BIELEFELD UNIVERSITY

FACULTY OF TECHNOLOGY

BACHELOR THESIS

Development of an adaptive tuning system for music which co-adapts to environmental noise

Author:
Arne Kramer-Sunderbrink

1. Supervisor:
Dr. Jiajun Yang

2. Supervisor:
Dr. Thomas Hermann

October 4, 2019

Contents

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Literatur und Hilfsmittel angefertigt habe. Wörtlich übernommene Sätze oder Satzteile sind als Zitat belegt, andere Anlehnungen hinsichtlich Aussage und Umfang unter Quellenangabe kenntlich gemacht. Weiterhin sichere ich zu, dass die Arbeit nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde. Dieser Datenträger dient der Vorlage bei den Prüfern und dem Prüfungsamt. Der Inhalt der Arbeit darf Dritten ohne ausdrückliche Genehmigung des Verfassers nicht zugänglich gemacht werden. Dies gilt insbesondere für die kommerzielle Nutzung, Server von Dritten oder die Überprüfung mit Hilfe von Plagiatssoftware.

Bielefeld, October 4, 2019

_____

Arne Kramer-Sunderbrink

1 Introduction

Most music can be viewed as a composition of single tones, but not all combinations of tones were created equal. Humans distinguish between combinations that sound pleasing and those that are unpleasant: Consonances and dissonances. Composers of film scores know of the effects dissonance and consonance have on us and use it to manipulate our feelings and attention. They use dissonance to put us in a state of tension, consonance to evoke feelings of rest, safety or even confidence (Pavlović and Marković, 2011). Think of the confident melody of the star wars main theme by John Williams that targets only the most unambiguously consonant notes of the scale: the tonic and the perfect fifth. Or think of the jarring high pitched chords of the famous shower scene in Bernard Herrmann's score for Hitchcock's Psycho, that are so effectively shocking that they have become a cliche in horror movies.

If those composers would be challenged to compose the soundtrack to a workspace or a study room, they would certainly make use of consonance to enhance its inhabitants well being and productivity. After all: Multiple empirical findings suggest that music in general can not only promote the well being of listeners, i.e. reduce stress (Labbé et al., 2008) and anxiety (Lee et al., 2005), generally rise the mood (Ferguson and Sheldon, 2013) and improve the listeners health even on a biochemical level (Chanda and Levitin, 2013), but also enhance performance on different kinds of tasks, in the workplace (Fox and Embrey, 1972; Lesiuk, 2005; Allen and Blascovich, 1994) or in academic and educational contexts (Hall, 1952; Črnčec et al., 2006). On the other hand, Dolegui (2013) for example has shown that music can have the opposite effect also. Whether music is beneficial in some specific context needs to be decided on a case-by-case basis. Ravaja and Kallinen (2004) suggest that the actual effect music has on performance does not only depend on the kind of music and the kind of task but can also depend strongly on personal dispositions of the listener. See Dalton and Behm (2007) for a systematic review of positive and negative effects of music on different kinds of performance.

Lets say we decided that it is a good idea to play music in some environment, what kind of music should we play? There are a number of musical parameters that have been shown to shape the effect of music: While 'beaty' music is good for repetitive tasks (Fox and Embrey, 1972), the opposite is true for tasks that require a higher amount of concentration: Legato music is better for memory related tasks than staccato (Schlittmeier et al., 2008), more generally, the fluctuation strength of the music should be low (Ellermeier and Zimmer, 2014), and especially music with lyrics tends to disturb the listeners concentration (Shih et al., 2009).

The role consonance has in producing the desirable effects mentioned above has not been studied until very recently. For example Masataka and Perlovsky (2013) found that, while dissonant music facilitates cognitive interference (e.g., when reporting the color of the ink of a word that designates a different color), consonant music mitigates it, and Komeilipoor et al. (2015) found that it is easier to synchronize and coordinate movement to a metronome if its pulses consists of a consonant chord. In general, there are good reasons to believe that there is a correlation between the positive effects of

music described above and consonance (Bonin and Smilek, 2015).

But the soundscape of the spaces we inhabit in our everyday life can never be controlled to the same degree as that of the movie theater: A humming fridge or ventilation system could ruin the consonance of our composition if it would be humming at just the wrong frequency. We cannot always control what frequencies we are subjected to in our everyday life – but what if we could control the music we play to adapt to the environmental noise we hear? What if we could adapt the tuning of some precomposed piece of music or even a live performance dynamically to pronounced frequencies in the environmental noise?

In fact, the ATMEN system (Adaptive Tuning of Music to Environmental Noise) described in the following text is able to do just that – it takes a precomposed piece of music in form of a midi file or a live performance in form of a midi stream on the one hand, and some recording of environmental noise in form of an audio file or an audio stream on the other hand, analyses the environmental noise for pronounced frequencies and plays back the midi data at frequencies that, while staying true to the original musical material, are fine tuned to minimize the inner-musical dissonance as well as the dissonance between the music and the pronounced frequencies in the environmental noise.

In particular, Section 2 will specify the concept of dissonance that will be used throughout this project. Section 3 will state the basic problem of tuning and look at previous adaptive tuning systems and how they can be utilized for tuning to environmental noise. Section 4 presents the implementation of ATMEN, especially the construction, mathematical definition and optimization of a dissonance measure tailor-made for our purposes, followed by an evaluation of the behavior of the system in Section 5 and finally a conclusion and outlook in Section 6.

The project is developed in Python and it is available as an open source library on Github[1].

2 The beating theory of dissonance

When a set of tones sound together, the result is perceived as pleasant or unpleasant i.e. consonant or dissonant. Entangled in this seemingly simple phenomenon are a number of complex concepts. Our perception of a (musical) sound depends on the mechanics of our ears, the intricacies of our neural signal processing, and our personal and cultural experience with music and other sounds. Accordingly, there are a number of different concepts related to the dissonance-consonance-dichotomy (see Parncutt and Hair, 2011), but we will focus only on one aspect here, commonly referred to as *roughness*.

Pythagoras is often credited as the first to attempt a mathematical description of musical dissonance. He discovered that the sound of two vibrating strings is perceived as pleasant when the length of the strings (which determines their frequency)

[1] `https://github.com/ArneKramerSunderbrink/adaptivetuning`

corresponds to a simple integer ratio. In this way, he discovered the intervals of the octave (2 : 1), the fifth (3 : 2), and the fourth (4 : 3) (Loy, 2011: 48). To this day, an interval is called *pure* or *just* if it corresponds to a simple integer ratio (see Table 1). 2000 years latter, Hermann von Helmholtz was able to explain this observation in terms of roughness.

Semitones	Interval	Short	Ratio
0	unison	u	1 : 1
1	minor second	m2	16 : 15
2	major second	M2	9 : 8
3	minor third	m3	6 : 5
4	major third	M3	5 : 4
5	fourth	4	4 : 3
6	tritone	t	45 : 32
7	fifth	5	3 : 2
8	minor sixth	m6	8 : 5
9	major sixth	M6	5 : 3
10	minor seventh	m7	9 : 5
11	major seventh	M7	15 : 8
12	octave	o	2 : 1

Table 1: The most common just intervals and their names used throughout this text.

When two close simple tones (pure sine waves) with frequencies f_1 and f_2 sound together, their sum can be described as a single simple tone of frequency $\bar{f} = \frac{f_1+f_2}{2}$ whose amplitude is modulated with the frequency $\Delta f = |f_1 - f_2|$ (see Figure 1). Roughly, these *beatings* are perceived as smooth and pleasant when they are small ($\Delta f < 10$ Hz, think of the characteristic vibrato of a vibraphone) and as rough and unpleasant at $\Delta f \approx 30$ Hz (see von Helmholtz 1968: 286, 317-318 and Roederer 1975: 27-29). For bigger intervals, the perception of beating fades and we perceive the resulting sound as two different beatless tones.

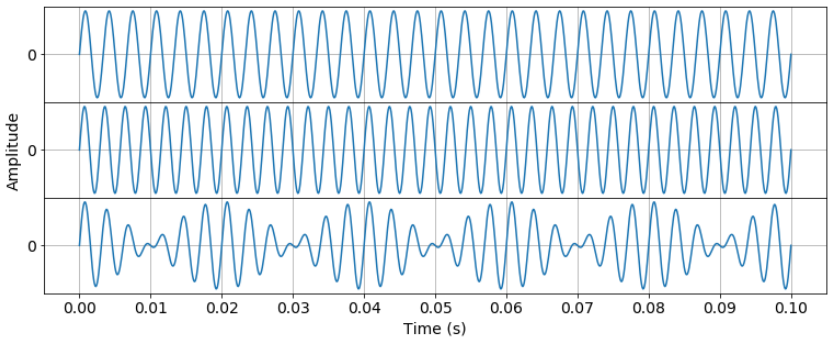

Figure 1: Adding two sine waves with frequencies 600 Hz and 700 Hz gives a sine wave of frequency $\bar{f} = 650$ Hz whose amplitude is modulated with $\Delta f = 100$ Hz.

More accurately, as Plomp and Levelt (1965) have shown, the shape of the roughness curve is not independent of the absolute frequencies of f_1 and f_2, but rather depends on the critical bandwidth (CBW) at $\bar{f}$. The concept of CBW, originally conceived by Fletcher (1940), is related to a great number of psychoacoustic phenomena. For example, the perceived loudness of filtered noise with a constant sound pressure stays constant until the bandwidth of the filter surpasses the CBW, only then the perceived loudness begins to increase. Similar effects can be observed in many different kinds of experiments, and in all those experiments similar bandwidths can be observed (Scharf, 1970: 159). CBW varies as a function of the center frequency $\bar{f}$ of the band: it stays approximately constant at 100 Hz for center frequencies below 500 Hz, from there on it starts to rise with a slope of approximately 0.2, which corresponds to the interval of a minor third around $\bar{f}$ (Fastl and Zwicker, 2007: 159) (see Figure 2). Plomp and Levelt found that the beating sensation occurs if $\Delta f < 1.2\ cbw(\bar{f})$ and is maximally unpleasant at $\Delta f \approx 0.25\ cbw(\bar{f})$ (see Figure 3).

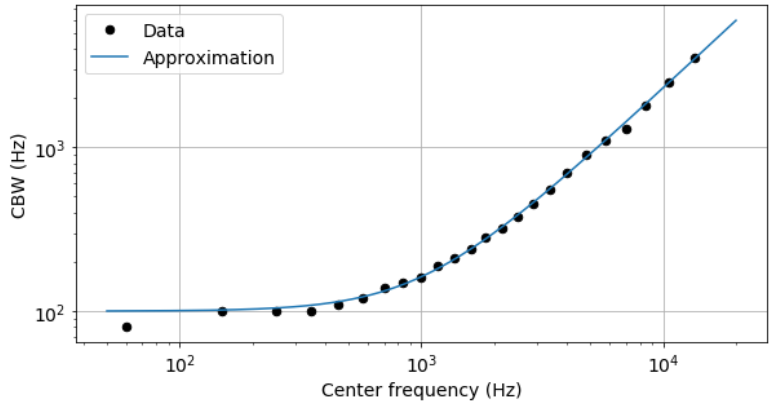

Figure 2: Critical bandwidth as a function of the center frequency $\bar{f}$ of the band: Empirical data by Zwicker (1961) and approximation from Zwicker and Terhardt (1980).

