

Live Performance Leveler V1.0

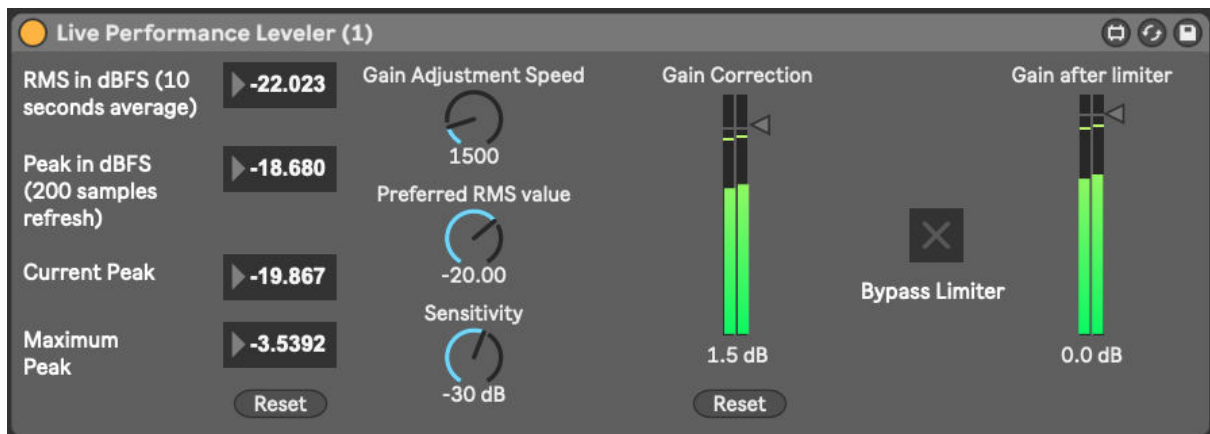


Table of contents

Table of contents	1
Introduction	2
The product	3
How it works	3
Building procedure in Max	5
Live Performance Referencer	5
Live Performance Leveler	6
Testing procedure	9
Future goals & applications	11
Conclusion	12
Reflection	13
Appendix 1: Instructions on installing and running the software	14
Appendix 2: Glossary	15
Appendix 3: References	16

Introduction

The club [Sameheads](#) is a small club (capacity 350 people) in Neukölln focussing on left-field and experimental music. Apart from DJs they also regularly host live performances and concerts at their club nights. In September 2019 they started their own weekly radio: [Heads Radio](#), which also features live performances.

I am the technician at Heads Radio and have noticed that the loudness of some live performances is varying a lot, unintentionally from the artist. Currently I am doing manual gain changes, which means watching/hearing the levels and changing the gain when necessary, to keep the loudness level consistent. I also use a compressor and limiter to further improve the consistency of the loudness and to prevent the audio signal from clipping. However, I don't want to overuse the compressor to keep the volume level consistent because that sacrifices the dynamic range (the difference between the quietest and loudest part in an audio signal). In a live performance it is important to keep the dynamics intact to a certain degree, with too less dynamic range a performance sounds flat and without movement. This is a tedious task and hard to do accurately in real time, which is needed with a live performance.

I think there is a way to automate this process with software that changes the gain automatically, based on the output level of the live performance. The same technique can also be used to compress and equalize (EQ) the audio signal and manipulate the stereo-imaging later on.

If done correctly, this will improve the sound of live performances because a program can react to changes instantly and without prejudice.

Existing mixing/mastering software

There is existing automatic gain level software and hardware available but they don't focus on the needs of live performances. For example there is an automatic gain control (AGC) in radio receivers which changes the gain very rapidly to keep it consistent (Kester 2005). However, this sacrifices the dynamic range which is not ideal for a live performance.

Currently there are also companies doing automated mixing and mastering driven by machine learning (e.g. [LANDR](#), [eMastered](#), [BandLab](#), [CloudBounce](#)) which do a good job at mixing and mastering recorded audio (Sterne et al. 2019). However, this isn't a solution for live performances and live radio since the processing needs to be in real time, not afterwards on the recording.

Live Performance Referencer & Leveler

I have two programs that work together to analyze the average volume of a live performance and automatically change the volume of a live performance to a preferred loudness level in real time. The analyzing part is done by the Live Performance Referencer and the volume change by the Live Performance Leveler.

In this documentation I will describe what the product is, how it works, how I built it, how I tested it and what the future goals and applications of this product are. Lastly, I will give my final conclusion and reflection on the product and the process.

Some terms I use might not be familiar to everyone. You can find the explanations of these terms in the [Appendix 2: Glossary](#).

The product

The Live Performance Referencer and Leveler is software created in [Max](#). I run Max on the digital audio workstation (DAW) [Ableton Live](#). The output of Ableton Live will go into [Open Broadcast Software \(OBS\)](#), which is the software used at Heads Radio to live stream. To get the output from Ableton Live into the input of OBS I routed it via the program [Soundflower](#).

Max is a visual programming language for music and multimedia developed by [Cycling '74](#). Max is a modular program with shared libraries, an application programming interface (API) and graphical user interface. This makes it really user friendly and easy to convert ideas. Max has been described as the lingua franca for developing interactive music software (Place et al. 2006).

