SONICLEVEL-TUNING APP FOR THE ICAD 2023 SONIC TILT COMPETITION

Stephen Barrass

Sonic Information Design
sonification.com
Canberra, Australia
stephen.barrass@sonification.com

ABSTRACT

The SonicLevel-Tuning App transforms the 2D visual task of using a spirit level into a one dimensional sonic task modelled on tuning a guitar string by ear.

1. LINK TO APK FILE

The SonicLevel-Tuning App builds on the Tiltification App [1] using code from the open source project Sonic Tilt.

The SonicLevel-Tuning App can be downloaded onto your mobile phone from `https://drive.google.com/file/d/1iNDpwW4pGZ1CBXfkqRD7wzgMwdTs6LHt/view?usp=share_link`.

2. INTRODUCTION

While thinking about the Sonic Information Design for the ICAD 2023 Sonic Tilt Competition I picked up my guitar and idly tuned the strings. "What sort of sound could tell you whether to go up or down" I wondered as I adjusted the knob to raise the pitch to bring the bottom string more in tune. "How could sound be used to tell whether you are close to a particular position" I mused as I listened for the beating sound that happens when the strings resonate at almost the same frequency. "Aha thats it" I thought as I fine tuned it so the beating gradually slowed to a single tone which happens when the string is perfectly in tune. "If only I could come up with a way to make a sonification that was as easy and precise as tuning a guitar string!" I thought to myself.

Tuning a musical instrument is a very common activity. You can get better at it with practice but nearly anyone can learn how to do it quite quickly. In the past musicians used a tuning fork to generate an exact reference tone, but today electronic tuners and mobile phone Apps are more common. However, the tuning process is the same - you manually adjust a knob up and down to change the pitch to make it closer to the reference, until you begin to hear interference beats that slow down as you approach the exact frequency. Most people are able to tune a string to the reference tone by listening. Learning to tune an instrument is good fun, especially once you get good at it, and an instrument that is in tune is much more pleasant to listen to than one that is not.

Figure 1: *Tuning an Acoustic Guitar*

3. YOUR SONIFICATION

The spirit level helps you to answer various questions that occur in different stages of the task.
1) How far is it from level at the moment? (close, medium, far).
2) Which way do I need to tilt it to make it more level? (left, right, up, down)
3) Is it level yet? (Yes/No).

My first attempt used 3 tones - a reference tone, an X axis tone and a Y axis tone. The angular distance from zero for each axis was mapped to a difference from the reference frequency for each axis tone. A negative angle caused the frequency to go lower than the reference and a positive angle caused it to go higher. When the axis value is zero the tone is in tune with the reference. I tested it out by closing my eyes and trying to answer the questions. However I couldn't easily answer which way to tilt it because I didn't know which tone was the X axis and which was the Y. It was like trying to tune two strings at once which is something that even experienced musicians don't try to do.

This lead me to try to reduce the task to tuning just one string by calculating a 1 dimensional radial distance from level from the two X,Y coordinates. This time I could tell that it was far from level because there were two tones with different pitches, and I could tell that I was tilting it in the right direction because the pitches got closer together, just like tuning a string. I could tell it was a mid distance away by a rough timbre effect. I could tell that it was nearly level when I started to hear a beating effect that slows smoothly from 5 Hz to 4 Hz to 3 Hz and then to a pure tone when it is perfectly level.

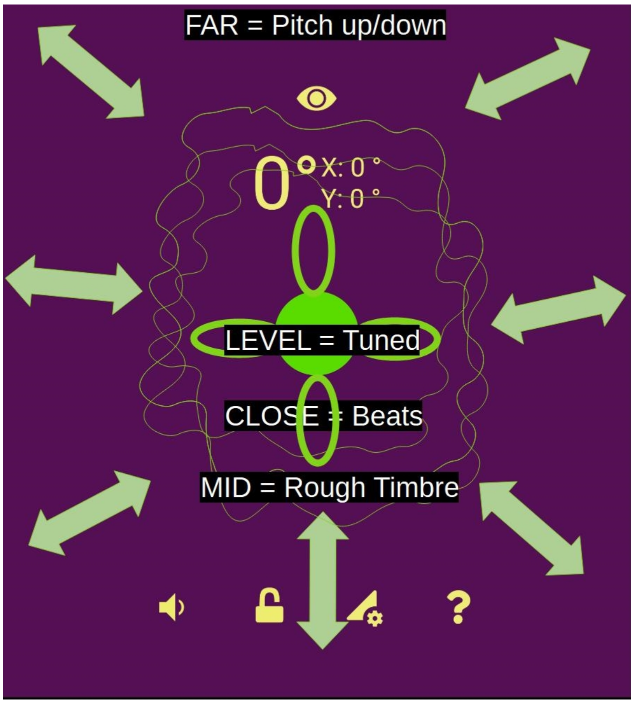

Figure 2: *Sonic Information Design.*

4. IMPLEMENTATION

The implementation in PD consists of two oscillators that produce simple sine tones. The reference oscillator produces a tone at 666 Hz. The tuning oscillator is offset from the reference frequency by the radius calculated from the X and Y tilt angles of the accelerometer. That's pretty much all there is to it !