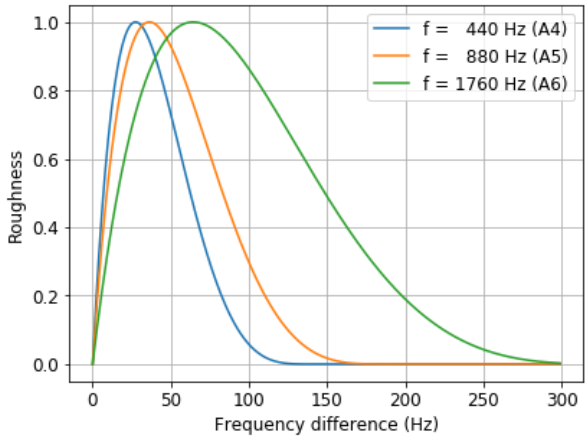

Figure 3: Roughness of two simple tones with frequencies f and $f + \Delta f$ as a function of Δf for $f = 440$ Hz, $f = 880$ Hz, and $f = 1760$ Hz.

In his seminal work "Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik" ("On the Sensations of tone as a physiological basis for the theory of music"), von Helmholtz (1968) formulated the following theory of dissonance based on the phenomenon described above: Most tones we hear are not pure tones but *complex tones* composed of multiple pure tones (*partials*). Often one of the partials (usually the loudest and lowest) is perceived as dominant. We call this tone the *fundamental* and the partials above it its *overtones*. The sound, or *timbre*, of a complex tone is determined by the relative position and loudness of its overtones – its *spectrum*. Now beating can occur between any pair of partials of two complex tones.

This way, Helmholtz is able to explain the consonance of intervals that correspond to simple integer ratios: Most musical tones have harmonic spectra, i.e. the frequencies of their partials are multiples of the fundamental frequency. While the consonance of a bigger interval like a perfect octave ($f_2 = 2f_1$) in comparison to a slightly detuned one cannot be explained in terms of beating between the fundamental frequencies f_1 and f_2 because the distance between them is to big to produce beating, it can be explained in terms of beating between the partials $2f_1$ and f_2, $4f_1$ and $2f_2$, $6f_1$ and $3f_2$, and in general between $2nf_1$ and nf_2 for $n \in \mathbb{N}$, (the relevance to the dissonance of the tone quickly decreases as the intensity of the overtones decrease). When the octave is in tune ($f_2 = 2f_1$) then $2nf_1 = nf_2$ for all n – the partials match perfectly and cannot produce beating. When the octave is slightly out of tune $(f_2 = 2f_1 + \epsilon)$ then $nf_2 - 2nf_1 = n\epsilon$ – the partials differ by a small amount and will produce beating. In general, for an interval $a : b$, the potentially matching pairs are (bnf_2, anf_1) for $n \in \mathbb{N}$. As you can see, for less simple integer ratios $a : b$, potentially matching partials will be less frequent and higher (and hence the corresponding intensities will be lower). This is why intervals corresponding to more complex frequency ratios sound less stable: Their partials cannot "lock into each other" as strongly as those of simpler ratios. Also, the partials that does not match tend to get more close to each other and will produce beating even without the beating of slightly mistuned matching partials – this is why intervals corresponding to more complex frequency ratios sound more dissonant, even when tuned to a rational interval (see Figure 4).

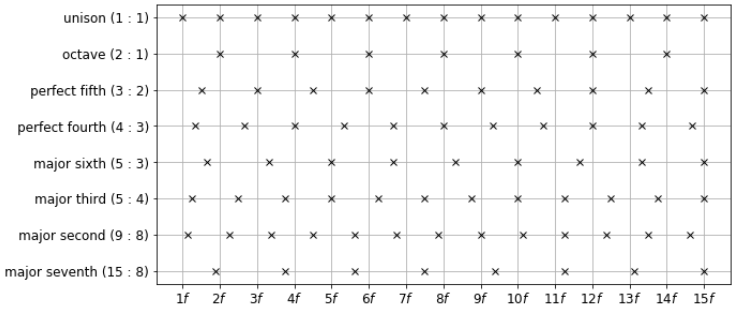

Figure 4: Locations of harmonic partials of the tones of a just diatonic major scale with fundamental frequency f.

This explanation can be visualized by plotting the dissonance of a pair of harmonic complex tones as a function of the frequency of one of the tones, keeping the other

tone fixed: If we compute the total dissonance of the complex tones by summing the roughness of every pair of partials as defined by some formula that yields the shape described above (Figure 3), we get a characteristic dissonance curve with local minima at the just intervals. If we draw a separate line for two complex tones with 1 to 11 partials, we can see clearly how every new overtone adds new local minima to our dissonance curve and how the familiar set of consonant just intervals (unison, octave, fifth, fourth, major third, major sixth, minor third) arises naturally for complex tones with six partials. Note also, that sharper and deeper minima appear at more simple frequency ratios (unison, octave, fifths), corresponding to their higher sensitivity to small detuning (see Figure 5). We also see other local dissonance minima emerge, for example between the major sixth and minor seventh, that sound strange and unfamiliar to us, because they are nor utilized in our western music, but nevertheless have a quality of stability that is similar to that of the more familiar intervals. When we change the timbre of the tones to be non-harmonic, we find even more unfamiliar local minima. This great flexibility of Helmholtz's theory, allowing us to not only confirm our familiar scales but to predict the stable intervals of exotic timbres and find alternative scales that share the quality of stability of our familiar scale, was explored by Sethares both in theory (Sethares, 1993) and praxis (Sethares and Hobby, 2016; Sethares et al., 2017), and it will be utilized later to tune musical tones to environmental noise.

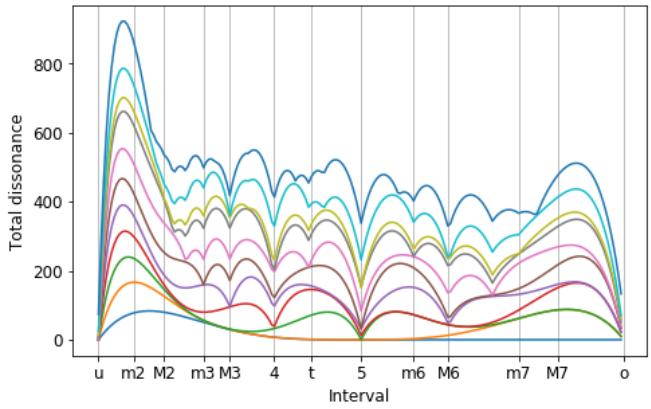

Figure 5: Dissonance curve of two complex tones with up to 11 partials whose spectrum is approximately that of a piano, i.e. the relative amplitudes of the partials are: 1, 1.46, 0.32, 0.3, 0.26, 0.16, 0.14, 0.18, 0.0002, 0.03, 0.05.

3 Adaptive tuning for musical instruments

3.1 The tuning problem

Our novel task to tune the frequencies of a set of notes to reduce the dissonance between them and some environmental noise is closely related to the old task of tuning an instrument to reduce the inner-musical dissonance of the tones is produces. In practice, instruments are usually tuned directly by applying the very same concept

of dissonance we explored in the last chapter: Two notes are played that should produce a consonant interval (usually the unison or octave) and the frequency of one of them is fine tuned until the sensation of beating disappears. However, in theory the question of how to tune a set of notes is almost always asked in terms of justness (we have seen how these concepts are related in Section 2): How do we choose the frequencies of a set of notes such that all intervals between all pairs of notes are as close as possible to some simple integer ratio. Why, in 2000 years of searching, has nobody found a "perfect" scale containing only just intervals? Somewhat surprisingly, this is mathematically impossible.

Let us specify the tuning problem more clearly: A scale assigns every pitch (e.g. the pitches of the western twelve tone scale: C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B in every octave) a frequency. These frequencies form intervals. For example, a typical just scale based on the reference pitch $\mathrm{C}4 := 260$ Hz will define $\mathrm{G}4 := 390$ Hz which corresponds to the interval of a just fifth $\mathrm{G}4 : \mathrm{C}4 = 390 : 260 = 3 : 2$. Usually we define only a finite set of pitches and repeat the same interval pattern every octave, where the octave corresponds almost universally in all cultures to the interval $2 : 1$ (remember that we have shown that the octave is the most consonant interval in Section 2). For example, we generalize our definition $\mathrm{G}4 := 390$ Hz using the rule $\mathrm{G}(4+n) := 2^n\ 390$ Hz for all octaves $4 + n$ in the pitch range we need. This not only makes the construction of a scale easier but first and foremost complies with the principle known as *octave equivalence* (Terhardt, 2015; Loy, 2011: 14): Musical notes (i.e. complex tones with harmonic timbres) that are an octave apart sound so similar to each other (see Figure 4) that they can be viewed as equivalent: they can generally be exchanged for each other without altering the harmonic character or function of the chord they appear in.

We obviously want to have more than one note per octave. Lets say we choose $\mathrm{C}4 := 260$ Hz as our first pitch. Then we want to find a second pitch $P = f$ between C4 and C5. These pitches will form the interval $i = P : \mathrm{C}4$ with $1 < i < 2$. Now we want our scale to allow for transposition. Simply put, this means we want to be able to play the same music no matter from which pitch we start. Hence we need a pitch P' that forms the interval i with P and hence the interval i^2 with C4. In general, to be able to transpose to every pitch in our scale, we need all the pitches $i^n\mathrm{C}4$ and all their octaves $i^n 2^m \mathrm{C}4$ in our scale. Since we don't want infinitely many different pitches in every octave, we want to *close the octave*, i.e. we want to to arrive at some octave of C4 by stacking the interval i, we want $m, n \in \mathbb{N}$ such that $i^n = 2^m$ to make sure that we only get a finite number of pitches in every octave, namely n. When we said earlier that we were looking for a *perfect* scale, this is what we meant: A scale that contains only rational intervals, allows for arbitrary transpositions, contains all octaves, more than one pitch, but only a finite amount of pitches per octave. Now we managed to make demands that cannot possibly be met, because $i^n = 2^m \Leftrightarrow i = \sqrt[n]{2^m}$. 2^m is an integer and the root of every integer is either an integer itself or irrational. i cannot be an integer since we required $1 < i < 2$ (we want more than one pitch per octave!), hence i must be irrational and cannot be just.