Because of this I have chosen Max to build the Live Performance Leveler in. I've chosen Ableton as my DAW because Max runs in Ableton as Max for Live, and not in other DAW's. Both of the programs I had used before, which also influenced my decision. I need Max to run in Ableton because I can only route the output of Ableton to go into OBS, not from Max directly.

The software consists of two programs. The Live Performance Referencer and the Live Performance Leveler. Information on how to install and run software can be found in [Appendix 1](#).

How it works

The Live Performance Referencer will measure the average dBFS of one minute from a reference live recording from the same genre and style which have the preferred dynamics. This can be chosen by the artists and/or sound engineer.

With the measured value by the Referencer, the sound engineer can set the *Preferred RMS value* in the Live Performance Leveler. If the RMS value of the live performance goes out of the preferred range, the Live Performance Leveler will automatically apply gain correction until it's back at the preferred value. For example, if the preferred value is -20 dBFS RMS, the live performance leveler will apply gain amplification as soon as the RMS value goes below -20 dBFS RMS and vice versa. In other words, when the live performance is too quiet, the software will make it louder until it's at the value set by the reference recording. When the live performance is too loud it will do the opposite, make it quieter until it's at the value measured by the referencer.

This technique brings with it a few difficulties. If the artist(s) intentionally did a quieter or louder passage the software will still correct it to match the value set by the reference. The software also needs to distinguish short periods of loudness change (micro-dynamics) and long periods of loudness change (macro-dynamics) and react to both changes differently. If not, the signal will instantly be boosted to match the range of the reference track, even if the signal goes below the threshold for a short period of time (e.g. the time in between kick drums). This will destroy the micro-dynamics within a performance and result in a flat sounding mix.

The micro-dynamics should be corrected with compression or limiting and the macro-dynamics with gain amplification or reduction.

To handle this, the Live Performance Leveler will react to changes with the following rules in the following order:

1. If the signal goes out of the RMS value set by the reference, it will correct with gradually increasing gain amplification or reduction.
2. If the signal peaks above -1 dBFS Peak it will be compressed by a limiter to prevent clipping.

Metering

The Peak dBFS value will be measured every 1 sample at a sample rate of 44100.

The RMS dBFS value will be the average of 10 seconds (441000 samples).

Gain correction

In order to make the gain amplification and reduction unnoticeable it needs to be done gradually over time. The more and the longer the signal goes out of range, the more amplification/reduction will be applied. The correction will take place until the signal is back at the value set as *Preferred RMS value*

The ratio the gradual change will take place can be changed in the software to the preference of the artists/sound engineer. The standard setting will be $(\text{RMS dBFS} - \text{Preferred RMS value})/1500$, which I have tested to be best on most live performances (see [Testing Procedure](#)). The value is changeable with a dial. The higher the value after the /, the slower the gain correction will take place and vice versa.

The sensitivity to which signals the software reacts with gain correction can be changed to the preference of the sound engineer. This is important at intended quieter passages done by the artist, e.g. a slow fade in at the beginning of their performance.

The gain correction will be reacting to the measured RMS value because the RMS value will come the closest to the perceived loudness of the human ear and thus is the most accurate value to use to correct volume changes.

Limiting

If the signal peaks above -0,96 dBFS it will be compressed by the limiter, a very fast compressor, with a very high compression ratio (e.g. 1000:1 to inf:1). This will prevent the audio signal from clipping.

Building procedure in Max

Here I will describe how I have built the Live Performance Referencer and the Live Performance Leveler. Because Max is a visual programming language I decided to also present this visually, with explanations in the patches itself.

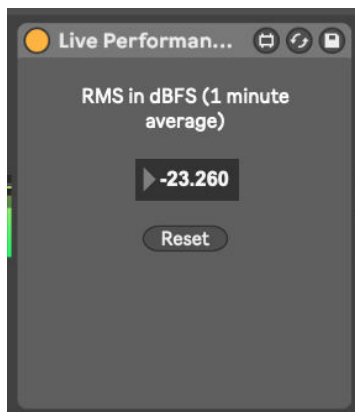
The Presentation Mode is how the software/device will look like when opened in Ableton Live. This is the user interface and will provide all the visible feedback and controls to control the Leveler and Referencer during a Live Performance.

The Patch Mode provides the overview on how the software/device functions. Max works with various objects with different functions which are connected to each other with patch cables. I have explained the different patches in the images below.

