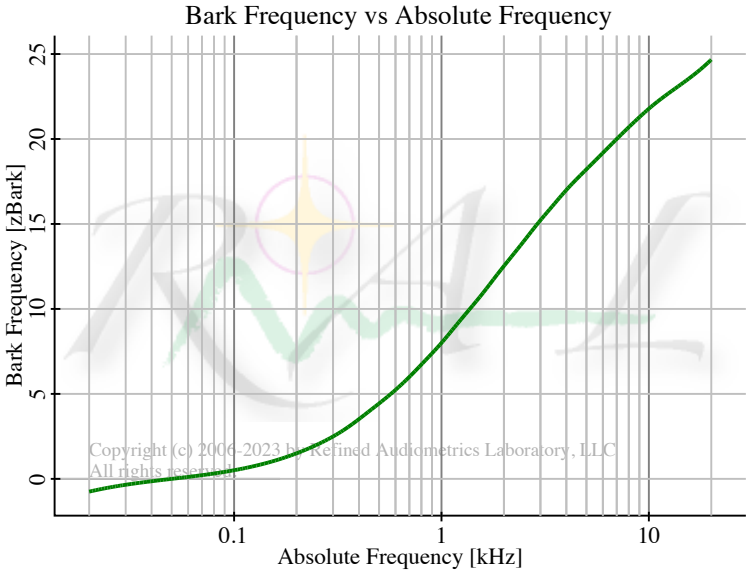

Figure 9: *How Bark frequency relates to the absolute frequency measure. Below about 500 Hz, Bark channels exhibit a nearly fixed width of 100 Hz. Above that they are related in a logarithmic manner.*

7.1 A Decent Compromise

Despite our hearing forming self-organized Bark channels, we can develop astonishingly good hearing corrections using only a relatively few fixed-frequency Bark channels. While

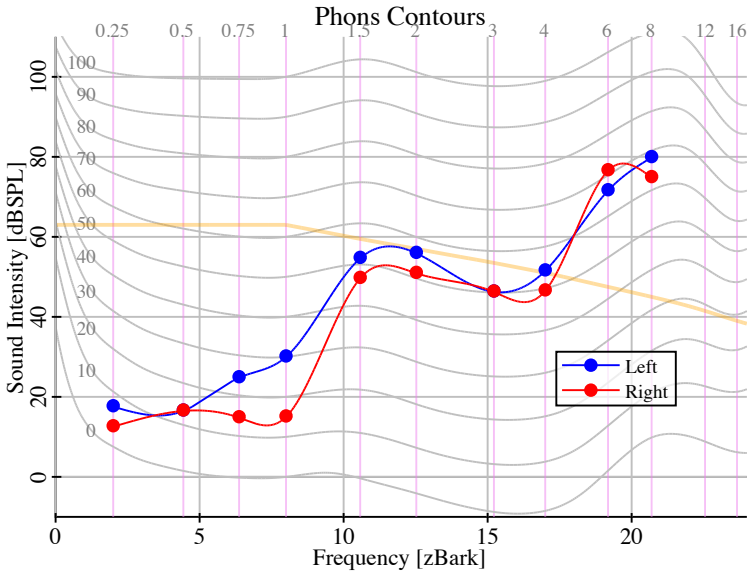

Figure 10: *Equal loudness contours in Bark frequency space. Notice how completely the audiology testing frequencies span the audible spectrum. But, perceptually, the entire spectrum of sound below Middle-C is crammed into the narrow left region below* $2\,z_{Bark}$.

our pitch perception is quite acute, our ability to discriminate loudness with respect to frequency is much more coarse. That's why a 2- or 3-band EQ suffices much of the time[1].

The standard audiology test frequencies, $\{250\,Hz, 500, 750, 1\,kHz, 1.5, 2, 3, 4, 6, 8\}$, at half-octave spacings, correspond to Bark channels, each about $2\,z_{Bark}$ wide. For hearing corrections, these two-Bark bands work very well.

A crude approximation is that for channels away from the frequency of the strongest component, the masking in higher frequency channels declines at the rate of $10\,dB/z_{Bark}$. For channels below the dominant frequency, the masking declines about $20\,dB/z_{Bark}$.

This is just the backwards view of an auditory filter. Stronger masking above means that an auditory filter has a gentler rolloff to frequencies below.

One can develop a Bark channelizing filter by using an asymmetric triangular Bark kernel in dB space, with these rolloff, and convolve that across a rectangular bandpass covering the frequency range between Bark channel boundaries. (cf., Figure 12)

Band edges can be specified as the geometric mean of adjacent band center frequencies. So

[1] *But not for hearing corrections!*

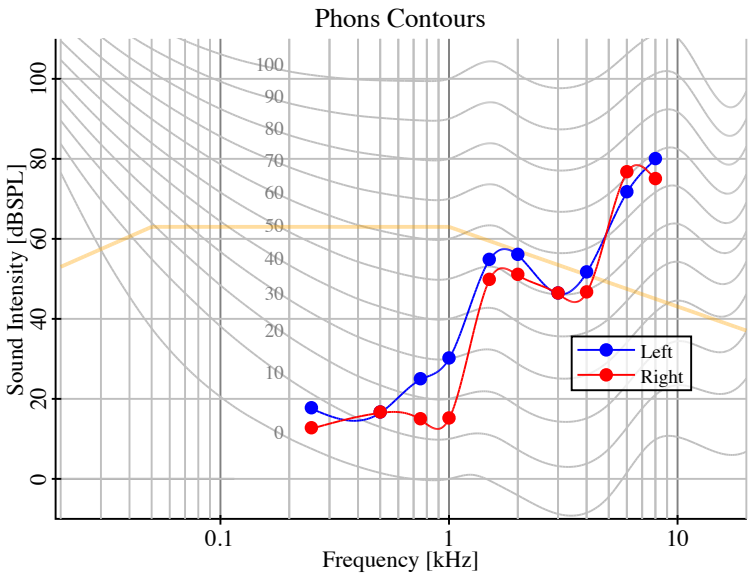

Figure 11: *Equal loudness contours as typically shown in Hz frequency space. How inappropriate might this be?*

for the 1 kHz Bark channel, the lower edge is at $\sqrt{750 \cdot 1000} = 866\,Hz$, and the upper edge is at $\sqrt{1000 \cdot 1500} = 1225\,Hz$. We map those over to Bark frequency space to construct our channel filters. The filters are all normalized to unit peak amplitude in Bark space. (cf., Figure 13)

Using the geometric mean is appropriate here because Bark frequency has a nearly logarithmic relation to absolute frequency, at the higher frequencies. A geometric mean in absolute frequency measure becomes a simple arithmetic average in Bark frequency measure.

7.2 What Are We Actually Measuring?

Since our hearing is channelizing into critical bands, the gain correction must apply to those individual bands. We don't want to tell the gain estimator the total power across our full $2\,z_{Bark}$ channel bandpass. Our Bark channels are wider than one critical bandwidth, which we take to be $1\,z_{Bark}$ wide.[2] We want to tell it how much was seen in each critical band.

[2]Some might argue that critical bands are a bit narrower than 1 z_{Bark} wide, preferring to use the ERBS measure of Moore and Glasberg 1990. These are about 60-80% of Bark width, depending on frequency. In such case one needs to reduce the power per critical band estimated here by those factors.
This has the effect of us making slightly stronger corrections, suggesting that threshold elevation settings could be reduced by a few dBHL.