In practice, the problem is even worse because we not only want rational intervals but *simple* interval ratios, i.e. $i = a : b$ with a, b small. Therefore we run into problems

even in a single octave and without demanding arbitrary transposition: Say we want the interval of a fifth to be $3:2$, $4:3$ for a fourth, and $5:4$ for a third. Say we want to play the three major triads in the key of C: C major, F major, and G major. C4 and G4 form the interval of a fifth in C major (in root position), hence we need to define $\mathrm{G}4 := 3/2\ \mathrm{C}4$; D4 and G4 form a fourth in G major (in first inversion), which yields $\mathrm{D}4 := 3/4\ \mathrm{G}4 = 9/8\ \mathrm{C}4$; F4 and C4 form a fourth in F major (in second inversion), so $\mathrm{F}4 := 4/3\ \mathrm{C}4$; A4 and F4 form a major third in F major, so $\mathrm{A}4 := 5/4\ \mathrm{F}4 = 5/3\ \mathrm{C}4$. Now A4 and D4 form the interval $40:27 = 1.\overline{481}$, which is dangerously close to a fifth and, even though it is obviously a rational interval, sounds extremely dissonant. The relative error of this flat fifth ($81/80$) is called the syntonic comma, and until the eighteenth century many scales were proposed that tried to avoid such commata while keeping as many just intervals as possible (Bibby, 2009: 24-26). In general all this scales sounded more in tune in some transpositions than in others, and as composers and musicians demanded more and more to be able to play in arbitrary keys (think of J. S. Bach's set of preludes and fugues in all 24 major and minor keys) people abandoned the demand for rational intervals and agreed on the now ubiquitous twelve tone equal tempered scale (12TET). In 12TET, the interval of a semitone is defined to be exactly one twelfth of an octave: $2^{1/12}$. If we use the modern standard pitch $\mathrm{A}4 := 440$ Hz as a reference, we get $\mathrm{A}\sharp 4 := 2^{1/12}440$ Hz, $\mathrm{B}4 := 2^{2/12}440$ Hz, $\mathrm{C}5 := 2^{3/12}440$ Hz etc. In this scale, every two notes that are the same number of semitones apart form the same interval, all keys are equally well in tune and the irrational approximations are remarkably close to the just intervals for most intervals. For example, $2^{7/12}$ is very close to the just fifth ($3:2$) and sounds just as consonant to most ears. On the other hand, the equal tempered major third ($2^{4/12}$) sounds noticeably sharp in comparison with the just major third ($5:4$). This led many musicians and theorists to view 12TET not as the final solution but a temporary compromise.

von Helmholtz (1968: 523) for example clearly and unambiguously expressed his aversion to 12TET and praised the human voice for its ability to "follow the desires of a delicate musical ear most easily and thoroughly" (von Helmholtz, 1968: 526, translation by the author). In fact, the problem just discussed does not apply to the unaccompanied voice (and to some degree other musical instruments like fretless string instruments for example) since it does not need to obey a fixed scale. Say a choir would like to sing a progression containing the chords of our example above: F major, G major, C major, followed by D minor (containing D4 and A4). The singers could use the just intervals described above for the first three chords and when singing the D minor, sing the D4 slightly lower ($10/9$ C4 instead of $9/8$ C4) than the D4 they sang in the G major chord. Of course, this is only possible when the conflicting intervals do not sound at the same time – when a choir is asked to sing a pentatonic cluster chord containing al the notes C4, D4, F4, G4, and A4 it would have to find some tempered solution. Nevertheless, the human voice is generally capable of producing a much more satisfactory harmony than an instrument which is bound to a fixed scale by locally adapting its scale to the currently sounding notes.[2]

[2]Whether a choir really does choose just intervals or any other ideal tuning is an open question (see

This observation inspired instrument makers in the early baroque to construct keyboard instruments with more than twelve keys per octave. The player was supposed to decide which version of a pitch to play (e.g. D4 = $9/8$ C4 or D4 = $10/9$ C4) depending on the local musical context. As we have seen, we cannot close the octave with a finite amount of pitches, so these instruments were still compromises, no matter how many keys they managed to put into an octave – for example von Helmholtz (1968: 523) advertised for an organ with 24 pitches per octave that can be selected via registers. These instruments never established themselves for the obvious reason of being practically unplayable (Stange et al., 2017: 4).

3.2 Knowledge-based adaptive tuning

In the twentieth century, the invention of electronic instruments opened up new possibilities for adaptive tuning: Early designs of adaptive tuning systems worked exactly like the organ Helmholtz envisioned, but delegated the hassle of choosing the right version for each played pitch to some logic circuit or algorithm (Stange et al., 2017: 4-6). These algorithms are essentially expert systems: They work by applying musical knowledge explicitly written into their code. A recent knowledge-based system that utilizes a very minimal knowledge base (a list of desired frequency ratios for every difference in pitch) very intelligently was conceived by Stange et al. (2017). They express the tuning problem for every set of simultaneously sounding notes as a system of linear equations and find a tempered least squares solution with the usual numerical methods. Even though this system can hardly be called an expert system, it is still based on *a priori* musical knowledge.

As we have seen in Section 2, following the traditional rules of music (e.g. trying to ensure just intervals whenever possible) will indeed reduce the dissonance of instruments with harmonic spectra. However, we are not only dealing with traditional musical instruments here, but also with environmental noise that is not guarantied to be harmonic or exhibit any kind of structure that allows the application of musical rules. More practically, the system of Stange et al. (2017) treats every note the same, it is not able to distinguish between different timbres. Yet, the results of our analysis of the environmental noise will be a set of simple frequencies. The algorithm would treat this set of simple sine waves as a full chord of complex tones. Imagine our environmental noise contains a number of low droning frequencies. These should not affect the tuning of some chords playing in a significantly higher register, because there would be no danger of beating between the partials of the music and the droning frequencies. Yet, if we would use these droning frequencies as an input to the system described above, it would try to tune the notes of the chords to form just intervals with the drones, unnecessarily complicating their inner-musical tuning and their tuning to higher environmental frequencies that are actually relevant to them.

Ward, 1970: 418-421).

3.3 Dissonance-based adaptive tuning

Sethares (1993) described an algorithm that minimizes the dissonance of a set of complex tones by computing the gradient of the total dissonance (in terms of roughness) with respect to the fundamental frequencies of the tones and using the gradient descent method to find local minima of the dissonance curve of the sound. When Sethares implemented his algorithm (Sethares, 2002), he was able to observe the following behavior (and we will later confirm his observation): The algorithm is able to gradually reduce the roughness of a given chord in every iteration until it converges to a local optimum. For harmonic timbres this will yield the just intervals whenever possible, and otherwise a tempered compromise. Unlike systems that are based on musical knowledge, a system such as this can take environmental noise into account naturally, we just have to add the roughness between the partials of the music and every pronounced frequency of the noise in our calculation of the total dissonance and it's gradient with respect to the fundamentals of the musical notes.

4 Implementation of ATMEN

4.1 Measuring dissonance

At the core of ATMEN will be an optimization task, and at the core of every optimization task is an objective function: the dissonance measure we are trying to decrease. The field of mathematical optimization has generated a number of clever methods available and readily implemented but their performance depends heavily on the properties of the objective function they try to optimize. It will be worthwhile then to construct our function with care.

Remember that the scaling of a function is irrelevant to the number and location and of its minima and hence irrelevant for the optimization of the function. Therefore the scaling of the dissonance function we will define in the following section will be completely arbitrary, whenever there are graphs of dissonance functions given, they are scaled to easier compare the shape of the depicted dissonance functions. On the other hand, optimization will be much more efficient when we are able to find a dissonance measure that satisfies certain conditions: If possible, it should be continuous or even better continuously differentiable. It should be efficient to compute. It should not be to ragged, i.e. it should not have a great amount of small local minima between the "real" local dissonance minima we try to find, and it should not have big plateaus where the dissonance does not change either. All this properties will allow us to make use of more efficient optimization methods and will make these methods converge faster to a meaningful local minimum. We cannot elaborate the reasons for this here, but think of a game of "Topfschlagen" (a german game similar to blindfolded "Hunt the Thimble" where you have to hit a pot with a spoon while others signal if you are getting nearer by shouting "hot" or "cold"): You would not want the person saying hot or cold to change his temperature-report often and abruptly (a raged and

discontinuous temperature curve), and you would not want them to not say anything for longer periods (a plateau of constant temperature). Note that this is true regardless of the specific search strategy the blindfolded player is using. Similarly, getting the objective function right will help us regardless of the optimization methods we will choose.

4.1.1 Construction of a dissonance measure tailor-made for ATMEN

General structure: It will be wise to decompose our dissonance calculation into a number of simple functions. As we have seen above, the basic procedure proposed by Helmholtz is to calculate the simple dissonance, which typically consists of a roughness factor d and a volume factor v, for every pair of partials in the sound we want to analyze and sum them up to get the total dissonance D of the sound. So if we assign every partial of the sound some index $i \in I$ with frequency f_i and amplitude a_i, that gives us

$$D = \sum_{i,j \in I} d(h(i,j))v(i,j) \tag{1}$$

where h is some measure of distance between two partials.

Distance measure: Lets start with the distance measure h. Originally, Helmholtz thought that d depends only on the difference between the frequencies of the partials.

$$h_H(i,j) = |f_i - f_j| \tag{2}$$

We have already see that Plomp and Levelt (1965) were able to show that the difference in proportion to the critical bandwidth at the mean frequency

$$h_{PL}(i,j) = \frac{|f_i - f_j|}{cbw(\bar{f})}, \text{ with } \bar{f} = \frac{f_i + f_j}{2} \tag{3}$$

is empirically more accurate. If we choose some reasonably plausible functions for d and v (more on that later) and plot the dissonance curve of two dissonance measures with $h := h_H$ and $h := h_{PL}$ for a higher fixed tone of 880 Hz (A5), we see that the choice of h_H is not only empirically inaccurate, but also unsuited for optimization: Big plateau-like minima that do not correspond to a perceived dissonance-minimum emerge between the very narrow "real" minima bordered by sharp dissonance peaks. This is because for 880 Hz, the distance between the partials has become very big in comparison with the shape of the curve of d and the "tail" of the dissonance function (the part on the right of the maximum) is not strong enough to "fill the holes" between the partials (see Figure 6). These minima will be impossible to find for any optimization algorithm that does not start exactly inside them. Again, imagine you play *Topfschlagen* but the person giving the directions almost always says "pretty warm". As you get near the pot they say "it's getting very cold" and only when you're directly on top of the pot they confirm "warm". As you can see in the same plot, with h_{PL} we get a curve that, is more plausible as a model of our dissonance sensation and and much easier to navigate by an optimization algorithm.

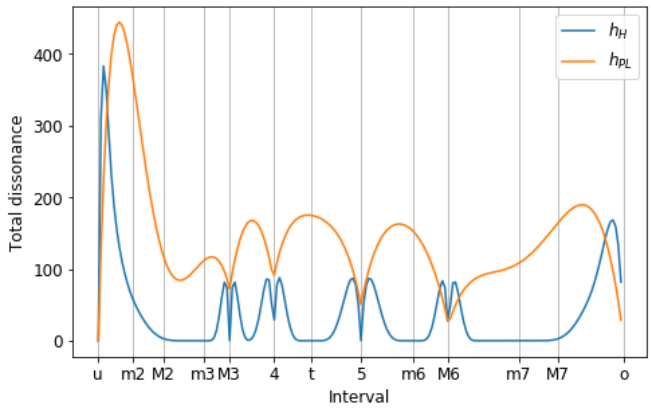

Figure 6: Dissonance curve of two dissonance measures using h_H and h_{PL} respectively. The lower tone is fixed at 880 Hz (A5), both complex tones exhibit the first five partials of a piano timbre.

Critical bandwidth: For the calculation of the critical bandwidths, we can simply use the formula by Zwicker and Terhardt (1980) that we already saw in Figure 2:

$$cbw_{ZT}(f) = 25 + 75(1 + 1.4(\frac{f}{1000})^2)^{0.69} \tag{4}$$

Volume measure: Choosing a good measure of volume is less trivial. Early theoretical texts on the beating theory of dissonance (von Helmholtz, 1968; Plomp and Levelt, 1965) did not specify how to handle partials with different volumes at all. The earliest implementations of the theory by Hutchinson and Knopoff (1978) and Sethares (1993) used the product of the amplitudes,

$$v_h = a_1 a_2, \tag{5}$$

which is straightforward but does not account for the dependency of the human perception of loudness on frequency. Whats more, the influence of moderately weak overtones gets very small and the corresponding minima get swallowed by the tails of the stronger partials (see Figure 7). More recent implementations recommend some measure of volume that is modeled after the human hearing sensation: loudness in sone (Sethares, 2005: 435-346) or loudness level in phon (Dillon, 2013; Bernini and Talamucci, 2014). As you can see in Figure 7, this not only accounts for the frequency dependency of the human hearing range but also mitigates the problem of the vanishing effect of weak partials. Roughly, this is because the translation into these scales contains a logarithmic transformation that makes the different values more similar. To calculate these loudness measures for a simple tone, some preliminary definitions are needed: For a sine wave $a\sin(2\pi ft)$ with amplitude a in Pa and frequency f in Hz, the *effective pressure* p is defined as:

$$p = \sqrt{\frac{1}{T}\int_0^T (a\sin(2\pi ft))^2 dt} = \frac{a}{\sqrt{2}}, \text{ with } T = \frac{1}{f} \tag{6}$$

The *sound pressure level* (SPL) L_p is given in dB and is defined by

$$L_p = 20 \log_{10}(p/p_{rt}), \text{ with } p_{rt} = 20\ \mu\text{Pa}, \tag{7}$$

where p_{rt} is the effective pressure at which a reference tone (a sine wave at 1 kHz) is barely heard. The *loudness level* L_N takes into account that humans do not experience loudness independently of the frequency of the sound: We are particularly sensible in the region from 1 to 5 kHz and are unable to hear sounds that are below 20 Hz or above 20 kHz. The loudness level of any sound is defined as the SPL of a reference tone at 1 kHz that is perceived to have the same loudness as that sound.