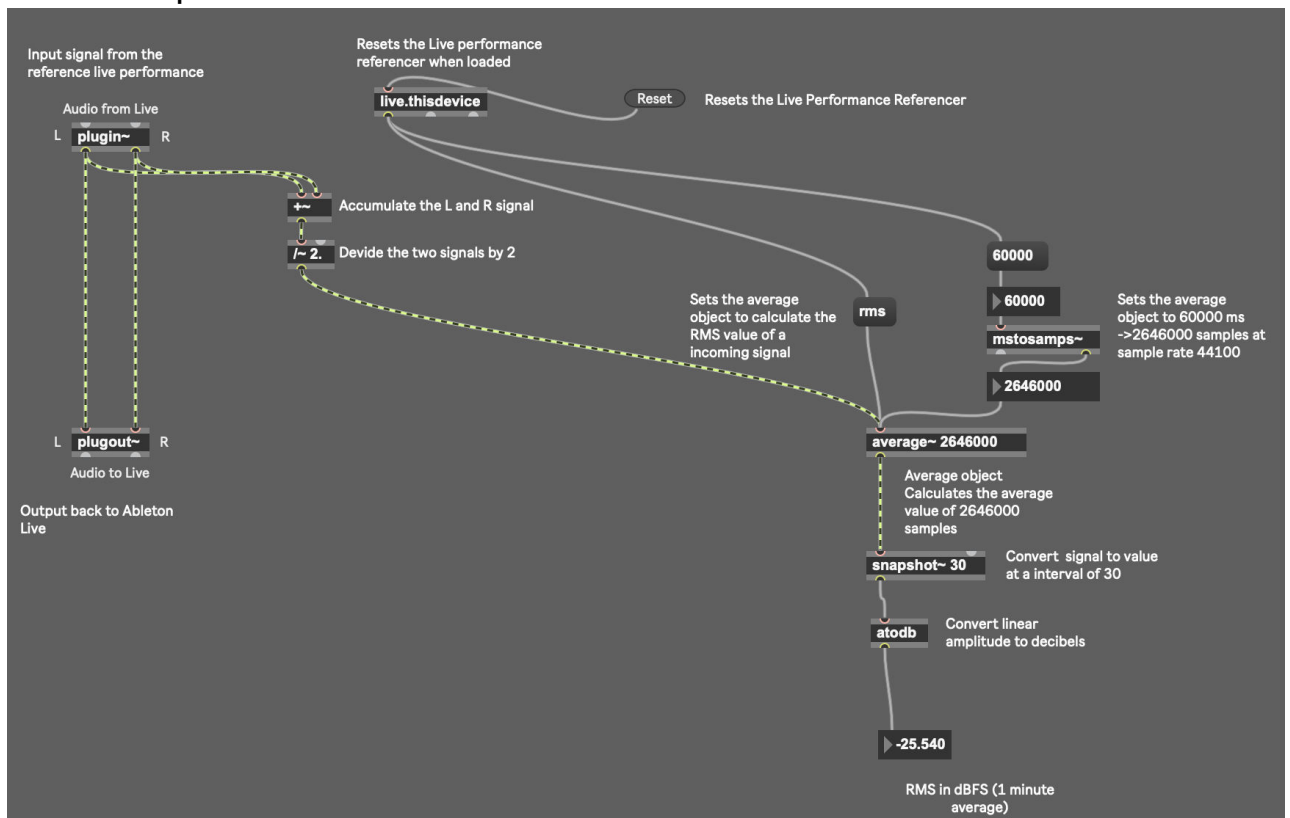
You can check out my video on how to install and run the software [here](#).

Live Performance Referencer

Presentation Mode:

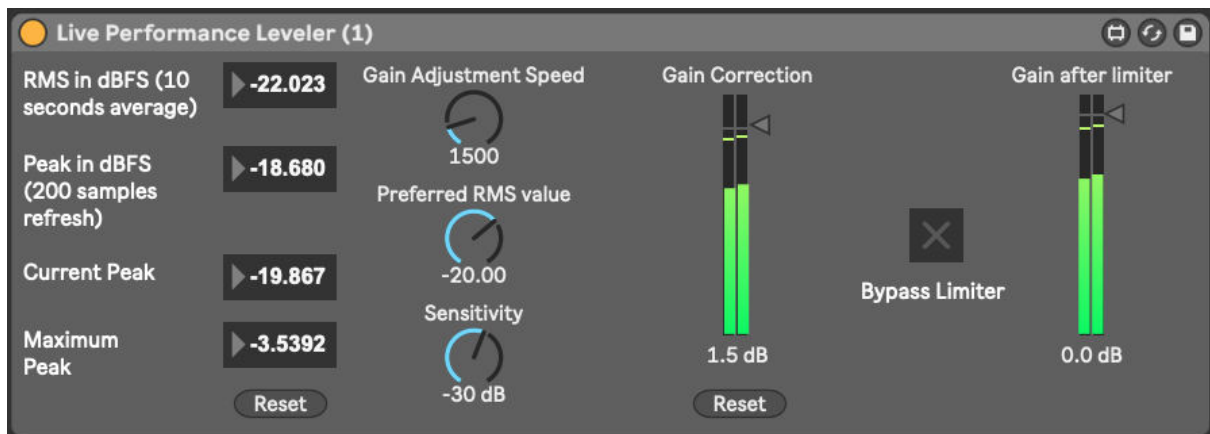


Patch mode explanation:

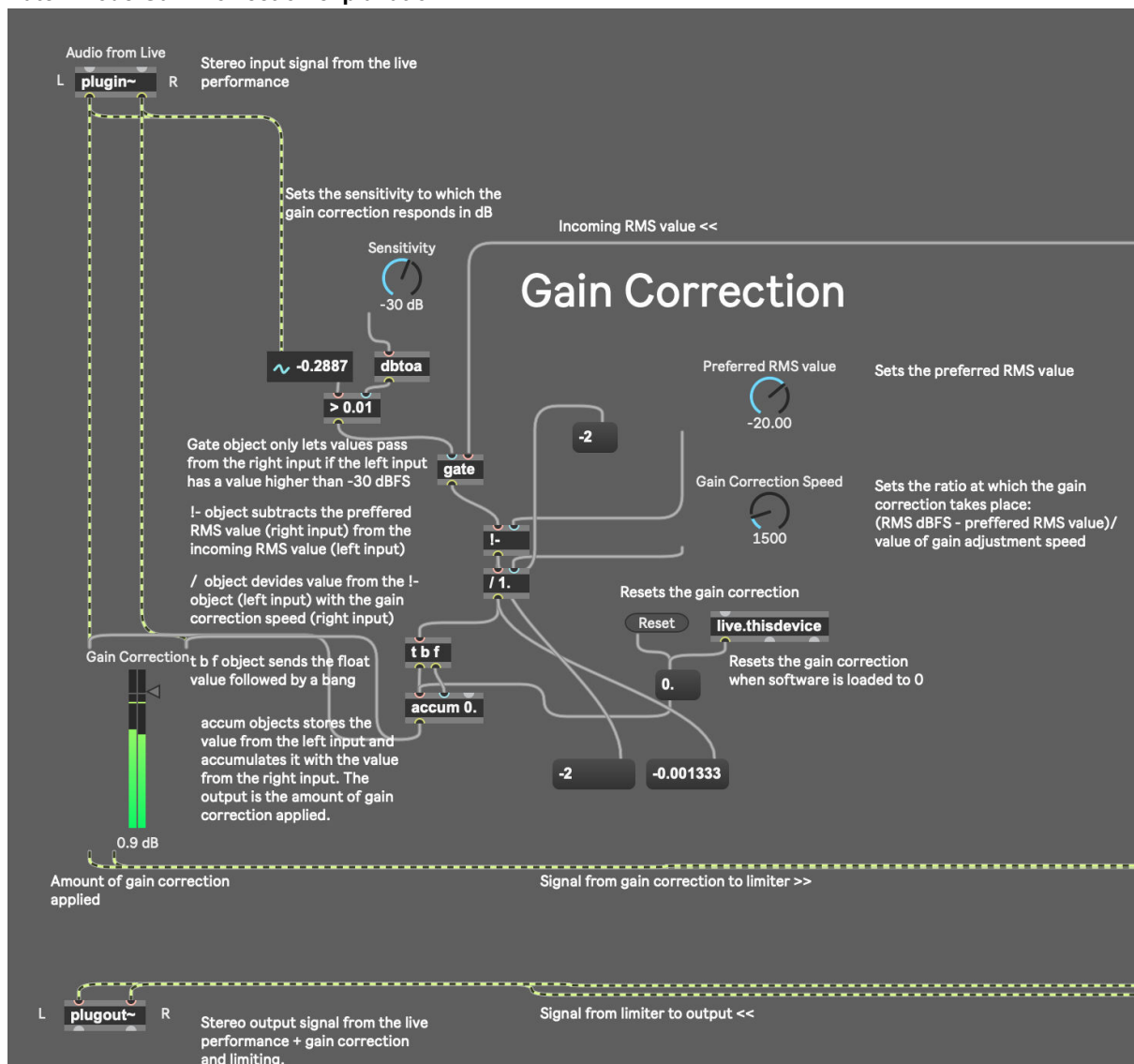


Live Performance Leveler

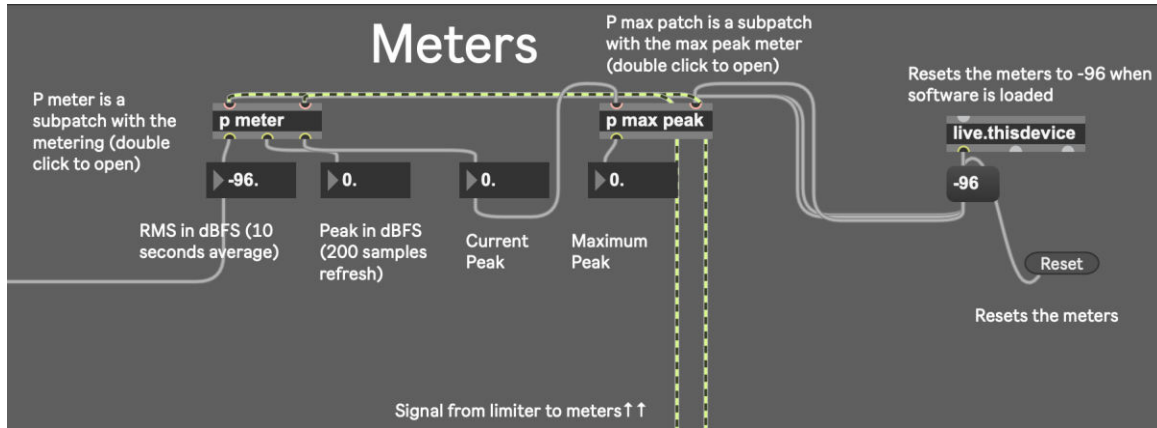
Presentation mode:



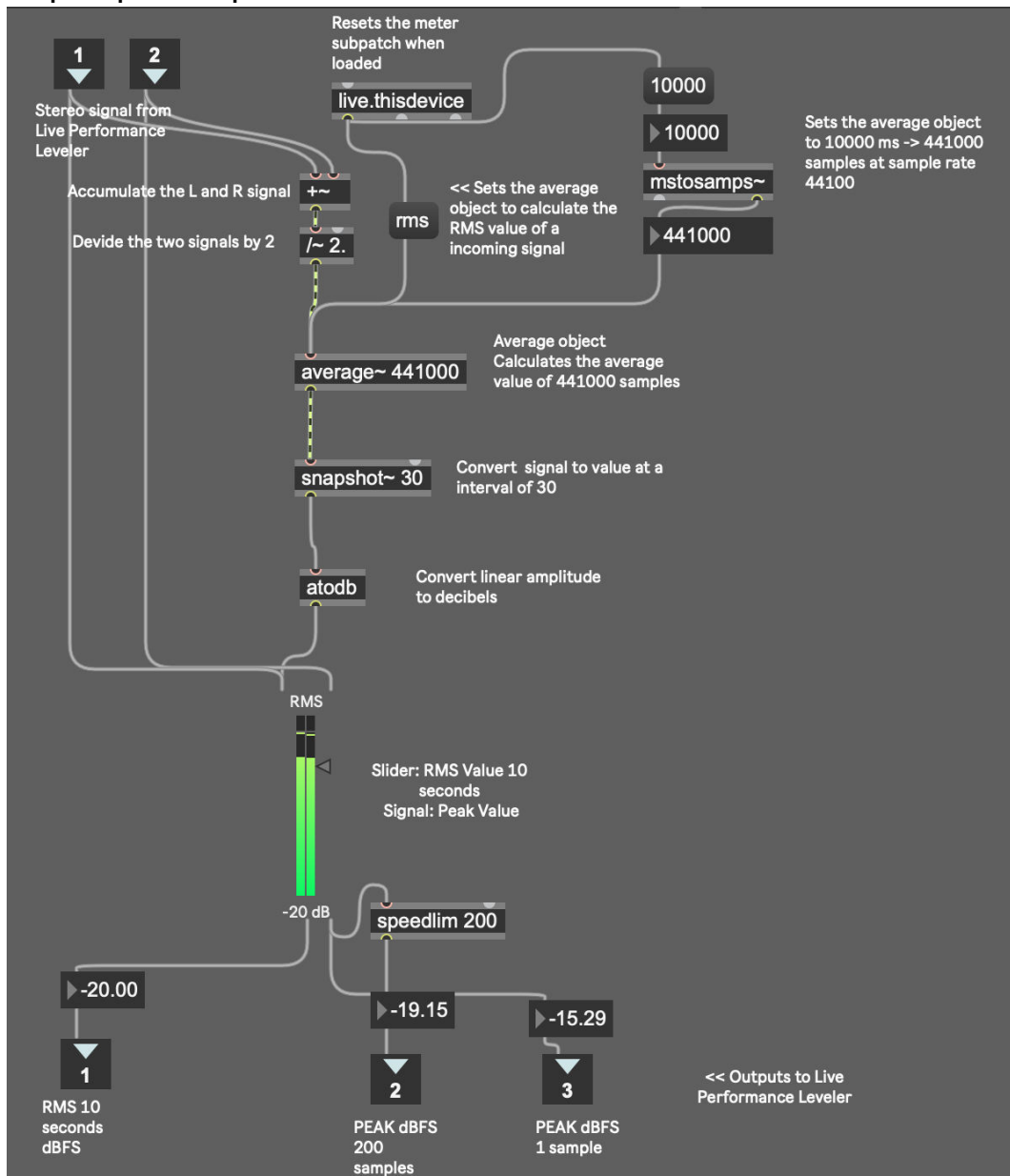
Patch mode Gain Correction explanation:



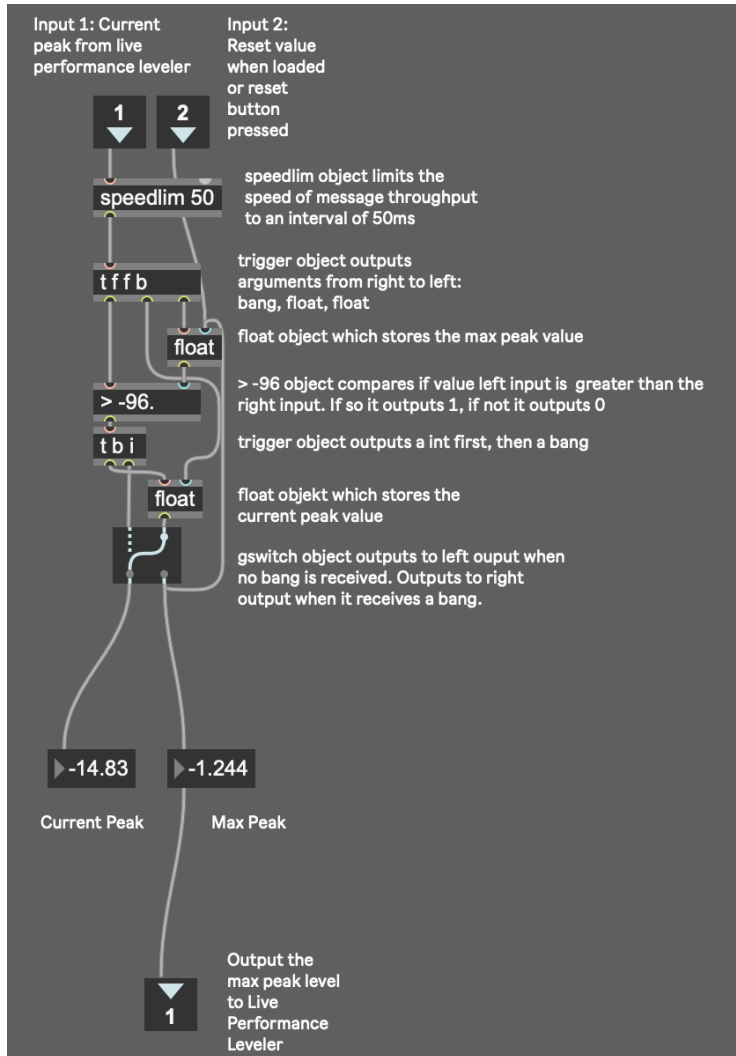
Patch mode Meters explanation:



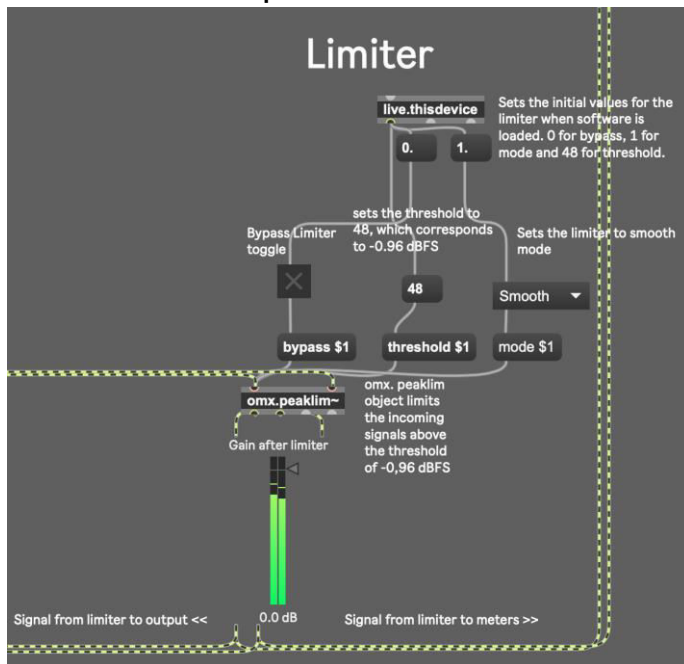
Subpatch p meter explanation:



Subpatch p max peak explanation:



Patch mode Limiter explanation:



Testing procedure

After I made the Live Performance Referencer and Leveler I started with a testing procedure to test if the software was working the way I wanted it to.

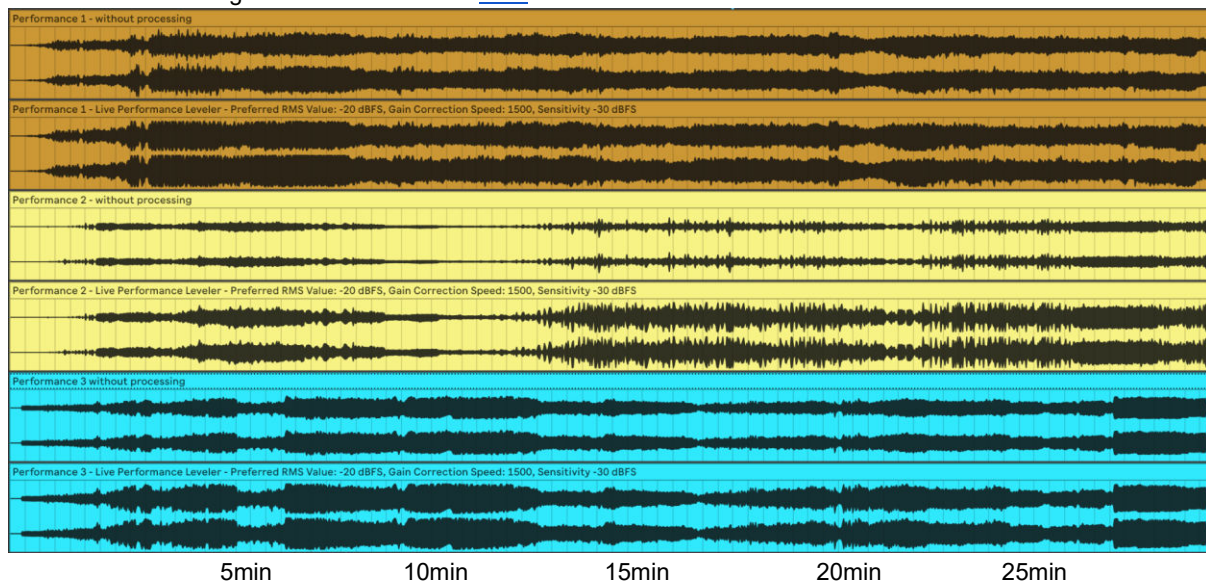
The first thing I wanted to test was the *Preferred RMS value* to put into the Live Performance Leveler. I used the Live performance Referencer on three recordings of a live performance at Heads Radio which had a consistent loudness. The RMS value (1 minute average) was around 18-22 dBFS throughout most of the performances. This is why I chose the value of -20 dBFS RMS as default value for the Live Performance Leveler.