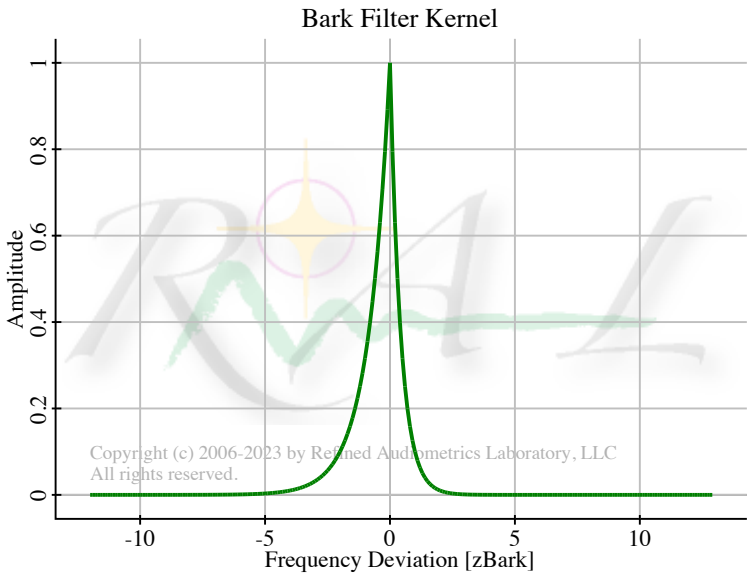

Figure 12: *Convolutional filter kernel for Bark channel filters. It has a rolloff of 10 dB/zBark below, and 20 dB/zBark above, the center frequency.*

The power we observe in a Bark channel is:

$$P_{tot} = \int_0^\infty F_c(z)^2 P(z)\, dz$$

for power density, $P(z)$, and filter amplitudes, $F_c(z)$. Dividing this by the number of critical banks in the channel tells the corrections gain processor how much correction they need. Using a single correction gain assumes they all have the same need.

A constant unit magnitude power density spectrum across a Bark channel will show an integrated power equal to:

$$ERB = \int_0^\infty F_c(z)^2 dz$$

This defines the equivalent rectangular bandwidth, ERB, of the channel filter. A unit power density across a rectangular, unit height, filter of this width, would produce the same measured power. The ERB width, in z_{Bank}, expresses how many critical bands are being managed by the channel.

By dividing the total power measured in the channel by the effective channel width, we furnish an average power across the channel filter. The critical bands contained within each Bark channel will be treated equally using this average power for each of them.

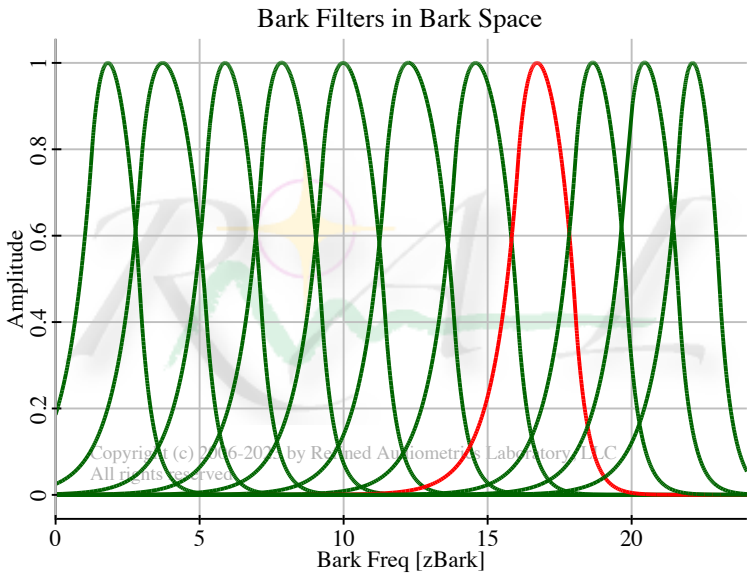

Figure 13: *A Bark channel filter bank displayed in Bark space. The 4 kHz band is highlighted in red, and has a center frequency of 17 z_{Bark}. For a view of how these appear in FFT space, see Figure 14)*

So we want to report:

$$P_B = \frac{\int_0^\infty F_c^2(z)\, P(z)\, dz}{\int_0^\infty F_c^2\, dz}$$

with all integrals performed in Bark frequency space.

The corrective gains obtained from this measurement will be a bit too large for some critical bands, and a bit too small for others. The extent of this mismanagement depends on the slope of hearing impairment across the channel, and on what spectral slope, in Bank space, is presented by the signal across the channel bandwidth. A single channel cannot discern any better than their average effects across all the critical bands within.

We will discuss how we can perform these integrals directly in the FFT space, using adapted filters, $\hat{F}(f)$, in absolute frequency space. The filters and their ERBs can be pre-computed. And so our power estimator should compute:

$$P_B = \frac{\sum_i F_i^2 P_i \Delta z_i}{\sum_i F_i^2 \Delta z_i} = \frac{\sum_i \hat{F'}_i^2 P_i}{\sum_i \hat{F'}_i^2}$$

where index, i, ranges over the bins of the FFT spectrum.

Once we have P_B we are ready to convert into the space of the fundamental hearing correction equation and compute a correction gain.

8 Measuring Bark Channel Power

As to the assumption that we can accurately measure dynamic sound power levels in each frequency band, we face an engineering dilemma.

If we make the frequency channels too sharp and abrupt at channel boundaries, then we are likely to suffer a zipper ratchet of gain as a chirped signal crosses many channels.

A solution might be to use many narrow Bark channels, each with smaller expected inter-channel gain differences. This could more accurately depict the self-organized channels. But then the computational load increases, and we'd like an inexpensive realtime correction system.

Our sense of loudness channelization is actually too crude to warrant the extra effort. And then there will happen an isolated pure sine tone, like a solo violin, making a chirped trip through the channels. Despite our attempt to diminish the inter-channel gain steps, we still get a stepped response, going from full off to full on, as it crosses into the next channel, and probably unwelcome artifacts in the corrected sound.

Conversely, if we allow too much inter-channel bleed, any strong signals in an adjacent channel will imply too much signal apparently present in our channel - starving our channel of the correction gain it actually needs. If we can compensate for the bleed, then we can use fewer bands and have much lower computational load for realtime correction. We avoid the gain ratchet by gracefully transitioning across band boundaries. But how much bleed is too much? And how to compensate?

8.1 What Does a Filter Really Do?

We have to operate in FFT space, not directly in Bark space. So we have to ask how we can make an equivalent filter in another frequency space? And what does a a filter actually do?

The way we map from a filter in one space to another, is to follow the grid in the target space and perform an inverse mapping from there back to the original space. This is the same way you map stretched images in graphic displays. Going the other way would inevitably leave gaps in the target space.