How to predict the loudness level of a simple tone from its SPL is a difficult question both empirically and in terms of its mathematical modeling (Suzuki and Takeshima, 2004: 924). We will settle for a rough approximation that Parncutt (1989: 82) calls the *auditory level* A and that is defined simply as the SPL above the threshold of hearing[3] (see Figure 8):

$$A = max(L_p - L_{pt}, 0), \tag{8}$$

where L_{pt} is the SPL at which our simple tone would be barely audible. We can approximate this threshold as a function of the frequency of our simple tone using a formula from Terhardt (1979):

$$L_{pt}(f) = 3.64 f'^{-0.8} - 6.5 \exp(-0.6(f' - 3.3)^2) + 10^{-3} f'^4, \text{ with } f' = \frac{f}{1000} \tag{9}$$

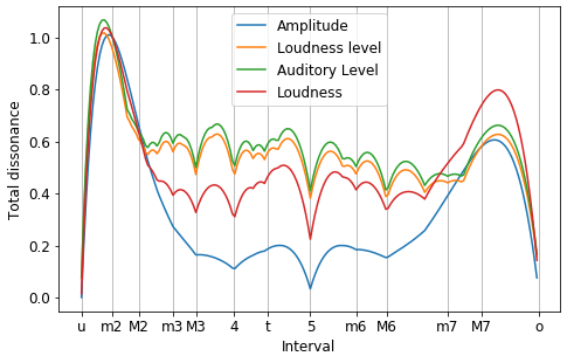

Figure 7: Comparison of different measures of volume.

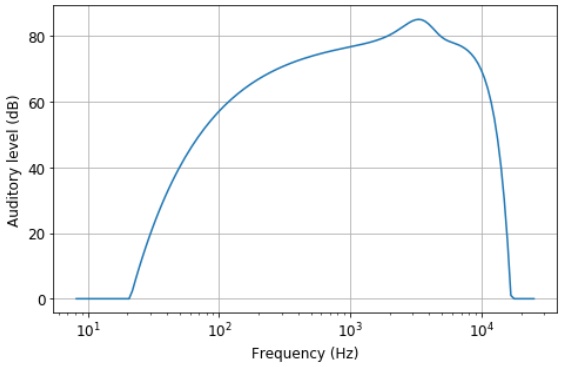

Figure 8: Auditory level of a sine wave of fixed amplitude $a \approx 0.3$ Pa as a function of the frequency of the wave.

Volume aggregation: Above we assumed that we aggregate the volumes of our partials by multiplying them, but other aggregation methods are possible: Sethares (2005: 435) defines the volume factor as the minimum of both volumes

$$v_S = \min(v_1, v_2) \tag{10}$$

[3]This is basically the loudness level if we ignore the fact that the equal loudness contours get flatter for higher loudness levels.

and Dillon (2013) takes the square root of the product

$$v_D = \sqrt{v_1 v_2}. \tag{11}$$

Both measures generally have the effect of making the volume factors of the different pairs more similar than using the product and thus mitigate the effect of the vanishing relevance of weak partials (see Figure 9) when using the amplitude. When using the auditory level there is hardly any difference. ATMEN uses the minimum simply because it is the fastest to compute.

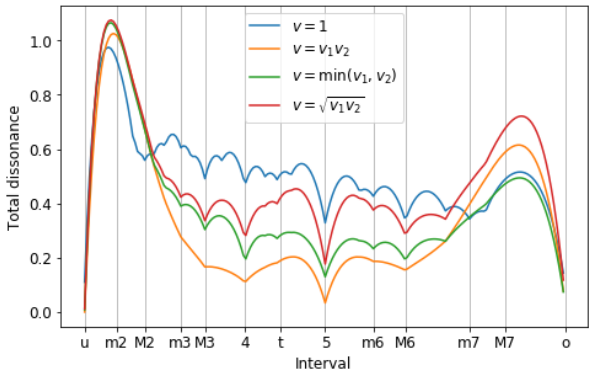

Figure 9: Comparison of different aggregation methods for amplitude.

Roughness meassure: The roughness measure that was used for the plots up to this point was defined by Dillon (2013):

$$d_D(h) = 4.906\ h\ (1.2 - h)^4 \text{ for } h < 1.2, \text{ else } 0. \tag{12}$$

It increases immediately with a steep slope which results in sharp minima on the dissonance curve. This may look more defined to a human eye but is actually hard to navigate for most optimization algorithms, since the function (as a function of f_i) is non-convex and not continuously differentiable at these points (see Figure 10). The following definition from Richard Parncutt (Bigand et al., 1996: 128) on the other hand produces a dissonance curve that is much less defined and has a tendency to merge close minima (see Figure 12).

$$d_P(h) = (h \exp(-4h))^2 \text{ for } h < 1.2, \text{ else } 0 \tag{13}$$

Apart from that, this definition is especially well suited for optimization: Its gradient is continuous, if we ignore the small discontinuity at $h = 1.2$ (see Figure 11). Even better – for h near zero (i.e. when to partials are already near together and we just have to push them a little bit to get a perfect fit), this function looks approximately like a U-shaped parabola (see Figure 13), which is very beneficial because the most efficient optimization methods, so called second order optimization algorithms, approximate the objective function as a second degree Taylor polynomial (which is itself a quadratic

function) so the approximation will be reasonably good where the objective function is approximately quadratic. On the machine this was implemented on, the conjugate gradient descent method provided by the `optimize` module of `scipy`, converges on average in 25.5 ms for d_P, in 60.2 ms for d_D. For this reasons, d_P is our best option for now.

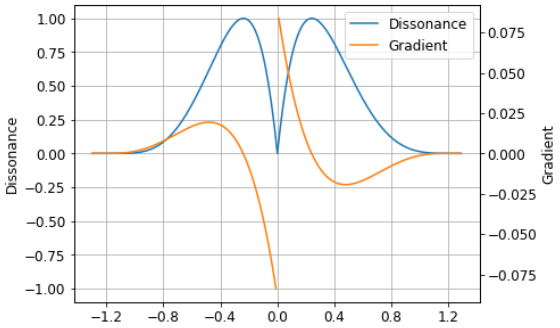

Figure 10: d_D and its gradient.

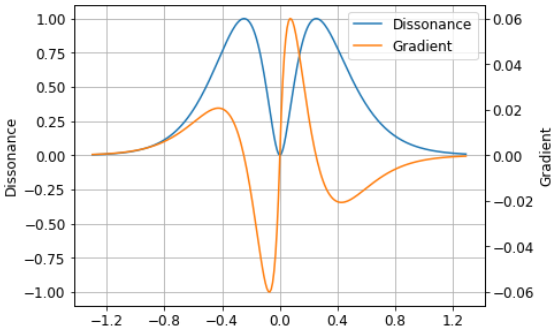

Figure 11: d_P and its gradient.

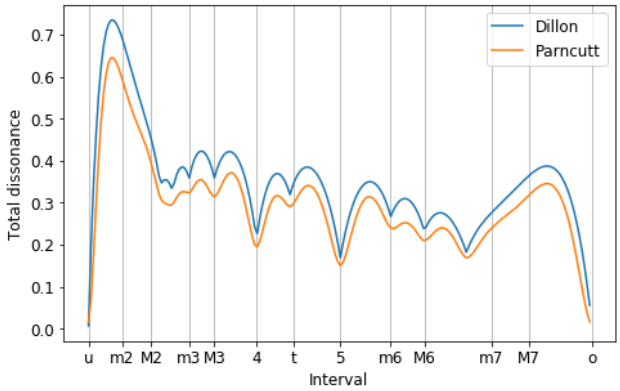

Figure 12: Comparison of dissonance curves using $d := d_D$ and $d := d_P$

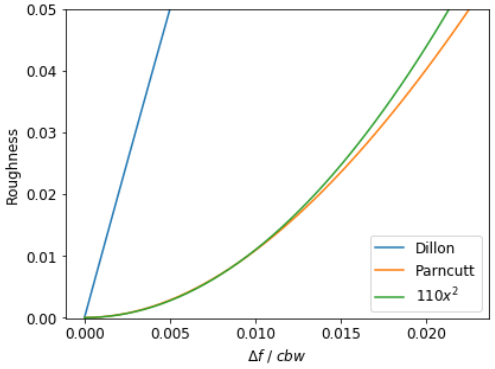

Figure 13: Roughness of a pair of partials with h very small.

4.1.2 Simplification of the dissonance measure and derivation of its gradient

Computing the gradient with respect to the fundamental frequencies of the complex tones of the function "as is" will be complicated and slow, primarily because of the dependency of the CBW and the auditory level on the frequency of the tones. Hence, before we continue with the derivation of the gradient of our dissonance measure later in this section, we will look at some ways to simplify it that will make the derivation much easier and reduce the computation time needed in every iteration of the optimization algorithm.

Quasi-constants: We are not trying to find a *global* minimum of our dissonant curve, that would be trivial: Just push every tone outside the human hearing range. This would be very efficient in reducing the dissonance of the sound indeed (What we cannot hear cannot be dissonant!), but would make for an useless musical instrument. To maintain the integrity of the original musical material, we do not want our tuned frequencies to differ from the originally intended ones by more than a third of a semitone. This is still more than sufficient to reduce the dissonance of our sound significantly – the equal tempered third differs from a just third by only 0.14 semitones.

As a consequence some of the values we defined in the last section will be practically constant with respect to the fundamental frequencies during the optimization: If we evaluate cbw_{ZT} at ϵf_i with $\epsilon = 2^{1/36}$ (the interval of a third of an equal tempered semitone) instead of f_i, the relative error $(cbw_{ZT}(\epsilon f_i) - cbw_{ZT}(f_i))/cbw_{ZT}(f_i)$ will be less than 3% and so will the relative error of the auditory level $A(i)$ for an amplitude of $a_i = 0.2$ Pa and f_i in the range of 40 to 11 kHz (see Figure 14). Of course, as f_i gets near the threshold of hearing and $A(i)$ goes to zero, the relative error of the auditory level will diverge to infinity and will only return to zero when both $A(i) = 0$ and $A(\epsilon f_i) = 0$. This means that the error can push frequencies that are close to the threshold of hearing outside (or inside) the human hearing range. If we accept that risk, we can compute the values $cbw_{i,j} = cbw_{ZT}(i, j)$ and $v_{i,j} = \min(A(i), A(j))$ once for every pair of partials and treat them as constants during optimization.

