Adaptive, Personalised “in browser” Audio Compression.

Andrew Mason
BBC Research and Development
Centre House, 56 Wood Lane
London, W12 7SB
+44 30304 09709
andrew.mason@bbc.co.uk

Matthew Paradis
BBC Research and Development
Centre House, 56 Wood Lane
London, W12 7SB
+44 30304 09889
matthew.paradis@bbc.co.uk

ABSTRACT

Audio quality is very important to the BBC’s audience, and unwanted loudness variations reduce the quality of the listeners’ experience. Dynamic range control applied by the broadcaster can go some way to avoiding problems but, because the broadcaster cannot take into account each listener’s individual environment, needs, or preference, it cannot please everyone all the time. The listening conditions are a significant factor to be taken into account when dynamic range control is applied. The Web Audio API offers the possibility of performing dynamic range control under the control of the listener, tailoring it optimally for their individual situation. We have developed a system that demonstrates that this is achievable in a modern web browser. In it, a compressor is controlled automatically by the environmental noise level measured using the microphone present in most mobile device audio players.

Categories and Subject Descriptors
H.5.1: [Multimedia Information Systems] Audio input/output

General Terms
Design, Experimentation, Human Factors.

Keywords
Object Based Audio, Web Audio API, Broadcasting.

1. INTRODUCTION

The way that broadcast audio reaches its audience is going through a period of significant change: even though traditional transmissions from "sticks on hills" still have a huge audience, more and more audio programming is being delivered through the internet or mobile telephony.

Although there are wide variations between individuals in the use of internet and mobile services [1] it is clear that mobile listening is increasing -- the number of people in the UK listening to the radio on a mobile phone or tablet has doubled in the last 4 years [2] and the mobile internet presents a huge opportunity for broadcasters.

When compared with traditional broadcast receivers (FM, DAB, digital terrestrial and digital satellite television) internet delivery can provide a much more flexible audio delivery path. Completely different decoders, or new applications to find and present programmes, can be downloaded onto a PC or mobile device, changing the capabilities of the device in seconds. Traditional receivers generally offer little in the way of upgradability, the changes usually being limited to superficial changes, rather than fundamental ones.

The internet has been used by the BBC for several experimental audio services, that would be impossible to try on conventional broadcast platforms. These include surround sound using binaural signals with selectable head-related transfer functions [3], live sports broadcast with the balance between commentator and stadium sound being adjustable by the listener [4], and virtual exploration of a sports venue with the audio scene adapting as the user "moves" around [5][10].

The bandwidth available to deliver services to fixed and mobile internet-connected devices is steadily increasing, through the extension of broadband and high-speed broadband provision, and of public wifi and 4G networks, enabling not only new types of audio experience, but higher sound quality. The BBC has used the internet to provide the best-ever quality delivery of stereo audio that it ever has, on services branded as "HD Sound" [6]

The days of internet streaming implying relatively poor audio quality when compared to traditional broadcasting are disappearing into history. However, some causes of audience dissatisfaction with audio persist and one of these is loudness, particularly variations in loudness.

A European Broadcasting Union (EBU) group has worked successfully to improve significantly the measurement and control of loudness of audio in broadcasting. It recommends [7] the adoption a new type of meter, and of a single target for the average loudness for all programmes. This removes from listeners the need to adjust the volume control at junctions between programmes or, sometimes with extra help from regulatory bodies, at junctions between programmes and commercial breaks. Its adoption is widespread throughout Europe, and it, or practices similar to it, are being adopted all round the world.

Whilst normalizing the *average* loudness of programmes goes a long way to improving audience satisfaction, one problem that this does not solve is that of the effect of the *range* of loudnesses within a programme coupled with large changes in noise level in the environment in which the listener is listening. A listener in an ideal listening environment will generally tolerate a much wider loudness range than one in a noisy environment. A listener in a noisy enviromnent could often find that the quietest parts of a programme are still too quiet when the volume control has been turned up so that the loudest parts are at the limit of comfort. Mobile listening means that the problem caused by variations in environmental noise is made more acute, leading to unsatisfactory experiences as the listener moves between quiet and noisy environments.

The approach taken by radio broadcasters to tailoring sound to suit noisy environments is to reduce the dynamic range by processing the signal before sending it to the transmitter. This means that all listeners, even those in quiet environments, receive a signal with a limited dynamic range. DAB (and digital TV) services can employ a different mechanism whereby the signal is broadcast in its original form, and gain control data is calculated and broadcast alongside the audio. The gain control data can be applied in the receiver in order to reduce the dynamic range. Unfortunately, this mechanism is not particularly effective in practice: it is not well used, the way it is presented to the user can be obscure, and it does not take account of the changing situation of the listener.