To test the other two variables I used three recordings with varying dynamics and loudness. For the *Gain Correction Speed*, I started out at $(\text{RMS dBFS} - \text{preferred RMS value})/100$. At this speed the Leveler almost instantly corrected lower and higher RMS values to the *Preferred RMS value*. This caused a gain correction that was hearable. Up till a gain correction speed of 1000 this was the case. At 1000 the correction was still fast but not noticeable anymore. However, at this pace the recordings still had unnatural macro-dynamics. From 1000-1500 the correction progressed into being less noticeable. From 1500 upwards I noticed that the Leveler was too slow to respond to changing dynamics and with fast changing dynamics in the recordings this caused excessive clipping (when the recording went from very quiet to loud in a short period of time). This is why I chose 1500 as the default value for the *Gain Correction Speed*.

The third variable to test was *Sensitivity*. With the *Sensitivity* dial at -inf dB the Leveler would apply gain correction to very quiet signals, or even silence (e.g. the beginning and end of a performance). This would result in unnatural amplifications at the beginning and end of performance. With a sensitivity up to -40 dB this was the case at the three recordings I tested it on. Above -30 dB the Leveler would be too unreactive. This is why I chose -30 dB as the default value for *Sensitivity*.

The best threshold I found for the limiter to react was -0.96 dBFS Peak, with the mode on smooth. This gave a compression without hearable artifacts and kept the signal from clipping. With these parameters the limiter was only compressing high peaks every +/- 10min.

To test the default values of the Live Performance Leveler I used it on three 30 minute snapshots from live performances at Heads Radio (beginning of the recording till 30min). The unprocessed and processed recordings are shown in the image below and hearable [here](#).



As you can see from the waveforms, the three recordings have been amplified by the Live Performance Leveler. (bigger waveforms means an amplified signal). Performance 1 also shows some limiting done on the peaks of the signal from +/- 3 - 7 minutes (flattened peaks on the waveform). To me all these changes were improving the loudness of the performances without hearable artefacts.

Live performances differ in dynamics and loudness so the sound engineer needs to keep an eye on the Leveler and change the variables if necessary. For instance, if there is excessive clipping after the gain correction, the *Preferred RMS Value* can always be dialed back, if the gain correction is too slow or too fast the *Gain Correction Speed* can be changed and if the Leveler is unresponsive to signals the *Sensitivity* can be dialed to a lower threshold.

The default values proved best on the three live performances I tested it on. This doesn't mean that this will be the case for all live performances. However, it is a good starting point.

Future goals & applications

The first version of the Live Performance Referencer and Leveler are made and tested for Sameheads and its radio; Heads Radio. However, with proper testing, it can be used by other organizations, who host live performances, as well. Think about concerts, other radios or festivals.

Right now, the Live Performance Referencer and Leveler are not connected to each other, so the sound engineer has to manually adjust the *Preferred RMS Value* dial on the Leveler to the value measured with the Referencer. This connection will be made in the next update.

Due to limited time the first version focuses on gain correction and limiting, but this will extend to compression in the second version. Later on I will also implement an automatic equalizer and stereo-field manipulator. This would improve the sound of live performances even more.

There are also possibilities for using the software on individual stems of a live performance, to level out the different instruments in a band automatically. This will be more difficult to build, but with enough time, certainly possible.

Ultimately, to make the Live Performance Leveler function best it needs to become self-learning, like other mixing/mastering services. Because real-time audio processing has a lot more uncertain variables to it in comparison to recorded audio this will be a challenge, but certainly one I would love to take on. This also means that the Live Performance Leveler has to have a big database with reference performances and obviously has to be connected to the internet to allow cloud computing.

The Live Performance Leveler is made in Max for live, which is a program that runs in the DAW Ableton LIVE. Because of this the software is only accessible to people who have a licence for both of these programs. For now, this was a good option to make the software. To increase the accessibility of the product it might be useful to make the Live Performance Leveler function as standalone software in the future.

Conclusion

The Live Performance Referencer and Leveler are tools that can help Sameheads and Heads Radio improve the sound of their live performances. Due to limited time I was only able to implement gain correction and limiting in the product. This will expand to compression in the second version and equalization later on.

The Leveler does gradual amplification or reduction of the gain. Because of this the macro-dynamics of a live performance are changed slightly, due to the amplification or reduction not being even on the whole performance. This is an effect that can benefit the performance, but can also be a disadvantage on some performances. This is a fact that needs to be considered when using the Leveler and needs to be thoroughly tested.