Each bin of the FFT target space has fixed frequency width: $\Delta f = F_{samp}/N_{blk}$. The i^{th} bin covers from $f_{i,<} = (i - 1/2)\Delta f$ to $f_{i,>} = (i + 1/2)\Delta f$. It is easy enough to map those band edges over to Bark space[3] to find the equivalent $z_{i,<}$ and $z_{i,>}$. But it would be a

[3] We use a cubic spline interpolation over the published Bark channel center frequencies:
(50 Hz 150 250 350 450 570 700 840
1 kHz 1.17 1.37 1.6 1.85 2.15 2.5 2.9
3.4 4 4.8 5.8 7 8.5 10.5 13.5 20.5)

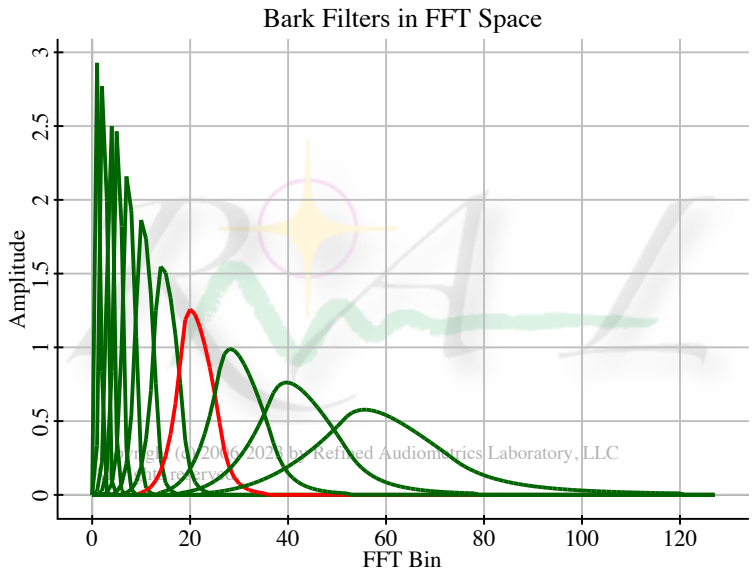

Figure 14: *A Bark channel filter bank on the FFT. The 4 kHz channel is highlighted in red. The amplitudes have been mapped to account for the varying width of each FFT bin as measured in Bark frequency, z_{Bark}. (cf., Figure 13) Notice too, how the rolloff appears in the wrong sense - another artifact of the frequency mapping.*

mistake to simply look up the amplitude of the Bark channel filter at the mid frequency between those two edges and carry that value over to the FFT domain.

A filter is used to find a weighted average power in some bandpass. What we will ultimately want to compute is:

$$P_B = \frac{\int_0^\infty F(z)^2 P(z) dz}{\int_0^\infty F(z)^2 dz}$$

with the integrals performed in Bark frequency space. The result, P_B, expresses power per unit z_{Bark}, where $P(z)$ is the power density over the passband, and $F(z)$ is our filter amplitude. Filters act on signals. Power is the square of the signal. The numerator is the integrated filtered signal power over the passband, and the denominator is the ERB of the filter.

So in our expression for P_B a filter acts as a weighting function on the power within the bandpass to give us a weighted average power per unit of frequency in the channel.

We have to obey conservation of energy. So the integrated power over a single FFT bin, weighted by the Bark channel filter, has to be the same no matter which space performs

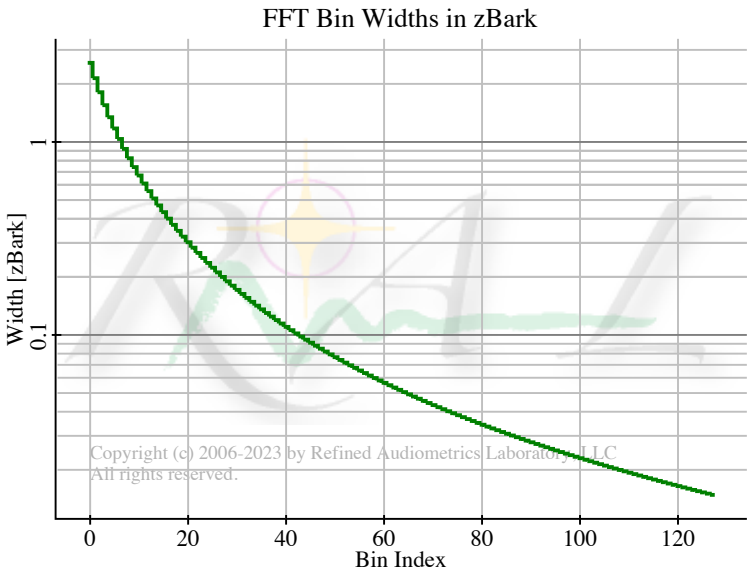

Figure 15: *FFT bins of constant absolute frequency width have varying widths when measured in units of* z_{Bark}.

the integration:

$$\Delta P_i = \int_{z_{i,<}}^{z_{i,>}} F(z)^2 P(z)\, dz = \int_{f_{i,<}}^{f_{i,>}} \hat{F}(f)^2 P(f)\, df$$

For the same signal, both sides have to be equal. But just as $z \neq f$, so too $dz \neq df$. So that means that the filter envelope in FFT space, $\hat{F}(f)$, is *not* just a frequency mapped copy of the filter, $F(z)$, from Bark space.

We estimate the power in an FFT bin as the magnitude squared complex amplitude in the bin, call it P_i. The right hand integral will be estimated as:

$$\Delta P_i = \int_{f_{i,<}}^{f_{i>}} \hat{F}(f)^2 P(f)\, df \approx \hat{F}_i^2 P_i\, \Delta f$$

The power value, P_i, is actually an average over the width of the FFT bin.

The left hand integral will use the midpoint values to produce: $\Delta P_i \approx F(z_{i,c})^2 P_i\, \Delta z_i$, where $z_{i,c} = (z_{i,<} + z_{i,>})/2$, and $\Delta z_i = z_{i,>} - z_{i,<}$.

And since the two integrals are equal, we now see that[4]:

$$\hat{F}_i = F(z_{i,c}) \sqrt{\frac{\Delta z_i}{\Delta f}}$$

When we map every cell in the FFT for Bark channel filters, we end up with the Bark channels looking like they do in Figure 14.

These Bark channel filters are really measuring power across Bark frequency. If we gang the individual filters together, then a constant power per Bark channel should show a reasonably flat spectrum in Bark space. The individual filters are not perfect, and show some variation in their ERB values. (cf., Figure 16) The variations are induced by slightly non-uniform spacing in Bark space, as dictated by the standard audiology test frequencies.

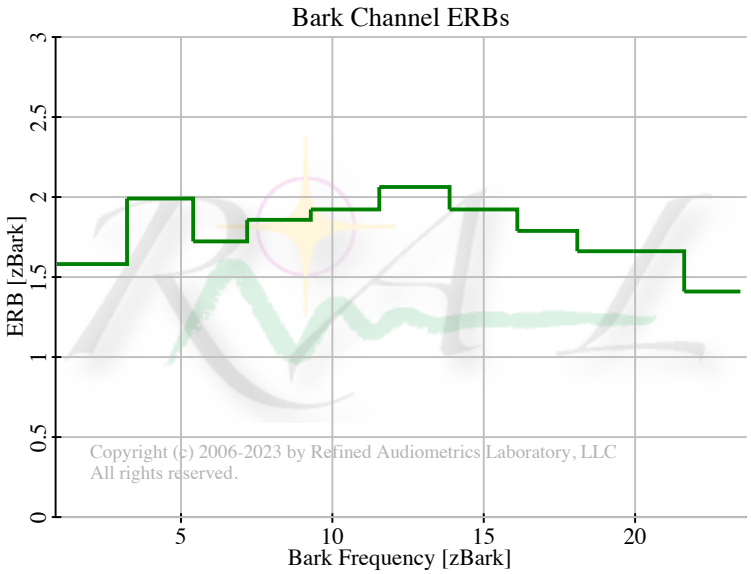

Figure 16: *Constant unit power per z_{Bark} in every Bark channel, integrated in Bark space, $\int F(z)^2\, dz$, is the same as their ERB measure. Ideally these would all have the same effective width. But the variation isn't too bad. The ratio 2 to 1.6 is less than 1 dB.*

However, if we do the same exercise using the FFT filters in absolute frequency space we

[4]We could have arrived directly at this result by using the discretized Jacobian of the transformation inside the integrals. But it was more fun to show the justification through physics.