The Web Audio API offers a way to change all this for audio streamed over the internet. Dynamic range compression and gain control can be performed in the browser, in response to the user's individual requirements. Devices that have a microphone enable the noise level in the listening environment to be taken into account.

2. PRINCIPAL OF OPERATION

An overview of the way in which adaptive personalized compression has been implemented is shown in Figure 1.

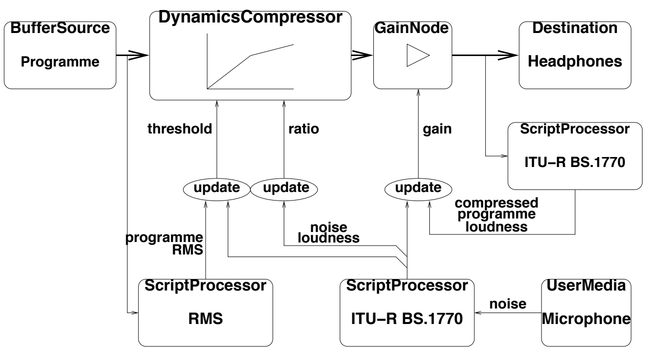

Figure 1: Web audio context of the adaptive personalised compressor

The noise level in the environment is continuously measured using the microphone in the device that is being used to play the audio programme. The programme loudness is kept 6LU above the environment noise by applying gain to the programme signal before it is sent to the output of the device. Measurement of the programme loudness and of the environment noise is made using the international standard algorithm in Recommendation ITU-R BS.1770, implemented using the ScriptProcessorNode. The use of a BiquadFilterNode would have been preferred for the application of the “k” frequency weighting in the loudness measurement algorithm, because this would be expected to be quicker than implementing a filter in Javascript. This was not possible because the BiquadFilterNode does not present a method for using the coefficients given in the Recommendation.

The gain is applied by a GainNode. Because the instantaneous value of gain required could fluctuate rapidly, smoothing is applied such that changes are made slowly, with a time constant of 1.5s for gain increases, and 0.2s for gain decreases.

A limit is set on the maximum gain that is applied, because at extremely high levels of environment noise boosting the programme sound would only add further to the risk of hearing damage.

Automatic adjustment of a volume control like this is only part of the solution, because the listener's tolerance for loudness *range* is also affected by high levels of environment noise.

The Web Audio API DynamicsCompressorNode is used to reduce the dynamic range of the programme. It has parameters to control threshold and ratio, knee width, and attack and release times. These are defined in the API as follows:

- "*threshold*" is the decibel value above which the compression will start taking effect
- "*knee*" is the decibel value representing the range above the threshold where the curve smoothly transitions to the "ratio" portion
- "*ratio*" is the amount of dB change in input for a 1 dB change in output
- "*attack time*" is the amount of time (in seconds) to reduce the gain by 10dB
- "*release time*" is the amount of time (in seconds) to increase the gain by 10dB.

Figure 2 illustrates the effect of threshold and ratio on the output signal level.

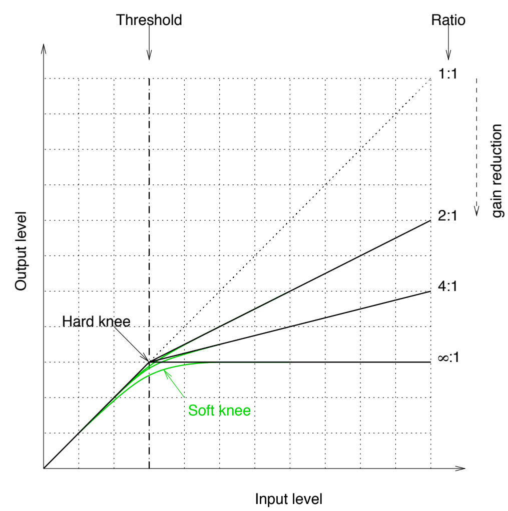

Figure 2: Transfer function of generic compressor, showing threshold, ratio, and knee

Information on the design of compressors is available in a paper by Giannoulis, et al. [8].

In this application, the knee width, attack and release times are set to fixed values, while the threshold and ratio are continually adjusted. The adjustment depends on the environment noise loudness and the programme level, and the mechanism used for this automatic control of the threshold and ratio parameters is derived from studies by Giannoulis et al. [9].