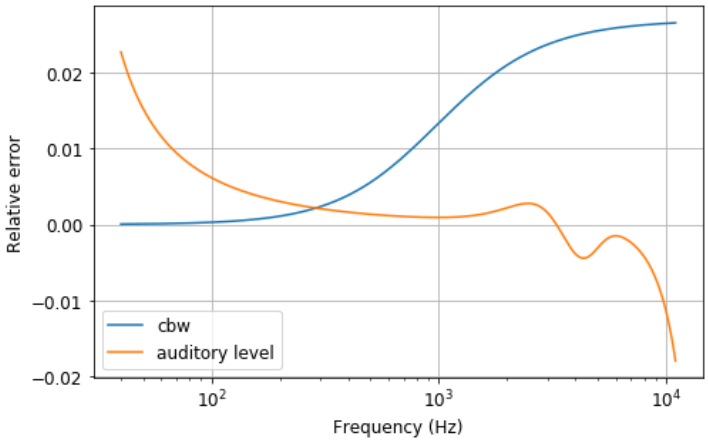

Figure 14: Relative error of the CBW and the auditory level (for $a_i = 0.2$ Pa) with respect to the frequency at which it is evaluated.

Reducing the number of summands: We can further reduce the computation time of the evaluation of our target function and its gradient by reducing the number of terms of the sum in Equation (1), i.e. finding some subset $R \subset I \times I$ such that we do not lose important information about our dissonance when we sum over pairs of partials in R only. Currently we evaluate all $|I \times I| = (nm)^2$ pairs of partials for n complex tones with m partials each. Since h and v are symmetric and $d(0)$ (the dissonance of the unison) is always zero, it is sufficient to calculate the simple dissonance of every

pair of partials only once, and omit the summands with $i = j$. This would leave a total number of $0.5nm(nm-1)$ pairs of partials to calculate.

The dissonance of pairs of partials that belong to the same complex tone is not constant with respect to the the fundamental of that tone because the dissonance of two simple tones generally decreases as their frequencies increase, even if the interval stays the same. Hence, increasing all frequencies as much as possible would be a good strategy to reduce the dissonance of the sound, but it is obviously not a desirable behavior for a musical instrument. It will therefore be a good idea to omit these inner pairs, to reduce the incentive of the algorithm to increase the frequencies of all tones regardless of their interval. Every complex tones has $0.5m(m-1)$ inner pairs. Omitting these leaves us with $0.5nm(nm-m)$ pairs.

Furthermore, most pairs of partials will yield a basic dissonance of zero because they are either not close enough to produce any roughness (i.e. $d = 0$) or one of the partials is outside the human hearing range (i.e. $v = 0$). Since we already found v to be quasi constant in the last section, a pair with $v = 0$ will never be relevant and can be omitted from R. Similarly, if the distance between two partials is bigger than approximately 146% of CBW, it will never be smaller than 120% of CBW during tuning and hence their roughness will always be zero (see Figure 15). In the end we find the set of relevant pairs R to be the set of all pairs of partials $(i, j) \in I \times I$ such that

- $i \neq j$ and $(i, j) \in R \Rightarrow (j, i) \notin R$
- i and j are not partials of the same complex tone
- $v_i > 0$, $v_j > 0$, and $h(i, j) < 1.46$.

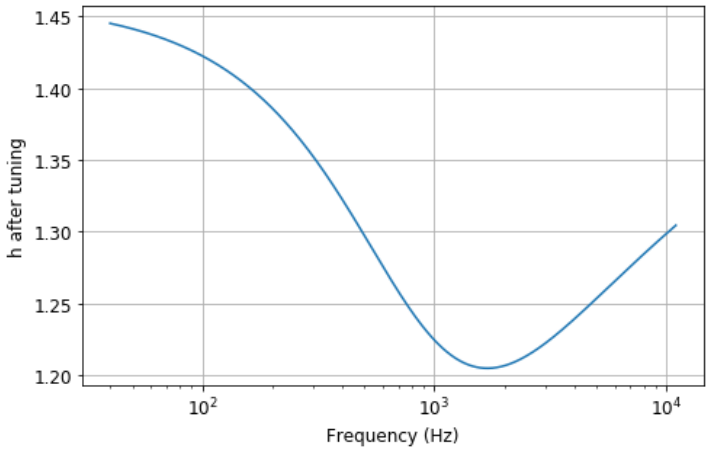

Figure 15: Distance h_{PL} of two partials that had a distance of $h_{PL} = 1.46$ before tuning after tuning the lower one 1/3 semitones higher and the higher one 1/3 semitones lower as a function of their mean frequency (before tuning).

Since the last condition depends on the specific set of partials I it is hard to say exactly how much it will help to reduce R, but in general it is by far the most efficient factor. For example, an A major chord on A4 in closed position with added octave

in 12TET (A4, C♯5, E5, A5) consists of $n = 4$ complex tones. Lets say each complex tone consists of $m = 11$ audible harmonic partials. The first two condition reduce the number of relevant pairs from 3025 to 1210. The last condition reduces it to 140.

Derivation of the gradient: Now that we have defined and simplified our dissonance measure, we can describe the computation of its gradient with respect to the fundamental frequencies of the tunable complex tones. Say we want to compute the partial derivative of the total dissonance D with respect to the frequency f_i of some partial i that is not fixed but part of a tunable complex tone.

$$\frac{\partial D}{\partial f_i} = \sum_{j \in R_i} \frac{\partial}{\partial f_i}(d(h(i,j))v(i,j)) \text{ with } R_i = \{j \in I | (i,j) \in R \vee (j,i) \in R\} \tag{14}$$

Since we consider $v(i,j) = v_{i,j}$ a constant during tuning we get

$$\frac{\partial}{\partial f_i}(d(h(i,j))v(i,j)) = v_{i,j}\ \frac{\partial d}{\partial f_i}(h(i,j)) = v_{i,j}\ \frac{\partial d}{\partial h}\ \frac{\partial h}{\partial f_i}. \tag{15}$$

If we ignore the small discontinuity at $h = 1.2$, the partial derivative of the roughness d_P with respect to the distance h is

$$\frac{\partial d_P}{\partial h} = 2h \exp(-8h)\ (1 - 4h) \text{ for } h < 1.2, \text{ else } 0. \tag{16}$$

Remember that we treat $cbw(i,j) = cbw_{i,j}$ as constant also, hence the derivative of h_{PL} with respect to f_i (ignoring the discontinuity at $f_i = f_j$) is

$$\frac{\partial h_{PL}}{\partial f_i} = \frac{1}{cbw_{i,j}} \frac{\partial}{\partial f_i}(|f_i - f_j|) = \frac{1}{cbw_{i,j}} \text{ for } f_i > f_j, \text{ else } \frac{-1}{cbw_{i,j}}. \tag{17}$$

Now we can compute the derivative of D with respect to the fundamental frequency f_a of a fundamental a:

$$\frac{\partial D}{\partial f_a} = \sum_{i \in I} \frac{\partial D}{\partial f_i}\ \frac{\partial f_i}{\partial f_a} = \sum_{i \in I} r_i \frac{\partial D}{\partial f_i} \tag{18}$$

where r_i is the relative position of partial with respect to the fundamental frequency, i.e. $f_i = r_i f_a$, if i is a partial of the tunable complex tone a, else $r_i = 0$.

Drift-corrected gradient: The gradient derived above still suffers from the "higher is better"-problem we briefly mentioned earlier in this chapter: The same interval will generally yield a lower dissonance value if the interval is played at a higher frequency, and accordingly the negative gradient of our dissonance function will, on average, be positive, pushing the fundamental frequencies upwards during optimization.

We can clearly see this behavior in practice when we use the conjugate gradient method with the gradient we derived above. For example, if we want to tune an equal tempered A major chord (440 Hz, 554 Hz, 659 Hz) and we allow the algorithm to search

for local optima within some broad interval around the original frequencies, say a fifth up and down ($f' \in (2/3f, 3/2f)$), then the tuned frequencies will be approximately 659, 824, and 989 Hz – a justly tuned E major chord. The algorithm not only fine tunes the relative position of the complex tones but above all pushes every frequency to the upper bound of the allowed range.

It is important to stress that this is not a bug of our dissonance measure but a feature of our actual perception of dissonance (von Helmholtz, 1968: 286). In (jazz) arrangement, for example, it is common to hand students a list of "lower interval limits" that specify for every interval the lowest pitch at which it should be used (Pease and Freeman, 1989: 62). Still, tuning every note uniformly sharper is not the kind of behavior we want in a musical instrument, so we have to find some kind of heuristic that prevents this uniform frequency shift.

Within orchestras, pitch drift is usually prevented by tuning every instrument to some fixed reference pitch, usually provided by a tuning fork. Interestingly, before concert pitch was fixed internationally to A4 := 440 Hz, there were multiple incidence of global pitch drift ("pitch inflation"): Each orchestra, trying to sound brighter than the next, increased their reference pitch. This way the pitch of A4 rose from 422 Hz to 450 Hz during the nineteenth century for example (Haynes, 2002).

To solve the problem of pitch drift in his adaptive tuning system, Sethares (2002) used fixed frequencies that are not actually played but only added to the computation of the dissonance and serve as an anchor to the flexible frequencies, like a virtual tuning fork. Another way to prevent pitch drift would be to allow the algorithm to optimize each frequency only within a very small interval like a $1/3$ semitones around the equal tempered frequency. Both options are not applicable for ATMEN: While we don't want our musical notes to drift to much from their intended frequencies just because "higher is better", we actually want to have this kind of flexibility to tune to frequencies in the environmental noise. Hence we will have to find a new solution to the problem that is compatible with the aim of tuning to environmental noise.

We could hope that the fixed frequencies of the noise sufficiently anchor the flexible frequencies but this is not generally true – there might be no pronounced frequencies in the environmental noise at all or they might be to weak or to far away from the musical notes to form relevant pairs that are strong enough to overwrite the "higher is better" incentive. A sophisticated solution would be this: Think of our tuning problem as a graph: Draw a node for every complex tone and every fixed frequency and an edge between two nodes if their connection is strong enough to prevent pitch drift (deciding this is obviously not trivial). Then, for every isolated subgraph, make sure there is at least one node representing a fixed frequency by manually fixing a musical tone if there is no environmental frequency in the subgraph already. This ensures that every flexible frequency is sufficiently anchored when we start our tuning procedure.

I want to propose a second method here that addresses the root of the problem more directly, not just its consequences: Consider the gradient of the roughness of a pair of

partials (i, j) with respect to the frequency f_i, if we fix the interval between them:

$$\frac{\partial d}{\partial f_i}(h(f_i, r f_i)) \tag{19}$$

where the interval $r = f_j/f_i$ is considered constant with respect to f_i. We can think of this term as the isolated incentive to tune higher regardless of the relative position of the partials. If we subtract this term (weighted by some correction factor c) from our original gradient, we obtain a drift-corrected gradient as a substitute for Equation (15):

$$\begin{aligned}
&\frac{\partial d}{\partial f_i}(h(f_i, f_j)) - c\frac{\partial d}{\partial f_i}(h(f_i, r f_i)) \\
=&\frac{\partial d}{\partial h}(\frac{\partial}{\partial f_i}(h(f_i, f_j)) - c\frac{\partial}{\partial f_i}(h(f_i, r f_i))) \\
=&\frac{\partial d}{\partial h}\frac{1}{cbw}\begin{cases}(1 - c(1 - r)) & \text{for } f_i > f_j \\ (-1 + c(1 - r)) & \text{else}\end{cases} \\
=&(1 + c(r - 1))\ \frac{\partial d}{\partial f_i}(h(f_i, f_j))
\end{aligned} \tag{20}$$

If we choose the correction factor $c = 1$ we will subtract the full correction term both when we compute the gradient with respect to f_i and when we compute the gradient with respect to f_j, thereby inversing the drift effect, pushing the frequencies uniformly downwards. But if we divide the correction term equally on both partials ($c = 0.5$) and try to tune the A major chord again, we receive the frequencies 441 Hz, 551 Hz, and 661 Hz – a justly tuned A major chord with a mean change in frequency of only 0.0557 Hz. On the other hand, if we add a fixed frequency at 460 Hz, the flexible frequencies are tuned to 460 Hz, 575, and 690 – a justly tuned A major chord, transposed approximately 3/4 semitones upwards. This amount of flexibility would not be possible with the anchoring methods described above. Unfortunately the full evaluation (Section 5) will expose that our algorithm is not always that well behaved.