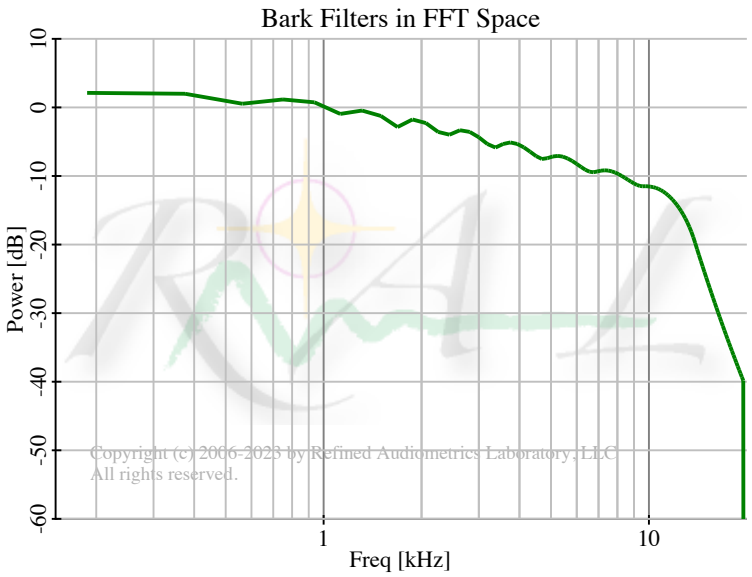

Figure 17: *Constant unit power per z_{Bark} across the Bark spectrum is pretty close to the same thing as Pink noise in absolute frequency space. Above 1 kHz we see a rolloff close to 10 dB/dec = 3 dB/oct.*

get Figure 17. But recall that we stipulated constant power per z_{Bark}. Not constant power per Hz.

Now you might argue that, look, we aren't presenting in Bark space, we are presenting sounds in real absolute frequency space. If I send a signal into the filter bank, I don't want it messing with my amplitudes like that. Let Bark space be the client, not us. That seems a reasonable argument.

So let's renormalize the FFT-space filters to unit height. (cf. Figure 18) That will just scale everything by a constant in both spaces for each filter. Now the Bark-space filters have nonuniform amplitude, in a reversal of the situation. And so when we finally form the ratio:

$$P_B = \frac{\int \hat{F}'(f)^2 P(f)\, df}{\int \hat{F}'(f)^2\, df}$$

nothing changes, despite our renormalized filters, $\hat{F}'(f)$. It seems to violate common sense, but the math doesn't lie. Constant scaling doesn't matter because it affects both numerator and denominator equally. Filter height does not matter for this kind of measurement, since we are always renormalizing by the ERB.

The real answer is just like Einstein showed everybody. You can use any ruler you want,

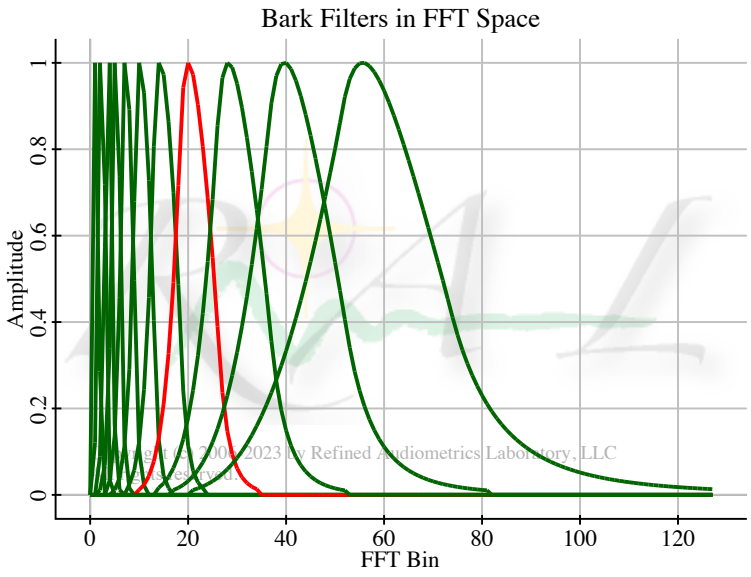

Figure 18: *FFT-space filters renormalized to unit amplitude. That seems better!*

but the physics doesn't change because of your labeling.

What *does* change through renormalization is total measured power, not average power. With unit amplitude filters in absolute frequency space, a white spectrum of noise with the same amplitude will indeed show increasing amounts of power in higher Bark bands. Lifting the filters in amplitude gives them greater effective bandwidth, and the ability to take in more power. But the average power doesn't change. The only thing that can change the average power is an increased signal amplitude.

In a way, that's comforting because we can't control spectral slopes. They will affect our filters in different ways for every different slope. But as long as the amplitude at some frequency remains constant, the average power at that frequency will remain basically unchanged. And that is the important thing for our hearing.

In practice, and to economize on CPU cycles, each of the normalized Bark channel filters is truncated where they fall below -40 dBc. The FFT bin indices at these locations are remembered and stored alongside the filter coefficients, so that filtering and power accumulation only needs to occur over a limited range of FFT bins.

With the renormalized filters in FFT space:

$$P_B = \frac{\sum_i F_i^2 P_i \Delta z_i}{\sum_i F_i^2 \Delta z_i} = \frac{\sum_i \hat{F'}_i^2 P_i}{\sum_i \hat{F'}_i^2}$$

where index, i, ranges over the bins of the FFT spectrum.