The ratio is varied according to the square of the environmental noise sound pressure level, starting with a value of unity at a SPL of 30dB (k-weighted). The curve is shown in Figure 3.

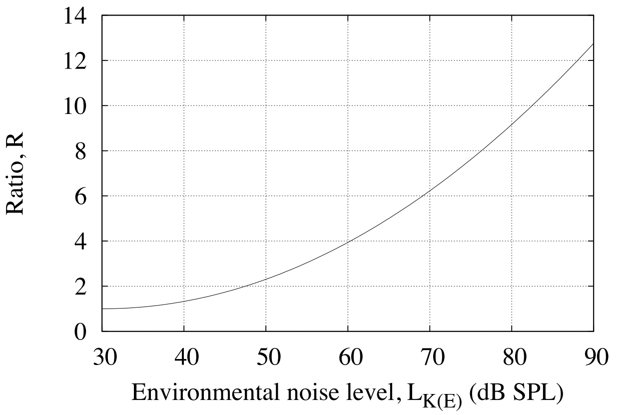

Figure 3: Compressor ratio as a function of environmental noise level

The threshold is adapted to stay around the RMS value of the programme, with an offset then applied, down or up depending on whether the environment noise sound pressure level is above or below 60dB (k-weighted). The result of this is shown in Figure 4, where the variation of threshold as a function of environment noise level is shown for the case when the RMS level of the programme is -25 dBFS. Different programme levels shift the curve along the vertical axis.

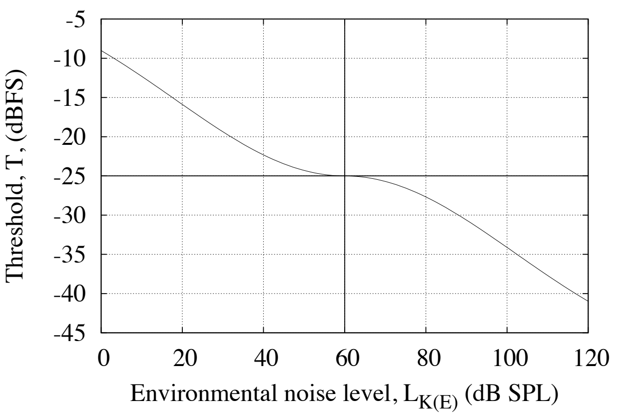

Figure 4: Compressor threshold as a function of environmental noise level, for an RMS programme level of -25 dBFS

Changes to the threshold and ratio of the compressor are smoothed, with a time constant of about 60ms.

The calculations of level and loudness are performed each time a new block of samples is ready, determined by the nodes' buffer size. The compressor parameters and gain parameter are updated asynchronously every 3ms by a process scheduled by a Javascript window.setInterval call.

3. USER INTERFACE

The experimental user interface developed to demonstrate the personalised compressor is shown in Figure 5. It was designed with as few controls as are required to operate it. In the figure, optional controls that are available, but not required, are displayed to the right of the main panel.

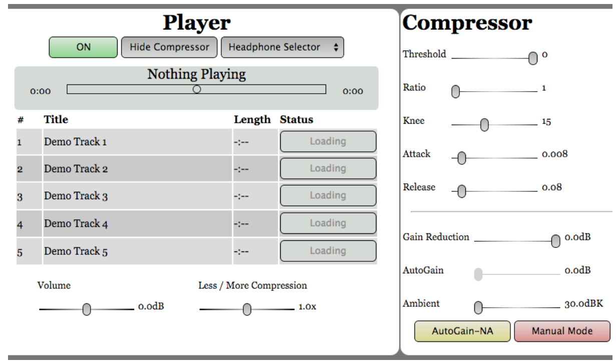

Figure 5: User interface of the development version of the personalised compressor

The interface was designed with both "traditional" computers and touch-sensitive devices in mind because consistency across devices is important for the user's experience.

The first task of the user interface is to perform the microphone calibration, which it does using a pop-up window telling the user what to do. Following that, the only controls offered are a volume control and a "Less/More" control. The volume control has a -20dB to +20dB range, preventing complete muting or too much boost. The "Less/More" control abstracts the multiplicity of controls of the compressor and presents them as one user control: less compression or more compression. Its value, in the range of 0 to 1, is used as a multiplier to scale the values of ratio and threshold.