4.2 Technical details of ATMEN

To demonstrate the feasibility of the ideas described above, ATMEN was initiated as an open-source project and made available on Github [4]. The software allows us to hear and play music that is adapted to environmental noise in real time. Examples that demonstrate the application of the software can be found in the repository. Note that this is meant to be an educational and somewhat experimental application, rather than a tool to produce music professionally. Creating a great sounding and versatile synthesizer with a low latency was not a focus of this project.

The project was done in Python, and utilized a number of different open source libraries: `mido`[5] was used to receive midi input and read midi files, `pyaudio`[6] was

[4] `https://github.com/ArneKramerSunderbrink/adaptivetuning`
[5] `https://mido.readthedocs.io`
[6] `https://people.csail.mit.edu/hubert/pyaudio`

used to record audio and play back audio files, `sc3nb`[7] was used to communicate with a SuperCollider synthesis server (SuperCollider[8] is an open-source environment for for real-time audio synthesis), `numpy`[9] and `scipy`[10] were used for audio analysis and optimization, and `threading`[11] was used to manage threads.

4.2.1 Basic structure and signal flow of the system

The different tasks of ATMEN can be assigned to the following areas:

- Midi processing: Receive midi messages or play a midi file, forward the information that a particular note should be played or stopped to an audio generator, request a new tuning on note-on messages.

- Audio analysis: Record audio or play an audio file, analyze the audio for pronounced frequencies, make the currently sounding frequencies available for adaptive tuning.

- Dissonance reduction: Given a set of complex tones with a certain timbre and a set of fixed frequencies, fine-tune the fundamental frequencies of the tones to reduce the total dissonance of the notes and the fixed frequencies and forward the tuned frequencies to the audio generator.

- Audio generation: Register and store information about the currently sounding notes (their pitch, timbre, amplitude, and current tuning) and make it available for dissonance reduction, communicate with the SuperCollider server to play notes according to the stored information.

In the implementation of ATMEN, this tasks are realized in four classes named `Midiprocessing`, `Audioanalyzer`, `Dissonancereduction`, and `Audiogenerator` accordingly (see Figure 16). A fifth class simply named `Tuner` manages their interaction in a multithreading environment. This is necessary because we have a number of simultaneous in- and outputs and some time-consuming calculations that have to run in parallel without blocking each other: We do not want midi processing to stop while recording audio or searching for optimized frequencies.

[7] `https://github.com/thomas-hermann/sc3nb`
[8] `https://supercollider.github.io`
[9] `https://numpy.org/`
[10] `https://scipy.org`
[11] `https://docs.python.org/3/library/threading.html`

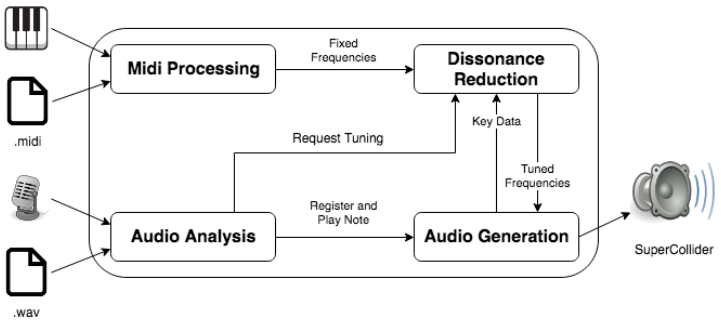

Figure 16: Basic structure of ATMEN.

The following data is shared among the threads: a list of all currently registered notes with all the corresponding information (pitch, timbre, amplitude, current tuning), a list of frequencies found in the environmental noise, and a number of locks and booleans that are used for communication between the threads, in particular requesting a tuning and stopping the system. The typical flow of information between these threads can be characterized as follows (see Figure 17):

- The audio analyzer thread is constantly recording audio or reading from an audio file and stores the found frequencies.

- The midi processing thread is constantly waiting for midi signals from a controller or playing a midi file. When receiving a new message, it starts a new message handler thread.

- When a message handler thread is started for a note on message it registers the new note and requests a tuning, then it sleeps for a predefined lag time. The lag is chosen appropriately to give the tuning thread enough time to compute an optimal tuning. When the time is up, the message handler tells SuperCollider to play a new note at the frequency that was updated by the tuning thread in the meantime.

- If the tuning thread receives a tuning request, it acquires the information it needs from the list of currently registered notes and the list of frequencies found in the environmental noise, calculates the dissonance minimizing tuned frequencies, and updates the tuning in the list of registered notes. The tuning thread also tunes in regular time intervals, even if no tuning was requested by a midi handler, to account for changes that are not signaled by a midi message, like a change in the environmental noise or the amplitude of a currently playing note.

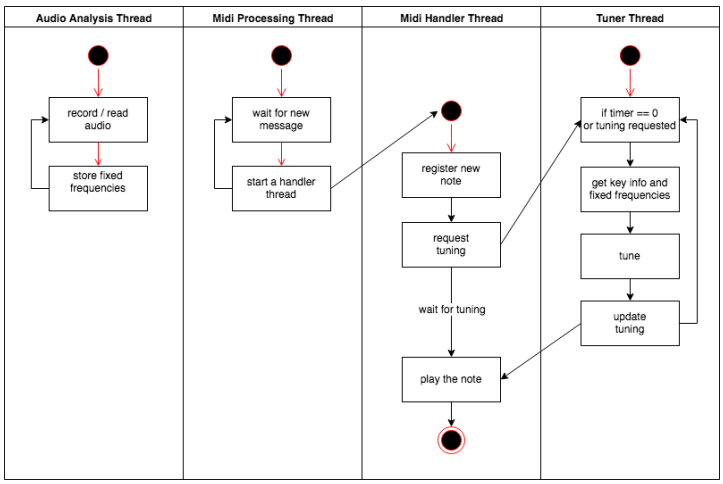

Figure 17: Signal flow of ATMEN when receiving a note-on message.

4.2.2 Properties of a tone in ATMEN

The audio generation component of ATMEN uses additive synthesis, which lends itself to our endeavor quite naturally: a timbre with n partials is defined by a list of relative positions of partials $p_1, \ldots, p_m$ and their corresponding relative amplitudes $a_1, \ldots, a_m$. When we request a note at frequency f and amplitude a, the synthesizer creates m sine waves with frequencies $p_1 f, \ldots, p_n f$ and amplitudes $a_1 a, \ldots, a_n a$, respectively and emits the sum of these sine waves. To control the dynamics of each tone, an *adsr* envelope is used (see Figure 18).

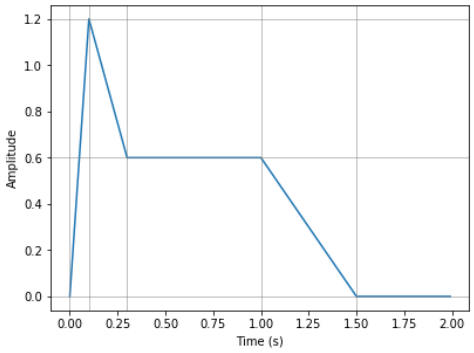

Figure 18: An adsr envelope as it is used in ATMEN with an attack time of 0.1 s, decay time of 0.2 s, sustain level of 0.5, and release time of 0.5. A key is pressed at $t = 0$ with total amplitude $a = 1.2$ and released at $t = 1$

The corresponding parameters (amplitude, frequency, partial positions, partial amplitudes, attack time, decay time, sustain level, and release time) are stored for every registered note in a `KeyData` object. Furthermore, a `KeyData` object stores a timestamp for the time it was pressed and the time it was released and uses these timestamps together with the envelope parameters to calculate the current amplitude of the tone when asked.

Note that the `KeyData` objects are created before the playback of the tone starts, which only happens after the lag time. This way, we can tune the notes using the stored information before they are actually sounding.

4.2.3 Analyzing environmental noise

In Section 4.2.2, we specified the properties of the musical tones we want to tune. The other input for our optimization algorithm are the fixed frequencies we have to retrieve from the environmental noise. This is how they are detected: We cut the signal (i.e. a list of samples) into blocks, downsample the signal and normalize it by subtracting the mean value and dividing by the maximal value. Then we use the `rfft` method from `numpy`'s `fft` module that computes the discrete Fourier Transformation for real valued input (FFT) and returns a list of complex values whose magnitudes represent the power present in every frequency bin.

Since this implementation of the FFT for real values computes only one half of the power spectrum (the other half would be redundant because the power spectrum of a real signal is symmetric) and because `rfft` returns an unnormalized spectrum, we have to multiply every magnitude by 2 and divide it by the number of samples (after downsampling) to retrieve the approximated amplitudes. We disregard bins that correspond to frequencies outside the human hearing range (bellow 20 Hz or above 18000 Hz) and use the function `find_peaks` from `scipy`'s `signal` module to identify pronounced frequencies and return the 10 most pronounced (see Figure 19).

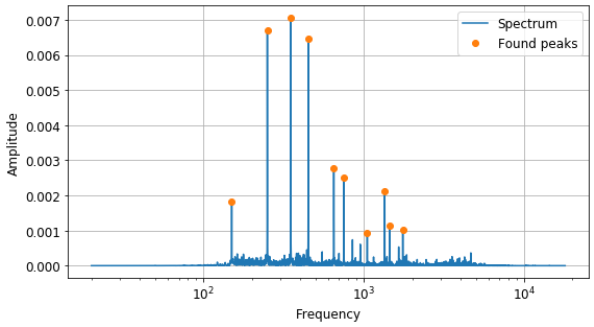

Figure 19: Spectrum of a huming guitar amplifier with the ten most pronounced frequencies marked.

As it turned out[12], a blocksize of $2^{15} = 32768$ samples, which corresponds to approximately 743 ms at the standard sample rate of 44100 samples per second, and a downsample factor of 8, i.e. taking only every eighth sample, works best for our purpose. This gives us a maximal frequency resolution (the width of the frequency bins) of approximately 0.178 Hz. The machine used to test the frequency detection did not need more than 2 ms to compute the FFT and find the pronounced frequencies in the spectrum. Yet the rather large block size means that we will not be able to follow

[12]The interested reader is encouraged to have a look at the notebook `pitch_detection_test` in the ATMEN repository to retrace the evaluation of the audio analysis parameters.

fast changing frequencies in the environmental noise. This is not a problem since tuning to these unstable frequencies would result in an unmusical gliding of the notes we want to avoid anyway.