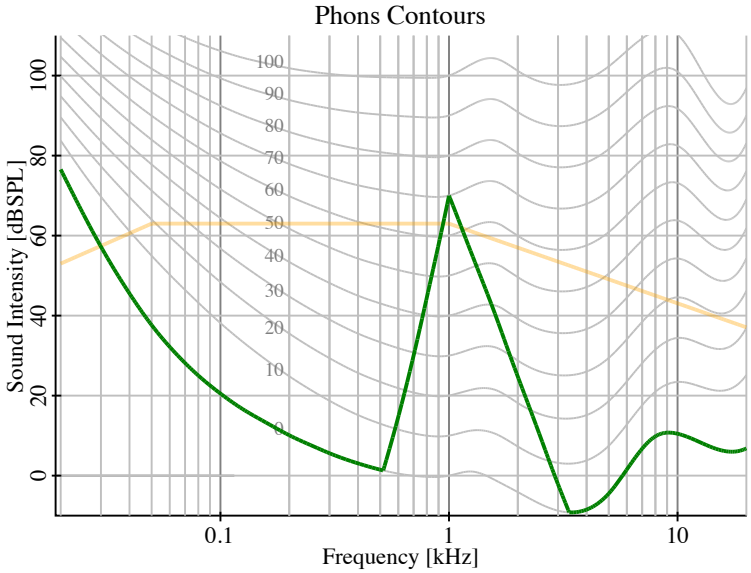

Figure 19: *Upward Loudness Masking. A strong carrier raises the ATH for everyone, and makes it more difficult to hear higher frequency components in the sound. Approximately -20 dB/zBark to the downside, and -10 dB/zBark to the upside. Faint orange trace represents a music spectrum at loud but comfortable levels.*

8.2 Loudness Masking

But that shallower rolloff toward lower frequencies might be trouble for us because most music has a spectrum that grows strongly toward lower frequencies. There will likely be lots of bleed from lower channels despite the attenuation of our filters. The good news is that the gain ratchet will be gone.

In effect we are describing the nature of upward loudness masking which already occurs in our ears. (cf., Figure 19) Since we are doing it here, and the ears will redouble, would this affect the sound too much? It turns out that nothing seems to change in listening tests. So, no compensation for power bleed seems to be needed.

Note that loud noise affects everyone - it's in the physics. But perceptually, even people with normal hearing will experience the elevated threshold effects we have been describing in this paper. The only difference is that people with impairment have thresholds that are stuck at higher levels, while people with normal hearing will gradually recover theirs.

Several US Patents were awarded for a device that listens to ambient noise and provides a Crescendo-like experience for everyone. We showed that people could carry on a normal cell-phone conversation in the middle of Times Square at rush hour, without having to cover their other ear. So too at a football game in the midst of cheers at a touchdown, when seated right behind the goal posts.

The system is adaptive and gradually transitions to normal phone use as you walk indoors to a quieter place. No manual volume adjustments necessary. This is *not noise-cancellation*, but rather, takes advantage of the physics of hearing and the perceptual equations we have shown here.

9 Crescendo Processing

In fact, the system described so far works superbly!

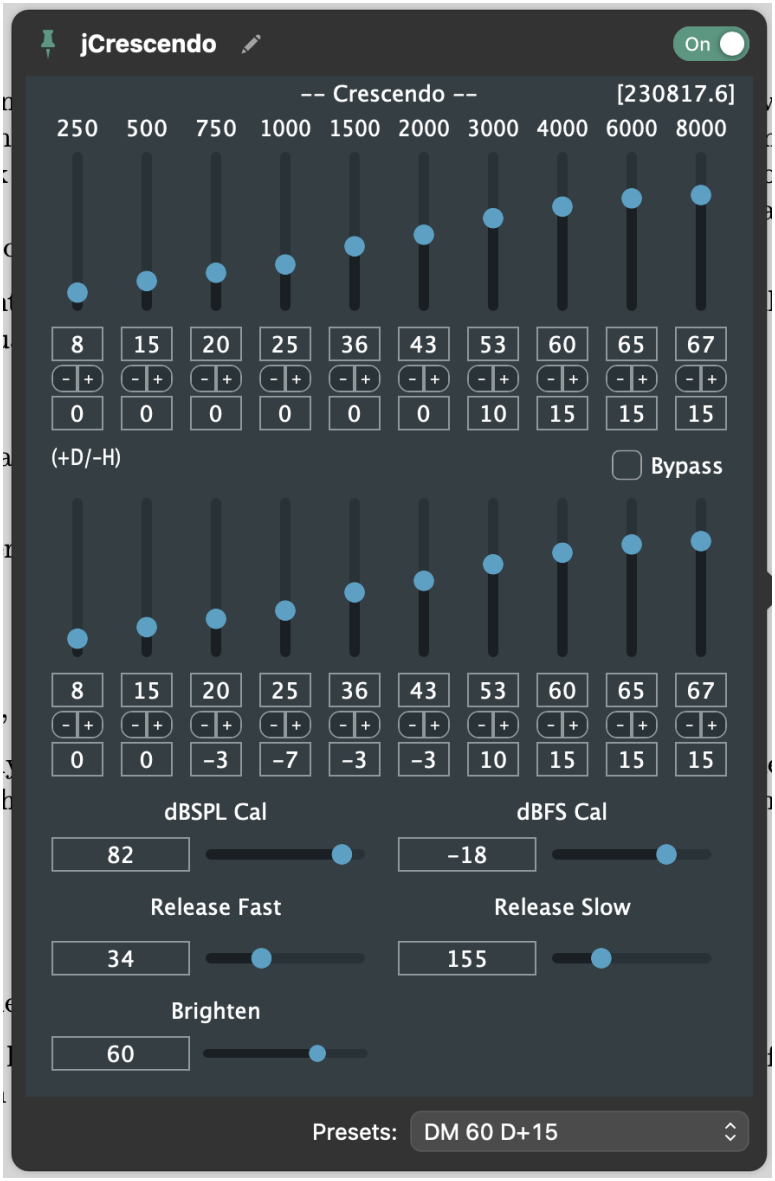

Figure 20: *A realtime DSP implementation of all of the above. It has a face that only a scientist could love.*

Crescendo (cf., Figure 20) is the name given to a DSP realization of this hearing correction system:

- It is a multi-band nonlinear compressor spanning the entire audible spectrum from 20 Hz to 20 kHz.
- The ski-slope shaped nonlinear compression in each frequency band serves to rectify the recruitment hearing curve, producing normal hearing, so long as you select the correct value for threshold elevation.
- It operates in real time, as a dynamic FIR filter, with an update rate of 375 Hz at 48 and 96 kHz sample rates, or 345 Hz at 44.1 and 88.2 kHz sample rates.
- It performs phase-linear overlap-append FFT processing with 50% overlap and a throughput latency of 192 samples (= 4 ms) for 44.1 and 48 kHz sample rates, or 384 samples for 88.2 and 96 kHz.
- FFT block size is 256 samples at 44.1 and 48 kHz, and 512 samples for 88.2 and 96.
- When sample rate changes, all the filters are either switched, or else recomputed with a Bresenham interpolation, so that we won't suffer any 10% frequency assignment change artifacts.
- It operates against stereo or monaural audio.
- It uses 11 Bark bands over the audible spectrum, each loosely $2\,z_{Bark}$ wide, mimicking the frequency behavior of human hearing in terms of bandwidth growth with frequency.
- Left and Right audio channels are processed in parallel.
- Each of the 11 Bark bands, in each stereo channel, operate in parallel.
- Users can completely customize their processing to suit their own hearing.
- If you don't know your own audiology details, you can use a single knob *Brighten* adjustment to select a setting that sounds best to you[5]. Then refine to taste.
- Compressor release dynamics are user adjustable.

[5]Over the past 20 years of trials with users, we found that most often, despite there being a scientifically rigorous method for calibrating Crescendo, most users simply went to the single-knob adjustment and moved it to a point where they most liked the sound. And in fact, many users with normal hearing, including a young mixing engineer, with well trained hearing, liked the sound of it. So perhaps we all have a bit of hearing deficit?
It actually got to the point where I began wondering if Crescendo is some kind of endorphin generator for most people? I considered doing an FMRI study on the brains of listeners while using Crescendo to see what might be happening.