As stated, the default controls are very minimal, but additional indicators and controls can be shown. Manual control can be taken of the compressor parameters, and they can be varied at will, but this usually only makes things worse.

4. EVALUATION

Feedback from an informal listening test conducted in the lab using open-backed headphones, a set of test programme material, and environmental noise from the BBC sound effects library played back over loudspeakers, was very encouraging. Listeners reported that the system was doing very much what they wanted, and that its operation was unobtrusive.

The choice of operating parameters appeared to have been made well, and listeners sometimes did not realise just what the processing had been doing until it was turned off. A few comments about excessive compression being apparent on one of the items could be addressed by simple adjustment of the "Less/More" slider.

The Web Audio API proved to be a solid platform for carrying out the implementation. However, as the specification is still under development, browser implementations vary in completeness and there are some inconsistencies that caused variations in the behaviour of the system.

The ScriptProcessorNode provides a flexible method for processing audio at the sample rather than stream level, but the performance of these nodes is not optimized and introduces a delay into the audio stream being processed which must be taken into account.

At the time of writing some platforms do not support all of the features required for personalized compression in the browser. In particular, the getUserMedia API for microphone capture is named differently by different browser vendors, and some do not support it at all.

5. CONCLUSIONS AND FURTHER WORK

This paper has described a demonstration system that has shown that personalised dynamic range control can easily be done in a web browser, responding to the environment around the listener. Demonstrations to listeners showed that the processing was unobtrusive and very effective at adapting to changes in environment noise. Future work will include the formal evaluation of the system.

6. ACKNOWLEDGEMENTS

The authors wish to thank Nick Jillings (Birmingham City University) and Zheng Ma (QMUL) for their work in developing the code for the demonstrator of personalised compression, James Vegnuti (Birmingham City University) for testing and optimising the parameters in it, and Chris Lowis (formerly of the BBC), Josh Reiss (QMUL), and Frank Melchior (BBC) for their guidance. The initial stages of this work were made possible by QMUL's Innovation Fund. The work has been carried out as part of the BBC Audio Research Partnership.

7. REFERENCE

[1] Ofcom, "Consumer Segmentation – Overview of the findings – July 2013", Ofcom, January 2014 [http://stakeholders.ofcom.org.uk/binaries/research/consumer-experience/tce-13/segmentation-report.pdf]

[2] RAJAR, "Listening to Radio Via A Mobile Phone", RAJAR Data Release, July 2014 [http://www.rajar.co.uk/docs/2014_06/RAJAR Q2 2014 Chart 6 - Mobile Phone Tablet listening - clean.pdf]

[3] Brun, R, "Saturday Drama: Under Milk Wood in Surround Sound", BBC, October 2014 [http://www.bbc.co.uk/blogs/radio4/posts/Saturday-Drama-Under-Milk-Wood-in-Surround-Sound]

[4] Mann, Churnside, Bonney, and Melchior, "Object-based audio applied to football broadcasts", Proceedings of the 2013 ACM International Workshop on Immersive Media Experiences, 21st ACM International Conference on Multimedia, Barcelona, October 2013

[5] BBC News, "BBC trials TV innovations at Commonwealth Games", BBC, July 2014 [http://www.bbc.co.uk/news/technology-28440076]

[6] BBC Press Office, "BBC launches HD Sound for radio", Ocober 2010 [http://www.bbc.co.uk/pressoffice/pressreleases/stories/2010/10_october/18/hd_radio.shtml]

[7] EBU, "R 128 - Loudness normalization and permitted maximum level of audio signals", EBU, June 2014 [https://tech.ebu.ch/docs/r/r128.pdf]

[8] Giannoulis, Massberg, and Reiss, "Digital Dynamic Range Compressor Design - A Tutorial and Analysis", Journal of the Audio Engineering Society, V. 60 (6), June 2012 [http://c4dm.eecs.qmul.ac.uk/audioengineering/compressors/documents/GiannoulisMassbergReiss-dynamicrangecompression-JAES2012.pdf]

[9] Giannoulis, Massberg, Reiss, "Parameter Automation in a Dynamic Range Compressor," Journal of the Audio Engineering Society, v61 (10), Oct. 2013 [http://c4dm.eecs.qmul.ac.uk/audioengineering/compressors/documents/Giannoulis Massberg Reiss - dynamic range compression automation - JAES 2013.pdf]

[10] Paradis, Gregrory-Clarke and Melchior, "VenueExplorer, object based interactive audio for live events" 1st Web Audio Conference, Paris, January 2015.