5 Evaluation of the behavior of ATMEN

The following evaluation of ATMEN is first and foremost a description of its behavior. To evaluate if the system is able to achieve the goals defined in Section 1, i.e. if tuning music with ATMEN in a noisy environment reduces the sensation of dissonance and enhances the positive effects the music has on listeners, an empirical study would be needed. However, this effort will only be worthwhile if we made sure that the system reliably shows the behavior we hope to cause these positive effects. As we will see, this is not entirely the case.

5.1 Tuning without environmental frequencies

In Table 2 you can see the interval structure of different chords tuned with ATMEN (starting from equal tempered frequencies). For easier comparison, interval size is given in cents (rounded). One cent is a hundredth of an equal tempered semitone, i.e. c cents correspond to a frequency ratio of $r = 2^{c/1200}$. For comparison: Two pure tones generate maximal roughness at about 80 cents difference (for frequencies > 500 Hz) and the smallest difference humans can detect is approximately 5 cents (Fastl and Zwicker, 2007: 186). In the highlighted rows, the dissonance calculated according to Section 4.1 is given.

- For the most consonant chords (major, minor, suspended second, minor seventh), the ATMEN fine-tunes the notes to match the just tuning very accurately and consequently yields similar dissonance values. Note that, other than a fixed just scale, ATMEN delivers this just tunings for arbitrary transpositions of the chord.

- In the major seventh chord, the major third and sixths are slightly flattened to push the major seventh away from the first overtone of the tonic.

- Similarly in the dominant seventh chord (C7), the minor seventh is detuned significantly to the peculiar interval between the major sixth and the minor seventh we already witnessed in Figure 5.

- The biggest violation of harmonic integrity can be observed at the chord with the minor second: When we do not limit its search range manually, ATMEN decides to detune it radically to a completely different chord (a major chord in first inversion) that is undoubtedly more consonant but not at all the intended chord.

- The most interesting tuning is that of the pentatonic cluster chord (C, D, E, G, A). Remember that we have shown in Section 3.1 that it is impossible to tune all intervals of this chord to small integer ratios. If we choose all intervals between C and the other tones just, we get a fifth between D and A that is 22 cents flat in comparison with a just fifth (3 : 2). The biggest deviations from just tuned intervals of the equal tempered version (the major thirds and the sixth) are only 16 cents off. However, the arguably best compromise is provided by ATMEN: Its worst intervals (major second C-D, major sixth C-A, and minor third E-G) are only 10 cents of.

Chord	Interval	ET	JI	AT
C maj		510	496	496
	c-e	400	386	386
	c-g	700	702	702
C min		514	499	499
	c-e♭	300	315	316
	c-g	700	702	702
C sus 2		496	493	490
	c-d	200	204	210
	c-g	700	702	706
C sus 2♭		690	662	446
	c-d♭	100	111	388
	c-g	700	702	887
C maj 7		1062	1020	1014
	c-e	400	386	379
	c-g	700	702	699
	c-b	1100	1088	1076

Chord	Interval	ET	JI	AT
C min 7		995	979	979
	c-e♭	300	315	314
	c-g	700	702	702
	c-b♭	1000	1017	1017
C7		1062	1034	990
	c-e	400	386	387
	c-g	700	702	703
	c-b♭	1000	1017	968
Pent. clust.		1707	1743	1702
	c-d	200	204	193
	c-e	400	386	392
	c-g	700	702	696
	c-a	900	884	895

Table 2: Interval structure of different chords tuned with equal temperament (ET), just intonation with tonic C (JI), and ATMEN (AT), in cents. The highlighted rows contain the rounded dissonance as defined in Section 4.1.

It is fair to say that ATMEN is very successful in tuning single chords without added noise. However, when tuning a succession of chords or a chord with a melody on top, two problems become apparent: The system is not able to distinguish tones that are harmonically relevant and those that are mere passing tones. For example if we play the simple descending melody G5, F5, E5 over a held C major chord (C4, E4, G4), ATMEN will not treat F5 as a melodic passing tone with little relevance to the harmony, but as a chord tone, therefore it tries to tune the the dissonant chord C4, E4, G4, F5 by lowering the frequency of E4 to 315 Hz to push its first overtone away from F5 at 703 Hz. This effectively turns our C major chord into a C minor chord. The second problem aggravates this: Since the melody notes surrounding the F5 are consonant with the C major chord, E4 is tuned to a just major third of 327 Hz on this tones. While the jumping between the major and the minor third successfully

reduces the "vertical" dissonance on the three time points in isolation, it sounds very irritating in succession. A similar problem arises when we play different chords in immediate succession such that the release time of one chord is not over before the next chord begins. In this situation we will hear the notes of both chords together for a short moment, and ATMEN will try to tune this polychords as best as it can, resulting in some rather extreme tunings. For example, if we play F major and G major in succession like this, for a brief moment the C of F major and the B of G major will sound together and ATMEN will try to push them away from each other. Again, this effect is all the more apparent, as the tuning of these tones in the brief moment of transition will vary noticeably from their tuning in isolation (see Figure 20).

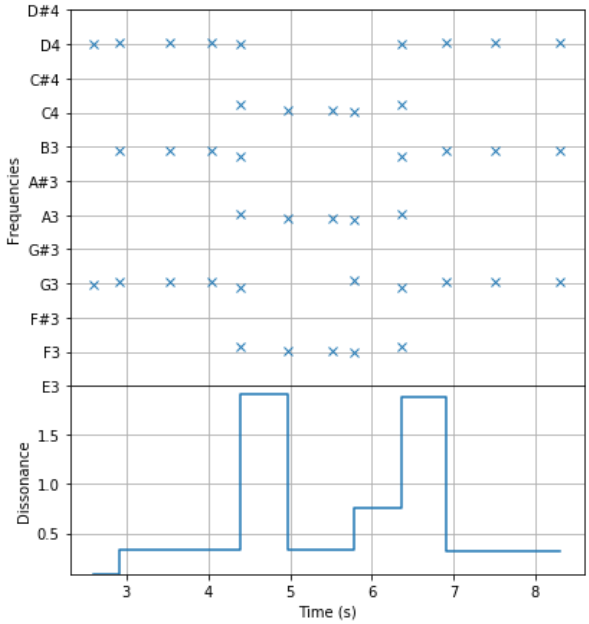

Figure 20: Tuning a succession of chords (G major, F major, G major). Notice the tuning of B3 and C4 when all six notes sound simultaneously on the transition points. Ticks on the upper y-axis correspond to equal tempered frequencies.

There are different possible methods to mitigate the described problems without manually forcing the frequencies to stay in small intervals around their original values. We could add a virtual echo of the previously played tones to the fixed frequencies as an incentive not to change the tuning of previously played notes to much from chord to chord. This is very similar to the concept of *intonational memory* used by Stange et al. (2017: 11). However, we have already seen that our own system is prone to overreact to dissonant tone clusters that can occur when the notes of a previous chord fade into the next, hence such a solution would have to be carefully tuned to harmonize with ATMEN.

5.2 Tuning to fixed frequencies

If we look at the basic shape of the roughness curve for two simple tones (Figure 11) we see three different areas: at a difference of 0-25% of CBW the tones attract each other, at 25-120% they repel each other, and for a difference of more than 120% of CBW they are indifferent towards each other (see Sethares, 1994: 13). If we play some chord (say A4, C♯5, E5) and add single a fixed simple tone at different frequencies, we find the exact same behavior (see Figure 21):

- If the frequency of the fixed tone is below 300 Hz, it does not influence the tuning of the chord at all.
- As it approaches the lowest partial of the music (440 Hz) it repels the chord upwards and since there is nothing above the chord to inhibit its rise, the results are rather chaotic.
- When it is close enough to 440 Hz it suddenly starts attracting the lowest partial and the chord rises together with the fixed tone in a more controlled manner.
- When it rises out of the attraction area it briefly pushes A4 downwards (only discernible in the zoomed plot in Figure 22) and then becomes irrelevant for all musical frequencies again.
- As the fixed tone starts getting in the range of the second partial of the chord (the fundamental frequency of C♯5 at 554 Hz) it begins to push the partial upwards, but this time more controlled because C♯5 gets also tuned to A4 below the fixed frequency.

This cycle of attraction and repulsion repeats as the fixed frequency wanders through the partials of the music, until it is high enough to rise above the range of all musical partials.

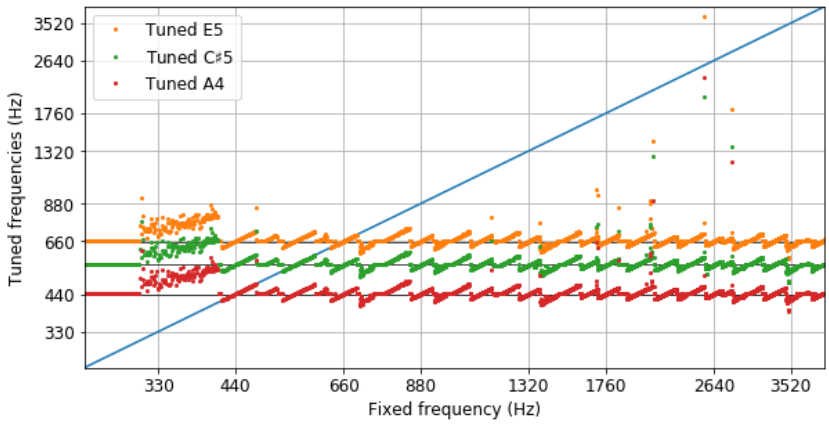

Figure 21: A single fixed frequency affects the tuning of a chord differently, depending on its position relative to the partials of the chord.

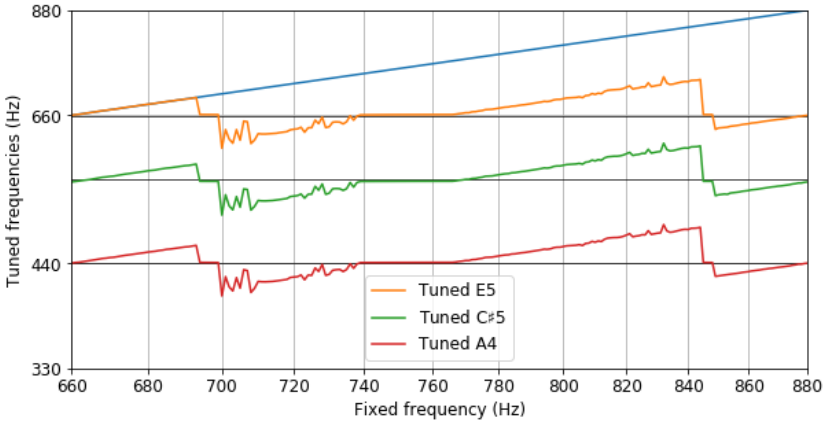

Figure 22: Same curve as in Figure 21 but zoomed to the range of 660 to 880 Hz.

In Figure 22 we see a fourth kind of behavior between attraction and repulsion: For an interval of 2 to 5 Hz around 697 Hz, the tuning of the chord seems to become indifferent to the fixed frequency. This difference corresponds to a maximum of the roughness curve where the gradient is indeed zero. This behavior is surprising nevertheless, because these maxima are not plateaus and balancing on such a local maximum is similar to balancing two marbles on top of each other – yet these balancing behavior can be observed very consistently with different optimization methods and even when adding random noise to the objective function and its gradient. What's more, on the transition into and out of this balancing areas, the chord frequencies sometimes explode into completely removed areas (see the higher frequencies in Figure 21). To completely fathom the cause of this behavior, we would likely need to implement our own minimization method to pin down the exact point where it fails.