- It is the *only!* system that can also correct conditions of HyperRecruitment and Decruitment hearing.
- It is ear-safe. It won't cause further hearing damage to you.
- Processing artifacts are held below -60 dBC.
- You will be thoroughly amazed by what you can still hear in the music.
- Better yet, since you normally live in a dark world, you will suddenly notice the sonic details that others cannot hear because their brains tune them out.

Crescendo was originally developed in the early 2000's for use on dedicated DSP chips (Motorola 56xxx, ADI Sharc, TI 32Cxxx) because, at that time, only a DSP chip had the horsepower to run such a system for realtime corrections.

Today, you can hardly find DSP chips, and I would argue that Arduino's don't even qualify. Of those that do still exist, you should be able to see that a Crescendo system is somewhat beyond the capabilities of an ADI SigmaDSP system. (Anyone remember the Transputers?)

I miss the modular buffer addressing and saturating arithmetic. But the processors in our laptops from Intel and ARM have since far exceeded what those early DSP chips could accomplish - despite their current lack of modular addressing modes and saturating arithmetic.

9.1 Description of Operation

Audio samples arrive in Left and Right audio channels. Both channels are processed in parallel. For each channel, arriving samples, of arbitrary buffer length, are accumulated into a circular buffer of length 3 half-blocks. After samples have filled the half-block in slot 0, slots 2 & 0 are used for power estimation and gain computation, and then the FIR filter produced by gain computation is applied to audio processing over slots 1 & 2.

This gives us a circular queue modulo 3, where the newest half-block of samples are combined with the previous half-block to obtain power estimates, and the resulting FIR filter is applied as a 2.7 ms lookahead against the two prior half-blocks of samples. The slot pointer is then incremented modulo 3 to be ready for the next batch of audio samples.

Power estimation proceeds by using a Hann window on the full block of data, then an FFT is performed. At that point we split into parallel processing for the 11 Bark bands across the spectrum. Each Bark band process is responsible for applying its Bark channel filter to the FFT data and computing power per critical bandwidth within its overall bandpass, then performing hearing correction gain computations as described above.

Once all the Bark bands have delivered their correction gains, the gains are converted into filter amplitudes for the FIR filter, using a cubic spline interpolation from the Bark channel dB space to the FFT bins. To ameliorate end effects in the time domain, due to FFT circular convolution, we perform an Inverse FFT to the FIR spectrum, window the time domain impulse response with a half-width Hann window, then go back to the frequency domain with a forward FFT. We now have a convolutional FIR filter to apply to the next half-block of audio samples,

At that point the audio data is convolved (without windowing) with the convolutional FIR filter, one full block at a time, and the central half-block of filtered data is sent out as a half-block of processed audio. Total throughput latency is 3 quarter-blocks of samples, or 4 ms, which makes Crescendo adequate for live performances.

However, depending on the host platform, the overall throughput latency for audio may be longer. We have no control over the host O/S. This is another reason to lament the demise of DSP chips. *Real men eat bare metal!*

9.2 FFT Agnostic Processing

The system initializes itself with normalization constants applicable to the whichever FFT algorithm being used. Could be from Apple Accelerate, Intel Signal Processing Libs, or even FFTW. Each of these libs have their own peculiarities of FFT-bin addressing and overall FFT normalization.

So we self-calibrate against any of them, using Hann windowed known signals of 0 dBFS

amplitude sine and cosine at the half-Nyquist frequency. The sum of total power from these two signals, estimated in the same manner that we estimate audio signal power, serves as the normalization constant for the process. This removes any oddities that might occur in FFT scaling per vendor, and also normalizes against the Hann windowing for power estimation.

Calibration waveforms are 0.0 in every even bin, +1.0 in every fourth bin 1 modulo 4, and -1.0 in every fourth bin 3 modulo 4. To get the cosine, make every odd bin 0.0, every 4th bin 0 modulo 4 is +1.0, and every 4th bin 2 modulo 4 is -1.0.

Power is accumulated over only the positive frequency bins. Bins 0 and Nyquist are taken at face power value, other bins use doubled result values for their power to account for the negative frequency components.

These calibration signals are exactly centered in an FFT frequency bin and represent the greatest power magnitude that we should encounter. The resulting calibration constant is converted to dB measure and simply subtracted from computed Bark channel dB power values prior to conversion from dBFS to dBHL.

Our audio signals are all real-valued, so their spectra have even symmetric real parts, and odd antisymmetric imaginary parts. But this symmetry is known in advance and allows us to process only the positive frequency bins of an FFT, saving a lot of needless computing cycles.

9.3 Preconditioning for dBSPL to dBHL Conversion

When Bark band processing begins, it applies its Bark channel selection filter to the incoming Hann windowed signal FFT, and sums the power across all the pertinent FFT bins, using a Bark width correction for every bin.

But that power computation also applies an inverted absolute threshold of audibility ($InvATH$) filter, which has unity gain at 1 kHz. Doing this here makes our eventual dBFS-to-dBSPL conversions into dBFS-to-dBHL. Everything is still in digital FS-space (full scale, not yet dBFS) at this point.

9.4 Compression Dynamics

At least 50% of the sound restoration quality depends heavily on how compensation gains are applied over time. We have two user adjustable release time constants: $fast$ and $slow$. One of our musician users arrived at 34 ms for $fast$, and 155 ms for $slow$. No discernible pumping occurs with these time constants.

We keep a *previous* power level for the Bark band, referring to the previous block processing, and a roving *mean* power level. Every new power level is folded into the *mean*

level with the *slow* time constant. Envelopes are applied to power levels, not their dB equivalents. Hence, in dB space, the envelopes become linear slopes.

We use a dual-release envelope on compensation gains:

- If the incoming power is higher than $previous + 6\,dB$, then we set *previous* to the current power for an immediate attack, set release to be $fast$, and set a *hold* count for 10 ms.
- Else, if the incoming power is higher than *previous*, then *previous* gets the new power level folded into itself with a $fast$ time constant. If *hold* was nonzero, it is restored to 10 ms.
- Else, if *hold* is nonzero, then decrement *hold* and we are finished with no change to *previous*
- Else, new power is less than *previous*, and we fold the new power level into *previous* with the release time constant currently in effect. Next, we look to see if *previous* has now fallen below $mean + 3\,dB$. If so, then we switch to a *slow* release.

Now we have a new value for *previous* that will be used for the next block of processing in the Bark band. We also use that value for computation of hearing correction gain.

9.5 Folding Values with Time Constants

Here is what we mean when we say that we fold a value into another with some time constant: we use an exponential average, a simple α-filter with one pole along the real-axis of the z-Plane.

$$Y_i = Y_{i-1} + \alpha \cdot (X_i - Y_{i-1})$$

This describes the algorithm corresponding to the discrete transfer function:

$$H(z) = \frac{\alpha}{1 - (1 - \alpha)\, z^{-1}}$$

The filter has a unit step response of:

$$Y_n = 1 - (1 - \alpha)^n$$

So if we want the step response to reach $67\% = (1 - 1/e)$ after one e-folding time constant period, then for sample rate, f_s and time constant, t_c, we must have $n = t_c \cdot f_s$, and so

$$\alpha = \left(1 - e^{-\frac{1}{t_c f_s}}\right)$$

When computing the α constants, just be sure to use the correct sample rate, f_s.