The testing I did was done on three recordings from live performances at Heads Radio. In these tests the Live Performance Leveler functioned well and did an overall improvement of the loudness, without noticeable changes in macro-dynamics. Although I think these performances represented the dynamics of most of the performances at Sameheads and Heads Radio, I cannot say for sure. Further testing needs to be done to prove the functionality and value of the product.

The final test was supposed to be done at a Live performance at Sameheads. Due to limited time I wasn't able to complete this before the deadline. However, I will do this test in the coming week and update this documentation.

The owner of Sameheads has already told me that he would love to use the Live Performance Leveler at his club and radio, if the software works as I described. The final test next week will give more information regarding this.

Reflection

The last two weeks have been a challenge. I had so many ideas in my head about how live performances could be improved at Sameheads, and I didn't know where to start. After some days of gathering ideas I started out by making a solid plan of action, which also took into account the time constraint for completing this product. I chose to focus on the most important variable: Gain control.

I wrote down questions and variables I had to take into account for getting a Gain Control on real-time audio processing prototype up and running.

After this I had to choose how I wanted to make the product. I chose to make the product in Max for Live, because this was software I knew how to use and was also suitable for the product I had in mind.

When I started building in Max I noticed that I wanted too many variables to influence the behaviour of the Live Performance Leveler. I over complicated things with dynamically changing compressors and predicting future change in a live performance to influence the current gain and compression. After a while, I had to start over and simplify. That's when I came up with the gain corrections as it is now, with one value (the *Preferred RMS Value*) compared to the current RMS value to influence the gain.

This didn't make things any easier because the gain would change too rapidly (as I described in the chapter Testing Procedure). That's when I started to add the gain correction speed and sensitivity patches. After this I wanted to build in a compressor to catch peaks of the audio signal and compress it, to decrease the dynamic range of some live performances (who naturally have a very high dynamic range, which sounds unnatural sometimes). Unfortunately, all the compressors I tried to build in Max sounded bad. With 2 days left I decided to move on and build a limiter to catch only the highest peaks and keep the signal from clipping. Fortunately, I succeeded in this and could start with the testing procedure. Because of the limited time, I only had time to test the product on 3 live performances unfortunately.

What I have learned from this is that it's best to keep things as simple as possible in the setup of a product. Start with the most basic things and extend with more complicated things later on.

Most importantly, I did enjoy the process a lot and cannot wait to continue to improve the Live Performance Leveler.

Appendix 1: Instructions on installing and running the software

To run the software you need to install Ableton LIVE Suite (version 10 or higher). A trial version can be downloaded on their [website](#)

After installing Ableton LIVE Suite, drag the [Live Performance Referencer.amxd](#) and [Live Performance Leveler.amxd](#) into one of the tracks of Ableton LIVE.

Watch [this](#) video for further instructions.

Links:

[Dropbox folder](#)

Appendix 2: Glossary

ACG: Automatic gain control, is a closed-loop feedback regulating circuit in an amplifier or chain of amplifiers, the purpose of which is to maintain a suitable signal amplitude at its output, despite variation of the signal amplitude at the input

Clipping: Clipping is a form of waveform distortion that occurs when an amplifier is overdriven and attempts to deliver an output voltage or current beyond its maximum capability.

Compression: The process of reducing dynamic range in an audio signal.

(Compression) Ratio: The compression ratio determines how much gain reduction the compressor applies when the signal passes a threshold level. For example, a ratio of 4:1 means that for every 4 dB the signal rises above the threshold, the compressor will increase the output by 1 dB.

Decibels relative to full scale (dBFS): is a unit of measurement for amplitude levels in digital systems which have a defined maximum peak level.

Dynamic Range: The difference between the quietest and loudest volume part in a piece of music. Its measured with the Peak amplitude - Root Mean Squared amplitude (RMS)

DAW: Digital Audio Workstation. A DAW is an electronic device or application software used for recording, editing and producing audio files

EQ: Equalization. EQ is the process of adjusting the volume of certain frequency bands in an audio signal.

Gain: Gain is the ratio between the volume at the input and the volume at the output of an audio signal.

Gain Correction:

Limiting: A limiter does the same as a compressor but with a very high ratio, thus stopping signals to pass a certain threshold.

Micro - dynamics: short time dynamic changes in a piece of audio, e.g. how different elements in music performance relate to each other in a small amount of time (<1 minute).

Macro - dynamics: long time dynamics changes in a piece of audio, e.g. how different pieces in a whole music performance are related to each other in loudness.

Peak amplitude - The maximum amplitude of an audio signal

RMS amplitude - Root Mean Squared amplitude(average over a time period)

Stereo-Imaging: Stereo Imaging is the manipulation of an audio signal within a 180-degree stereo field, for the purpose of creating a perception of locality within that field.

Appendix 3: References

Kester, W. (2005) Communication Amplifiers. Op Amp Applications Handbook.

Place, T.; Lossius, T. (2006). A modular standard for structuring patches in Max. Jamoma. In Proc. of the International Computer Music Conference 2006.

Sterne, J. and Razlogova, E. (2019) Machine Learning in Context, or Learning from LANDR- Artificial Intelligence and the Platformization of Music Mastering. Social Media + Society. SAGE Journals.