In a tests with real world noise (a humming guitar amplifier) similar behavior could be observed: If possible ATMEN tried to align partials of the music with the environmental noise (see Figure 23), if noise and music were not near enough to attract each other, the system tried to separate the frequencies further often resulting in unpredictable and unstable tunings.

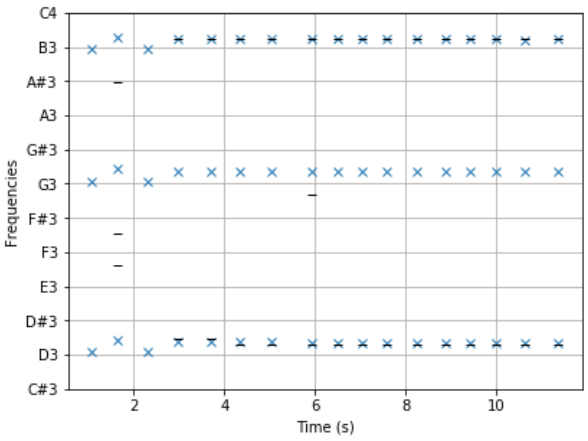

Figure 23: A G major chord (D3, G3, B3) is tuned to the sound of a humming guitar amplifier. Notice how the tuning of the chord (blue 'x') jumps up and down half a semitone as is gets pushed around by frequencies in the environmental noise (black '-') until a relatively stable frequency is close enough to attract and thereby anchor the tuning of the chord.

6 Conclusion and outlook

The main goal of the present project was to demonstrate that dynamically tuning music to environmental noise is in principal possible and to explore some of the available tools. For that purpose, the open source software ATMEN was devised, described, and evaluated here. We found that the beating theory of dissonance is both clear and flexible enough as a theoretical framework for our task. A measure of dissonance in this vein was forged to our needs. In particular, we were able to correct its gradient in a way that prevents uniform pitch drift. When evaluating the system, we inspected some of its strengths and weaknesses. Confirming the pioneer work of Sethares (1993), we could show that there is a great potential for adaptive tuning guided by roughness-reduction, as a more general and more flexible approach than knowledge based systems both for adaptive tuning of musical instruments in isolation and in the context of environmental noise.

Yet, ATMEN is far from a mature system. Even though we have seen very promising results, they are not reliable enough for application in non-experimental contexts. There are many parameters that require a more systematic exploration and evaluation. In particular, we have seen that it is wise to carefully choose and adjust a dissonance function. Furthermore, it could be worthwhile to implement a custom optimization algorithm that utilizes the information we have about our dissonance measure, in particular the approximate size and shape of local optima, more effectively. In our evaluation we noted that ATMEN currently does not produce tunings that are necessarily "horizontally" consistent and tends to overreact to chords that contain very dissonant intervals, even if they are only a result of passing tones or two chords fading into each other. Maybe it is possible to solve these problems directly, similar to the

problem of uniform pitch drift, but if not, measures need to be taken that both make the tuning more reliable and still leave it flexible enough to effectively adapt to fixed frequencies. Furthermore, we attempted to reduce the inner-musical dissonance as well as the dissonance between music and noise with a single method. Maybe it is better to implement the latter more as a meta-tuning step, like finding the optimal transposition of the music as a starting point for the fine-tuning described here.

Apart from these practical challenges, it is not clear whether the aims of tuning to environmental noise described in Section 1 are feasible even in theory: While there is conclusive evidence that environmental noise can compromise performance and even poses a number of serious health risks (Basner et al., 2014) and we have seen some evidence in Section 1 that music can have the opposite effect, whether *adding* music to noise can alleviate its negative effects has not been studied. Nor has it been studied whether dissonance between environmental sounds and music is even perceived as dissonant. After all, humans are generally good at processing different sound sources separately – it is entirely possible that the dissonance between the music and the environmental noise is less of a problem than the slight inconsistencies we introduce to the music by tuning it to the noise.

References

Allen, K. and J. Blascovich (1994). Effects of music on cardiovascular reactivity among surgeons. *JAMA 272*(11), 882–884.

Basner, M., W. Babisch, A. Davis, M. Brink, C. Clark, S. Janssen, and S. Stansfeld (2014). Auditory and non-auditory effects of noise on health. *Lancet 383*(9925), 1325–1332.

Bernini, A. and F. Talamucci (2014). Consonance of complex tones with harmonics of different intensity. *Open Journal of Acoustics 4*(2), 78–89.

Bibby, N. (2009). Tuning and temperament: closing the spiral. In J. Fauvel, R. Flood, and R. Wilson (Eds.), *Music and Mathematics*, pp. 12–27. Oxford: University Press.

Bigand, E., R. Parncutt, and F. Lerdahl (1996). Perception of musical tension in short chord sequences: The influence of harmonic function, sensory dissonance, horizontal motion, and musical training. *Perception and Psychophysics 58*(1), 125–141.

Bonin, T. and D. Smilek (2015). Inharmonic music elicits more negative affect and interferes more with a concurrent cognitive task than does harmonic music. *Attention, Perception, & Psychophysics 78*(3), 946–959.

Chanda, M. L. and D. J. Levitin (2013). The neurochemistry of music. *Trends in Cognitive Sciences 17*(4), 179–193.

Dalton, B. H. and D. Behm (2007). Effects of noise and music on human and task performance: A systematic review. *Occupational Ergonomics 7*(3), 143–152.

Dillon, G. (2013). Calculating the dissonance of a chord according to helmholtz theory. *European Physical Journal Plus 128*(8).

Dolegui, A. S. (2013). The impact of listening to music on cognitive performance. *Inquiries Journal/Student Pulse 5*(9).

Ellermeier, W. and K. Zimmer (2014). The psychoacoustics of the irrelevant sound effect. *Acoustical Science and Technology 35*(1), 10–16.

Fastl, H. and E. Zwicker (2007). *Psychoacoustics: Facts and Models*. Berlin: Springer.

Ferguson, Y. and K. Sheldon (2013). Trying to be happier really can work: Two experimental studies. *The Journal of Positive Psychology 8*(1), 23–33.

Fletcher, H. (1940). Auditory patterns. *Reviews of Modern Physics 12*(47), 47–65.

Fox, J. G. and E. D. Embrey (1972). Music – an aid to productivity. *Applied Ergonomics 3*(4), 202–205.

Hall, J. C. (1952). The effect of background music on the reading comprehension of 278 eighth and ninth grade students. *Journal of Educational Research 45*(6), 451–458.

Haynes, B. (2002). *A History of Performing Pitch: The Story of A*. Lanham: Scarecrow Press.

Hutchinson, W. R. and L. Knopoff (1978). The acoustic component of western consonance. *Interface 7*(1), 1–29.

Komeilipoor, N., M. W. M. Rodger, C. M. Craig, and P. Cesari (2015). (dis-)harmony in movement: effects of musical dissonance on movement timing and form. *Experimental Brain Research 233*(5), 1585–1595.

Labbé, E., N. Schmidt, J. Babin, and M. Pharr (2008). Coping with stress: The effectiveness of different types of music. *Applied Psychophysiology and Biofeedback 32*(3), 163–168.

Lee, O. K. A., Y. F. L. Chung, M. F. Chan, and W. M. Chan (2005). Music and its effect on the physiological responses and anxiety levels of patients receiving mechanical ventilation: a pilot study. *Journal of Clinical Nursing 14*(5), 609–620.

Lesiuk, T. (2005). The effect of music listening on work performance. *Psychology of Music 33*(2), 173–191.

Loy, G. (2011). *Musimathics: The Mathematical Foundations of Music Volume 1*. Cambridge: MIT Press.

Masataka, N. and L. Perlovsky (2013). Cognitive interference can be mitigated by consonant music and facilitated by dissonant music. *Scientific Reports 3*, 2028.

Parncutt, R. (1989). *Harmony: A Psychoacoustic Approach*. Berlin: Springer.

Parncutt, R. and G. Hair (2011). Consonance and dissonance in music theory and psychology: Disentangling dissonant dichotomies. *Journal of Interdisciplinary Music Studies 5*(2), 119–166.

Pavlović, I. and S. Marković (2011). The effect of music background on the emotional appraisal of film sequences. *Psihologija 44*(1), 71–91.

Pease, T. and B. Freeman (1989). *Arranging 2, Workbook*. Boston: Berklee college of music.

Plomp, R. R. and W. J. M. Levelt (1965). Tonal consonance and critical bandwidth. *The Journal of the Acoustical Society of America 38*, 548–560.

Ravaja, N. and K. Kallinen (2004). Emotional effects of startling background music during reading news reports: The moderating influence of dispositional bis and bas sensitivities. *Scandinavian Journal of Psychology 45*(3), 231–238.

Roederer, J. G. (1975). *Introduction to the physics and psychophysics of music*. Heidelberg: Springer.

Scharf, B. (1970). Critical bands. In J. V. Thobias (Ed.), *Foundations of modern auditory theory*, pp. 159–202. New York: Academic Press.

Schlittmeier, S. J., J. Hellbrück, and M. Klatte (2008). Does irrelevant music cause an irrelevant sound effect for auditory items? *European Journal of Cognitive Psychology 20*(2), 252–271.

Sethares, W. (1993). Local consonance and the relationship between timbre and scale. *The Journal of the Acoustical Society of America 94*, 1218–1228.

Sethares, W. (1994). Adaptive tunings for musical scales. *The Journal of the Acoustical Society of America 96*, 10–18.

Sethares, W. (2002). Real-time adaptive tunings using max. *Journal of New Music Research 31*(4), 347–355.

Sethares, W. (2005). *Tuning, Timbre, Spectrum, Scale*. London: Springer.

Sethares, W. and K. Hobby (2016). Inharmonic strings and the hyperpiano. *Applied Acoustics 114*, 317–327.

Sethares, W., K. Hobby, and Z. Zhang (2017). Using inharmonic strings in musical instruments. In O. A. Agustín-Aquino, E. Lluis-Puebla, and M. Montiel (Eds.), *Mathematics and Computation in Music. MCM 2017. Lecture Notes in Computer Science*, pp. 104–116. New York: Springer.

Shih, Y.-N., R.-H. Huang, and H.-S. Chiang (2009). Correlation between work concentration level and background music: A pilot study. *Journal of Prevention, Assessment & Rehabilitation 33*(3), 329–333.

Stange, K., C. Wick, and H. Hinrichsen (2017). Playing music in just intonation - a dynamically adapting tuning scheme. *Computer Music Journal 42*(3), 1–22.

Suzuki, Y. and H. Takeshima (2004). Equal-loudness-level contours for pure tones. *The Journal of the Acoustical Society of America 116*, 918–933.

Terhardt (1979). Calculating virtual pitch. *Hearing Research 116*(2), 155–182.

Terhardt (2015). Cross-cultural perspectives on music and musicality. *Philosophical transactions of the Royal Society of London 370*(1664).

von Helmholtz, H. ([1863] 1968). *Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik. (German) [On the Sensations of Tone as a Physiological Basis for the Theory of Music]*. Hildesheim: Olms.

Ward, W. D. (1970). Musical perception. In J. V. Thobias (Ed.), *Foundations of modern auditory theory*, pp. 407–447. New York: Academic Press.

Zwicker, K. E. (1961). Subdivision of the audible frequency range into critical bands. *The Journal of the Acoustical Society of America 33*(2), 248–249.

Zwicker, K. E. and E. Terhardt (1980). Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of the Acoustical Society of America 68*(2), 1523–1525.

Črnčec, R., S. J. Wilson, and M. Prior (2006). The cognitive and academic benefits of music to children: Facts and fiction. *Educational Psychology 26*(4), 579–594.