Do you mean audio sample rate, F_{samp}? Or do you mean half-block processing rate = $2\,F_{samp}/N_{blk}$, for block size, N_{blk}?

For our processing, we mean the half-block processing rate.

9.6 Changing Spaces

Before embarking on gain calculations, we need to transform the incoming power level to the Phon-space of those equations. First convert from FS to dBFS, then from dBFS to dBHL, and then from dBHL to Phon-space.

$$dBFS = 10\,log_{10}(FS)$$

In the conversion from dBFS to dBHL we have the same scaling, but different zero points. The initial calibration of Crescendo establishes the conversion from dBFS to dBSPL. The user sends an unprocessed 1 kHz sine wave, with some known[6] dBFS peak amplitude, through the system, and records the dBSPL at the speaker using a sound level meter.

Knowing what dBFS we used, and what dBSPL emanated, we record those two numbers and use them to convert all future dBFS power levels to known dBSPL measure.

$$Signal_{dBSPL} = Signal_{dBFS} - (Cal_{dBFS} - 3.01) + Cal_{dBSPL}$$

Once calibrated, leave the amplifier volume control alone. All future volume adjustments should be made ahead of Crescendo, so that Crescendo can know what you will be hearing.

Conversion of dBSPL to Phon takes place using MiniMax Rational Approximations to the ISO-226 model for Normal Equal Loudness Contours. This is a 2-dimensional interpolation across both frequency and intensity. But the band channel frequency is a known fixed frequency. So each band caries its own approximation over sound intensity.

The same ISO-226 model is used for conversions from Phon back to dBSPL, but uses different (inverse) MiniMax Rational approximations. This model was also used to obtain the $InvATH$ filter.

[6] There will be some possible confusion here, since some meters show Peak values, others show RMS values, and still others show LUFS, which is k-Filtered RMS+3 dB. The whole audio field is confused about these things. So just be aware that you may need a $+3\,dB$ trim on the dBFS Cal setting to compensate for the confusion.
What we actually want to know is the RMS level which, for a sine wave signal, is 3 dB less than its full-scale amplitude. But since LUFS meters are becoming more common we want the dBFS Cal setting in the GUI to reflect the Peak or LUFS value, and we perform our own subtraction of 3 dB. Have I just added to the confusion?

The ISO-226 model relates Phon to dBSPL, but we compute approximations that model Phon to dBHL and vice versa. We treat the difference between HL and SPL elsewhere. Why? Because the ATH varies significantly across the higher frequency Bank channels and we want to take that into account within the Bark channel power estimators. The approximations use the ISO-226 model in the form (dBHL + ATH) vs Phon, where dBHL is the argument of the approximation. ATH is the 0 isophon contour of the model.

9.7 Channel Gain Computations

From power estimation and envelope control, we are handed a critical band power level measured in dBHL, P_{dBHL}. This is converted to Phons measure, P, using a MiniMax interpolation of the ISO-226 model for this band frequency, F_B:

$$P = \text{dBHL\_To\_Phons}(P_{dBHL}, F_B)$$

Before using the gain equations from above, we check to see if the power level, P, in phons is dropping down below 20 phon. If so, we zero out the gain to prevent amplifying the noise floor. On the way back up, we must first cross 30 phon. This hysteresis prevents us from toggling rapidly on/off on noisy signals.

Now the fundamental hearing correction equation, including the fraction of live hair cells, is applied to the incoming phons level, P, to compute the needed compensation level in Phon-space:

$$f_{live} = \frac{1}{1 + S(P_{thr})/S(120)}$$

$$P + dP = \text{Phons}\left(\frac{\text{Sones}\,(P) + (f_{live}\,\text{Sones}\,(P_{thr}) - \text{Sones}\,(0))}{f_{live}}\right)$$

We then convert back to dBHL-space, using an inverse MiniMax interpolation for ISO-226, and subtract the incoming level to obtain a gain adjustment value in lab-frame, SPL, units of dB:

$$G_{dB}(dP) = \text{Phons\_To\_dBHL}(P + dP, F_B) - P_{dBHL}$$

We use the high fidelity solution, not its approximation. It might be argued that the approximations would sound just as good. As we have seen, restricting dBHL Threshold adjustments to 1 dB increments more than outweighs the difference between the exact solution and the approximations. But we know exactly what to do, and it isn't much more trouble to use the correct equations.

After obtaining the gain, G_{dB}, we smooth out any upward changes in gain using the previous block gain with a time constant of 20 ms. Downward changes are immediately accepted in whole, in the interest of ear safety. Then the smoothed gain is limited to

a maximum permissible amount before reporting its linear multiplicative value for the pending FIR filter spectrum.

The gain, G_{dB}, in dB is converted to a linear filter amplitude (*not!* power magnitude), using:

$$Ampl(dP) = 10^{G_{dB}/20}$$

9.8 Putting the Channels Back Together

Once all the Bark channels have their filter amplitudes computed, we spread the gains across an FIR spectrum using cubic spline interpolation from Bark space to the absolute frequencies of the FFT bins.

The FIR filter is then computed by inverse FFT to the time domain, where a half-width Hann window is applied to prevent edge wrap effects during convolution with the next half block of audio samples. A frequency domain filter is then obtained by transforming back to frequency space with a forward FFT.

FIR convolution is performed in the frequency domain since that offers the highest efficiency. A full block of audio containing the prior half-block plus a new half-block is transformed, without windowing, to the frequency domain, using an FFT. That audio spectrum is multiplied by the frequency domain FIR filter, and the result is inverse transformed back to time domain.

Since FFT spectral multiplication is performing circular convolution in the time domain, we pick out the middle half-block of the convolved audio buffer, which is free of circular convolution wraparound effects, and send that along as the next half-block of processed audio.

Repeat this whole process for every incoming half-block of audio, every 2.7 ms. An FIR filter this short would not be much use until it spans about 3 periods of the lowest frequency of interest. That's just above 1 kHz for us. And there, and higher, is where most hearing damage occurs in typical sensioneural hearing impairment. Accuracy grows with increasing frequency, and that's exactly what we need to happen. We can take your 250 Hz and 500 Hz hearing as your version of great hearing, and go from there.

Convolution of audio with this dynamically updated FIR filter, whose gain profile is adjusted in response to incoming audio, is a multi-band nonlinear audio compressor. That's a *Crescendo Engine!*

9.9 What the Brighten Knob Does

Over the past 20 years, we found that most users don't know their own audiology. And even if they do, they like to skip the calibration of Crescendo and just dive into using it.

They often don't even need a Crescendo, but they like the sound of it.

So we ask them to set up their amplifier to a loud, but comfortable, level. And we give them some typical starting values, like 77 dBSPL for -23 dBFS (or -15 dBFS if you follow Apple's standards). Most people gravitate to 77 dBSPL in a small room.

Then we have them vary the Brighten knob until they like the sound best. They can turn down the volume if they like, but do it ahead of Crescendo, not at the amplifier. Afterward, they might trim up a few threshold elevations in individual bands to suit their taste.