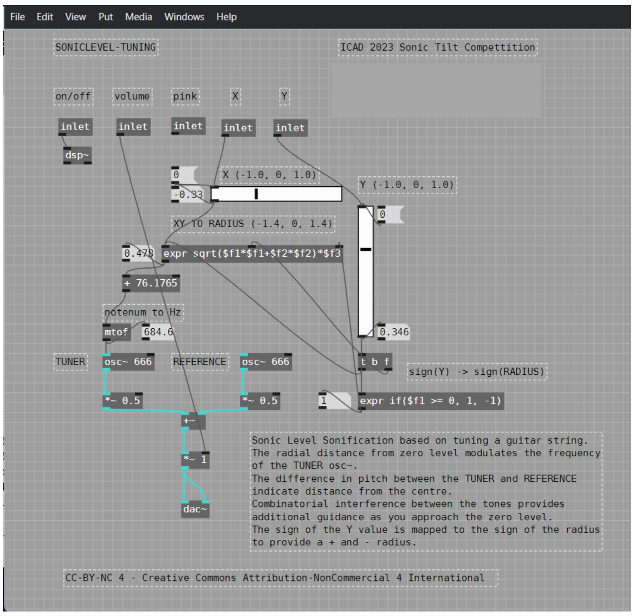

Figure 3: *PD Patch*

The changes in the radius are mapped to midi notenumbers which are linear changes in pitch. The *mtof* unit is then used to transform from *notenumber* to frequency. This means that equal changes in the data are mapped to equal changes in pitch perception.

Although the technical design is extremely minimal the two simple oscillators produce psycho-acoustic and acoustic effects that provide quite different kinds of information with different levels of resolution throughout the interactive tuning process. When the tones are more than 100 Hz apart you hear two distinct pitches which you can compare as higher or lower than each other. When the tones are at a medium distance (20 to 100 Hz) apart you hear a rough timbral effect due to overlapping of the frequencies in the same critical band in the cochlear. When the tones are less than 20 Hz apart you hear a beating amplitude modulation at a rate equal to the difference between the frequencies which is caused by acoustic interference before the sound enters the ear.

A video of the SonicLevel-Tuning App can be found at `https://youtube.com/shorts/FzZ403d_cd0`.

5. EVALUATION

I did a rudimentary evaluation by timing how long it took to level the phone by hand in 10 trials using the Visual, Sonic, and Visual+Sonic modes.

	Visual	Sonic	Visual+Sonic
Mean	16.4	23.8	14.8
StdErr	1.6	3.02	1.13

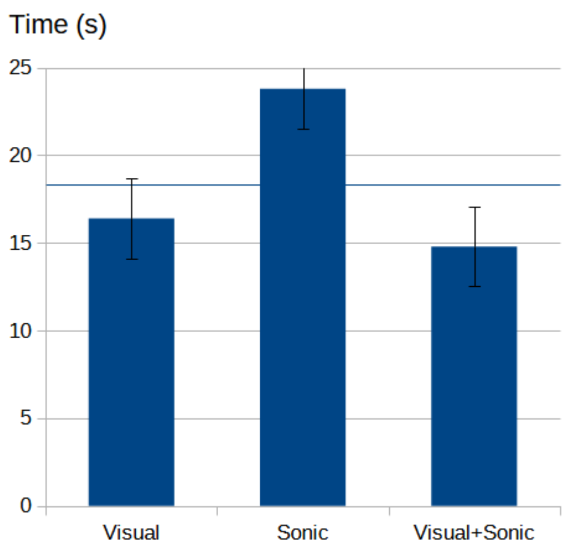

Figure 4: *Time to Level over 10 trials in Visual, Sonic and Visual+Sonic modes*

These results from a small number of trials by me alone cannot be taken too seriously, but the process helped me to pay attention to, and reflect on, the relationship between the task and the sounds I heard. My performance in the Sonic mode was 50% slower than the Visual mode (23 s vs 15 s), while the combined mode was 10%

faster (1.4 s) than the Visual mode. I found it easy to get close to level in the Sonic mode, but it took longer to find the exact centre because although the sound provides information about closeness I couldn't tell which direction to tilt. This meant I had to keep trying random movements to find the exact level which was frustrating. The sonification design could be improved by adding some kind of directional information when close to level, but I like the simplicity of the tuning metaphor and the implementation. Although I found the SonicLevel-Tuning App to be fun and rewarding, people around me thought it was annoying.

6. ACKNOWLEDGMENT

Big thanks to Tim Ziemer for all the time and effort required to organise the Sonic Tilt Competition, and to his students for building the App which provides a learning tool and software framework for sonification researchers who would like to make Killer sonification Apps for the masses.

7. REFERENCES

[1] M. Asendorf, M. Kienzle, R. Ringe, F. Ahmadi, D. Bhowmik, J. Chen, K. Huynh, S. Kleinert, J. Kruesilp, Y. Lee, X. Wang, W. Luo, N. Jadid, A. Awadin, V. Raval, E. Schade, H. Jaman, K. Sharma, C. Weber, H. Winkler, and T. Ziemer, "Tiltification — an accessible app to popularize sonification," in Proc. 26th International Conference on Auditory Display (ICAD2021), Virtual Conference, June 2021, pp. 184–191. [Online]. Available: https://doi.org/10.21785/icad2021.025