We find that for most typical cases of sensioneural hearing loss, they tend to exhibit a fixed slope in their threshold elevations, when viewed in Phons-Bark space. The slope is about $3.28\,dB/z_{Bark}$, declining toward lower frequencies. The straight line decline isn't so apparent when you look at an audiogram in dBHL-Hz space.

The slope is fixed, but the zero point is not. What sets people apart is how much damage has happened, but not how that damage is apportioned to each Bark frequency band. That seems to be dictated by physics.

So, choose an afflicted frequency band, like 4 kHz, and find the amount of damage to that channel. Then all the other channels can be estimated from using the common slope. That's exactly what the Brighten slider does - it varies your estimate of the 4 kHz threshold elevation, and makes all other band thresholds follow along that slope in Phons-Bark space.

Now why would this happen?

9.9.1 The Storm Model for Hearing Damage

In the cochlea, the highest frequencies are detected by the sensors closest to the oval window where sound enters the cochlea. Lower frequencies are sensed further along the basilar membrane. Bass is sensed at the far end, or apex, of the cochlea.

We know this, thanks to Dr. Bekesey, who won the 1961 Nobel Prize in Medicine for his *Position Theory of Pitch Sensation*. In fact, Bark frequencies are laid out linearly in position along the basilar membrane, with Bark 0 at the apex, and Bark 24 at the oval window.

Consider a tropical storm approaching landfall. That is an analog of loud noise impinging on the oval window of the cochlea. As the storm lands, it does its greatest damage to the shoreline (the highest frequencies in our hearing). And as the storm continues inland it continues damaging, but with gradually depleting energy and consequently less damage (the lower frequencies of our hearing).

One can model the storm damage as an exponential decline - the depletion of energy from causing damage is in direct proportion to what energy remains. This is a classical

situation, first described by Isaac Newton. And it is probably the first differential equation encountered by most people.

But an exponential decline in power space looks exactly like a linear slope in dB power space. Hence the pretty good approximation of $3.28\,dB/z_{Bark}$, declining toward lower frequencies.

That particular slope must be dictated by viscosity of the cochlear fluid, and the coupling efficiency along the basilar membrane. But our value was empirically determined from dozens of audiograms from other people.

People don't have audiology that precisely follows the straight line slope. But departures are often less than about ±5 dBHL. So after you have an initial estimate for your hearing thresholds, found by moving the Brighten slider to the level that sounds best to you, you can then trim up the individual channels to suit your taste.

That is how 90% of our users set up their Crescendo systems. It turns out that, despite physics, and just like our sense of visual color, our sound preferences rely heavily on our individual perceptions, and our personal tastes. Physics be damned!

10 Audio Monitoring Chain

Since we use FIR filtering against the untreated audio, our gain corrections are linear phase corrections. The whole process can be considered a time varying FIR filter with an update rate of 375 Hz. Nothing much can happen in just 2.7 ms, and that keeps the updates performed on the FIR filter to relatively small changes.

Sidebands generated by the update modulation are kept small, and the inevitable aliasing which occurs is smaller still - well below the level of the carrier. And thanks to loudness masking in our ears, we can't hear what little sideband power develops. Sideband artifacts have been measured as $< -60\,dBC$.

Crescendo can significantly increase the level of the highest frequencies, depending on how severe the hearing impairment. So to prevent clipping distortion on the way out of the DSP, we pre-scale the output from Crescendo when necessary, and then make up for lost volume levels at the amplifier.

In my own system, I use -15 dB pre-scaling, set up an unprocessed -23 dBFS sine wave and subsequent pre-scaling to produce 77 dBSPL at the speaker using the volume control of the amplifier. This is plenty loud in a small room or in headphones. You only need dBx levels of 83 dBSPL for large theaters. Most of my listening occurs at a nominal 70 dBSPL level in headphones.

All future volume control adjustments must occur ahead of Crescendo so that it knows what SPL level you will be hearing. By not putting the pre-scaling into the Crescendo plugin, and using pre-scaling in a separate plugin after Crescendo, means that you won't be blown out of your seat if you accidentally disable Crescendo processing. From this perspective, you can (almost) view Crescendo as a unity gain processing stage.

Crescendo assumes it is feeding a spectrally flat transducer. You should introduce all coloration EQ ahead of Crescendo. People with impaired hearing do not hear EQ in the same way as people with normal hearing. EQ becomes exaggerated in a loudness dependent manner. A little peaking or dipping at loud signal levels becomes a lot of peaking or dipping at soft levels. So we avoid that by flattening the speaker and headphones after Crescendo, then reintroduce headphone or speaker coloration, as desired, in front of Crescendo. That way we can all hear the same thing.

The final processing chain, on the way out to the speakers, is:

$$Audio \rightarrow Crescendo \rightarrow PreScale \rightarrow FlattenEQ \rightarrow Amplifier \rightarrow Speakers$$

All other audio processing should be placed ahead of Crescendo, along with a volume control.

Since I spend so much time listening through headphones, and since recordings are not specifically mixed for headphone listening, the stereo separation is too extreme when the left channel feeds only the left ear, and right channel feeds only the right ear. Loudspeakers allow the left and right channels to mix in the room air before reaching each ear.

So I simulate that with high-pass filtering on the Side channel, and use partial Left/Right cross-feeds at higher frequencies. One could also invoke a small delay in the cross-feeds to induce the Haas Effect.

Secondly, since I listen at diminished volume levels, I use a calibrated Loudness Compensation, which drops the monitoring level and lifts mostly bass and a small amount of treble. So I get to hear the spectral details that normally only show up at louder listening levels.

Next, I do use some headphone re-colorizing EQ to try to get some semblance of the Harmon profile for my monitoring chain.

Finally, when in my lab, the central air conditioner is nearby and running a lot of the time - living in the hot desert does that. So I lift the bottom levels up with a parallel compressor running at a high compression ratio, some makeup gain, and a low threshold. This has the effect of boosting the weakest sounds down around 40 dBSPL to get them above the air conditioner bleed in my headphones, while leaving the louder portions of music untouched.

Headphone cups on closed-back headphones can only attenuate outside high frequencies, but freely pass deep bass, like 60 Hz. Recordings of my room ambient noise made through an artificial head, with and without headphones, look almost identical in the bass region below 1 kHz. Above 1 kHz the headphone covered ambient noise rolls off nicely. So I have to live with an ambient noise level of around 47 dBSPL, and need to boost the musical floor above that.

All of these processors - the Calibrated Loudness Compensation, headphone cross feeds, headphone re-colorizing, and parallel floor-lift compressor, are planted in front of my Crescendo in my monitoring chain.

My own hearing impairment is described as moderately severe. I have woofers for ears, with -24 dB/octave rolloff above 1 kHz. Yet, through Crescendo I can hear the exquisite sibilance of my wife's speech at 10 kHz. Most people have their sibilance down around 6-7 kHz. Hearing aids typically process from 500 Hz to 5 kHz. I use Crescendo all day long, every day, for the past 20 years.

Just as an aside, nearly all of the development of Crescendo took place with filter design, system analysis, math solving, etc., using Lisp, the computer language. I continue that tradition with maximum enthusiasm.

All of the graphs presented in this paper were generated by my Lisp system. I am able to explore what-if situations right from the keyboard in an ad-hoc immersive, interactive, incremental, environment unlike any others. Long live Lisp!